

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 1

1. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print (
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

```

```

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}

public static void main(String[] args){

```

```

        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

7. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi

- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata **abstract** o **final**

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
```

```
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

10. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}
```

```

    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.
- D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X() {
        System.out.print("X");
    }
}

```

```

    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25

- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 2

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:


```

B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

4. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

```

```

}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 1444444... (ciclo infinito)

- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

9. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

10. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
    }
}
```

```
System.out.print(e.print());
System.out.print(j.print());
}
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
    }
}
```

```

        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.
- D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 3

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}
```

```
class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
```

```

        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC

- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

6. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

7. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

-
8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
B. CCC
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Risultato diverso dai precedenti.

-
9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X

- B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

-
10. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
C. Definire `final` la variabile `i`.
D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
E. Nessuna delle precedenti.

-
11. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}
```

- A. 5
B. 6
C. 7
D. 8
E. Nessuna delle precedenti.
-

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
```

```
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

```
    }  
  }  
}
```

- A. 13Exception in thread "main" java.lang.Exception
 - B. 12
 - C. 1423Exception in thread "main" java.lang.Exception
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {  
    X() {  
        System.out.print("X");  
    }  
    abstract void z();  
}  
  
class Y extends X {  
    void z() {}  
    Y() {  
        System.out.print("Y");  
    }  
  
    public static void main(String[] args) {  
        X x = new Y();  
    }  
}
```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

16. Dato un file contenente il seguente codice:

```
package pk;  
  
public class A {  
    public final class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
 - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 - C. Un'istanza della classe I può essere costruita con sf new A.I().
 - D. Un'istanza della classe I può essere costruita con new A().new I().
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 4

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
}
```

```
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

5. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
```

```
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
    if("abcd" == s4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

6. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

7. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
```

```

        q();
    }
    catch( MyExc2 h ) {
        System.out.print(2);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc1() );
    }
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

9. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6

C. 7

D. 8

E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

A. A S B J 25

B. A S 20 B J 25

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

11. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile `i`.
- D. Definire protected la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Dato un file contenente il seguente codice:

```
package pk;

public class A {
```

```
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `A`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
- C. Un'istanza della classe `I` può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe `I` può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }

    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 5

1. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo *m*?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
```

```
Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

4. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;

- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
```

```
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC

- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X

- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

11. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

if(s1 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if("abcd" == s4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412

- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile `i`.
- D. Definire protected la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

14. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `A`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
- C. Un'istanza della classe `I` può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe `I` può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 1444444... (ciclo infinito)
 - B. 1463Exception in thread "main" MyExc1
 - C. Errore a tempo di compilazione
 - D. 143Exception in thread "main" MyExc1
 - E. Nessuna delle precedenti
-

16. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
 - B. y = (B1) b;
 - C. b = (B2) y;
 - D. y = (B1) h;
 - E. Nessuno dei precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 6

1. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
```

```
((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

3. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

4. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
```



```

    } else {
        System.out.print(0);
    }
    if(s1 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if("abcd" == s4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
}

```

```

}
public B g() {return this;}
public static void main(String[] args){
    new C().m();
}
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

9. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}

public static void main(String[] args){
    X x = new Y();
}
}

```

- A. X

- B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

11. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
 - B. y = (B1) b;
 - C. b = (B2) y;
 - D. y = (B1) h;
 - E. Nessuno dei precedenti
-

12. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
 - B. m = a;
 - C. m = (Error []) q;
 - D. q = a;
 - E. q = (Boolean []) m;
-

13. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
 - B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - C. Definire final la variabile i.
 - D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - E. Nessuna delle precedenti.
-

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
 - B. 12
 - C. 1423Exception in thread "main" java.lang.Exception
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 7

1. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

2. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

3. Quale output si ottiene invocando il metodo `p`?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

4. Qual'è l'output di questo codice?

```
class MyExcl extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
```

```

        q();
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new Exception() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

```

```

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

A. ABC

B. CCC

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Risultato diverso dai precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
    public static void main(String[] args){
        X x = new Y();
    }
}
```

A. X

B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

A. A S B J 25

B. A S 20 B J 25

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
```



```

    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

12. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.
- D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }
}

```

```

}

public String print() {
    return name + " ";
}

}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

```

}
```

- A. 1444444... (ciclo infinito)
 - B. 1463Exception in thread "main" MyExc1
 - C. Errore a tempo di compilazione
 - D. 143Exception in thread "main" MyExc1
 - E. Nessuna delle precedenti
-

16. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
 - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 - C. Un'istanza della classe I può essere costruita con sf new A.I().
 - D. Un'istanza della classe I può essere costruita con new A().new I().
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 8

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire **private** la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire **final** la variabile *i*.
- D. Definire **protected** la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
```

```

    }
    catch( MyExc1 k ) {
    }
    finally {
        System.out.print(2);
    }
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

if(s1 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if("abcd" == s4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata **abstract** o **final**

15. q: Date le dichiarazioni:

```
Object [] a;  
Error [] m;  
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

16. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {  
    String name = new String();  
    static Integer value = new Integer();  
  
    static void getCurrency(Integer v) {  
        Currency c = new Currency();  
        c.value = v;  
    }  
  
    public static void m() {  
        Currency.getCurrency(50);  
        Currency.getCurrency(100);  
    }  
  
    public static void main(String[] argv) {  
        Currency.m();  
    }  
}
```

- A. 5
 - B. 6
 - C. 7
 - D. 8
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 9

1. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p ();
        }
        catch( MyExc2 e ) {
            System.out.print (1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print (2);
        }
    }
}
```

```
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print (3);
    }
    catch( Exception j ) {
        System.out.print (4);
    }
    finally {
        throw( new MyExc2 () );
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency (50);
        Currency.getCurrency (100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6

C. 7

D. 8

E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

A. A S B J 25

B. A S 20 B J 25

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }
}
```

```
public String print() {
    return name + " ";
}
```

```
}
```

```
class Student extends Person {
    Student(String n, int a) {super(n,a);}
}
```

```
public String print() {
    return name + " " + age + " ";
}
```

```
}
```

```
class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

```
}
```

A. A S B J 25

B. A S 20 B J 25

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}
```

```
class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}
```

```
public static void main(String[] args){
    X x = new Y();
}
```

```
}
```

A. X

B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

7. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

9. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

11. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;

- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}
```

```
class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

A. 13Exception in thread "main" ja-

va.lang.Exception

B. 12

C. 1423Exception in thread "main" java.lang.Exception

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

16. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

C. Definire final la variabile i.

D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 10

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
```

```
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

3. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.
- D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
    }
}
```



```

    catch( MyExc3 z ) {
    }
    catch( MyExc1 k ) {
    }
    finally {
        System.out.print(2);
    }
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

```

```

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);

```

```

        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

7. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

8. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `A`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
- C. Un'istanza della classe `I` può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe `I` può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

9. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
```

```
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}

public static void main(String[] args){

```

```

        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

```

```

    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 11

1. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

3. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

4. Qual'è l'output di questo codice?

```
class MyExcl extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
```

```

        System.out.print (5);
    }
    finally {
        throw( new MyExc2 () );
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire **private** la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire **final** la variabile *i*.
- D. Definire **protected** la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p ();
        }
        catch( MyExc2 e ) {
            System.out.print (1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
    }
}

```

```

        finally {
            System.out.print (2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print (3);
        }
        catch( Exception j ) {
            System.out.print (4);
        }
        finally {
            throw( new MyExc2 () );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Quale output si ottiene invocando il metodo *p*?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print (1);
        } else {
            System.out.print (0);
        }
        if(s1 == s3) {
            System.out.print (1);
        } else {
            System.out.print (0);
        }
        if("abcd" == s4) {
            System.out.print (1);
        } else {
            System.out.print (0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}
```

```
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

11. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:


```

B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

```

```

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
    }
}

```

```

catch( MyExc2 h ) {
    System.out.print(2);
    throw( new MyExc2() );
}
finally {
    System.out.print(3);
    throw( new MyExc1() );
}
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
 - B. 1463Exception in thread "main" MyExc1
 - C. Errore a tempo di compilazione
 - D. 143Exception in thread "main" MyExc1
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 12

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
}
```

```
    }
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

3. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire **private** la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire **final** la variabile *i*.
- D. Definire **protected** la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;

E. q = (Boolean []) m;

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

8. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;  
Super h;  
B1 y;  
h = new B2 ();  
b = new B2 ();  
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {  
    String name;  
    int age;  
  
    Person(String n, int a) {  
        name = n;  
        age = a;  
    }  
  
    public String print () {  
        return name + " ";  
    }  
}  
  
class Student extends Person {  
    Student(String n, int a) {super(n,a);}  
  
    public String print () {  
        return name + " " + age + " ";  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        Person e = new Student("A S", 20);  
        Student j = new Student("B J", 25);  
        System.out.print(e.print());  
        System.out.print(j.print());  
    }  
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

10. Quale output si ottiene invocando il metodo p?

```
class G {  
    private String [] a3 = new String [4];  
    void p() {  
        String [] a2;  
        String s2;  
        a2 = new String [4];  
        s2 = "abc";  
        m(a2, s2);  
    }  
    void m(String [] a1, String s1) {  
        String s3;  
        String s4 = "abcd";  
        s3 = "abc";  
        if(a3 == a1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s1 == s3) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if("abcd" == s4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

11. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {  
    String name = new String();  
    static Integer value = new Integer();  
  
    static void getCurrency(Integer v) {  
        Currency c = new Currency();  
        c.value = v;  
    }  
  
    public static void m() {  
        Currency.getCurrency(50);  
        Currency.getCurrency(100);  
    }  
}
```

```

}

public static void main(String[] argv) {
    Currency.m();
}
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {

```

```

    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception

B. 12

C. 1423Exception in thread "main" java.lang.Exception

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}
```

```
class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

A. A S B J 25

B. A S 20 B J 25

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

16. Dire quale delle seguenti affermazioni è vera:

A. La lunghezza di un array può essere variata dopo la sua costruzione

B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi

C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri

D. Un array non ha un'unica superclasse

E. Una classe non può essere dichiarata abstract o final

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 13

1. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
```

```

String name = "B";
abstract B g();
}

class C extends B {
String name = "C";
void m() {
System.out.print(
((A)this).name +
((B)this).name +
((C)this).name);
}
B g() {return this;}
public static void main(String[] args){
new C().m();
}
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

5. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

6. Dato un file contenente il seguente codice:

```

package pk;

public class A {
public final class I {
}
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
public static void main(String [] argv) {
try {
System.out.print(1);
q();
}
catch( MyExc2 h ) {
System.out.print(2);
throw( new MyExc2() );
}
finally {
System.out.print(3);
throw( new MyExc1() );
}
}
static void q() {
try {
throw( new MyExc3() );
}
catch( MyExc3 k ) {
System.out.print(4);
throw( new MyExc3() );
}
catch( MyExc1 t ) {
System.out.print(5);
}
finally {
System.out.print(6);
}
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
```

```
        System.out.print (
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

11. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
 - D. Un array non ha un'unica superclasse
 - E. Una classe non può essere dichiarata abstract o final
-

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
    }
}
```

```
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new Exception() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}
```

- A. 13Exception in thread "main" java.lang.Exception
 - B. 12
 - C. 1423Exception in thread "main" java.lang.Exception
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l) {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

15. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 111

B. 011

C. 100

D. 010

E. 101

16. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

C. Definire final la variabile i.

D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 14

1. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

2. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
    }
}
```

```
    }
    catch( MyExc2 h ) {
        System.out.print(2);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc1() );
    }
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}
```

- A. 14444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}
```



```

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

5. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }
}

```

```

        public String print() {
            return name + " ";
        }
    }

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
    }
}

```

```

    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

13. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
    }
}
```

```

    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}

```

```

    }
    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 15

1. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

```
} else {
    System.out.print(0);
}
if("abcd" == s4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

2. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
    if(s1 == s3) {
        System.out.print(1);
    }
}
```

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
    }
}
```

```

        System.out.print (e.print ());
        System.out.print (j.print ());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print (e.print ());
        System.out.print (j.print ());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print (
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

```

    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

8. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

9. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

10. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `A`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.

- C. Un'istanza della classe l può essere costruita con sf new A.I().
- D. Un'istanza della classe l può essere costruita con new A().new l().
- E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
    }
}
```

```
}
catch( MyExc2 x ) {
    System.out.print(5);
}
finally {
    throw( new MyExc2() );
}
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
    }
}
```

```
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}
```

A. 412

B. 4

C. 442

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

16. Dire quale delle seguenti affermazioni è vera:

A. La lunghezza di un array può essere variata dopo la sua costruzione

B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi

C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri

D. Un array non ha un'unica superclasse

E. Una classe non può essere dichiarata abstract o final

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 16

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

2. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;

- D. q = a;
- E. q = (Boolean []) m;

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }

    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
    if("abcd" == s4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

8. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

10. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
}

```

```

static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.
- D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
    }
}

```

```

catch( MyExc2 h ) {
    System.out.print(2);
    throw( new MyExc2() );
}
finally {
    System.out.print(3);
    throw( new MyExc1() );
}
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

14. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
 B. 12

C. 1423Exception in thread "main" java.lang.Exception

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

16. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
 B. y = (B1) b;
 C. b = (B2) y;
 D. y = (B1) h;
 E. Nessuno dei precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 17

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
```

```
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
 - B. `m = a;`
 - C. `m = (Error []) q;`
 - D. `q = a;`
 - E. `q = (Boolean []) m;`
-

4. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
 - D. Un array non ha un'unica superclasse
 - E. Una classe non può essere dichiarata `abstract` o `final`
-

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
 - B. A S 20 B J 25
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

7. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }
}
```

```

public static void main(String[] argv) {
    Currency.m();
}
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

```

```

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }
}

```

```

public static void main(String[] args) {
    X x = new Y();
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

```

```

abstract class B implements A {
    String name = "B";
    abstract B g();
}

```

```

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

11. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}
```

```
class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {

```

```

    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 13Exception in thread "main" java.lang.Exception
 - B. 12
 - C. 1423Exception in thread "main" java.lang.Exception
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

15. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.

- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - C. Definire final la variabile i.
 - D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - E. Nessuna delle precedenti.
-

16. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
 - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 - C. Un'istanza della classe I può essere costruita con sf new A.I().
 - D. Un'istanza della classe I può essere costruita con new A().new I().
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 18

1. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?


```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

5. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

6. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

```

```

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
```

```
System.out.print(1);
q();
System.out.print(2);
}
finally {
    System.out.print(3);
    throw( new Exception() );
}
}
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 14444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

13. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi

14. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

15. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }
}
```

```

public static void m() {
    Currency.getCurrency(50);
    Currency.getCurrency(100);
}

public static void main(String[] argv) {
    Currency.m();
}
}

```

- A. 5
 - B. 6
 - C. 7
 - D. 8
 - E. Nessuna delle precedenti.
-

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);

```

```

        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
 - B. 4
 - C. 442
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 19

1. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
```

```
Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

4. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l) {
        System.out.print("X");
    }
}
```

```
abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

8. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
}

```

```

B g() {return this;}
public static void main(String[] args){
    new C().m();
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

10. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
    }
}

```



```

        System.out.print (e.print ());
        System.out.print (j.print ());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print (1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print (2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print (3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print (4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print (5);
        }
        finally {
            System.out.print (6);
        }
    }
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

13. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.
- D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print (1);
            q();
            System.out.print (2);
        }
        finally {
            System.out.print (3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print (4);
        }
        catch( MyExc2 x ) {
            System.out.print (5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
```

```
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

A. 412

B. 4

C. 442

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

16. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

A. m = (Error []) a;

B. m = a;

C. m = (Error []) q;

D. q = a;

E. q = (Boolean []) m;

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 20

1. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}
```

```
class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

4. **q:** Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 14444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

6. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
```

```
String s4 = "abcd";
s3 = "abc";
if(a3 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s1 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if("abcd" == s4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

7. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

8. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe l può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe l può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe l può essere costruita con sf new A.I().
- D. Un'istanza della classe l può essere costruita con new A().new l().
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {

```

```

        System.out.print(2);
    }
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

```

```

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
 - B. A S 20 B J 25
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

```

```

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}

```

```

}
public static void main(String[] args){
    X x = new Y();
}
}

```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

16. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
 - D. Un array non ha un'unica superclasse
 - E. Una classe non può essere dichiarata abstract o final
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 21

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

2. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)

- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();
```

```
static void getCurrency(Integer v) {
    Currency c = new Currency();
    c.value = v;
}

public static void m() {
    Currency.getCurrency(50);
    Currency.getCurrency(100);
}

public static void main(String[] argv){
    Currency.m();
}
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25

- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

10. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
    }
}
```

```

s3 = "abc";
if(a3 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s1 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if("abcd" == s4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

11. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}

```

```

public static void main(String[] args){
    X x = new Y();
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

```

```

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}

```

```

public static void main(String[] args){
    X x = new Y();
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile `i`.

D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

E. Nessuna delle precedenti.

15. Dire quale delle seguenti affermazioni è vera:

A. La lunghezza di un array può essere variata dopo la sua costruzione

B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi

C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri

D. Un array non ha un'unica superclasse

E. Una classe non può essere dichiarata abstract o final

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
```

```
        q();
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new Exception() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}
```

A. 13Exception in thread "main" java.lang.Exception

B. 12

C. 1423Exception in thread "main" java.lang.Exception

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 22

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

2. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;

- D. q = a;
- E. q = (Boolean []) m;

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class Al {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

5. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }

    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

```
    }  
  }  
}
```

- A. 412
 - B. 4
 - C. 442
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {  
    static String name;  
    static int age;  
  
    Person(String n, int a) {  
        name = n;  
        age = a;  
    }  
  
    public static String print() {  
        return name + " ";  
    }  
}  
  
class Student extends Person {  
    Student(String n, int a) {super(n,a);}  
  
    public static String print() {  
        return name + " " + age + " ";  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        Person e = new Student("A S", 20);  
        Student j = new Student("B J", 25);  
        System.out.print(e.print());  
        System.out.print(j.print());  
    }  
}
```

- A. A S B J 25
 - B. A S 20 B J 25
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {  
    String name = "A";  
    void m();  
}  
  
abstract class B implements A {  
    String name = "B";  
    abstract B g();  
}  
  
class C extends B {  
    String name = "C";  
    void m() {  
        System.out.print(  
            ((A)this).name +  
            ((B)this).name +  
            ((C)this).name);  
    }  
    B g() {return this;}  
    public static void main(String[] args){  
        new C().m();  
    }  
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {  
    String name;  
    int age;  
  
    Person(String n, int a) {  
        name = n;  
        age = a;  
    }  
  
    public String print() {  
        return name + " ";  
    }  
}  
  
class Student extends Person {  
    Student(String n, int a) {super(n,a);}  
  
    public String print() {  
        return name + " " + age + " ";  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        Person e = new Student("A S", 20);
```

```

        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.
- D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {

```

```

        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X

- B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

15. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
    }
}
}
```

- A. 111
 - B. 011
 - C. 100
 - D. 010
 - E. 101
-

16. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
 - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 - C. Un'istanza della classe I può essere costruita con sf new A.I().
 - D. Un'istanza della classe I può essere costruita con new A().new I().
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 23

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
 B. A S 20 B J 25
 C. Errore a tempo di compilazione.
 D. Errore a tempo di esecuzione.
 E. Nessuna delle precedenti.
-
2. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();
```

```
static void getCurrency(Integer v) {
    Currency c = new Currency();
    c.value = v;
}

public static void m() {
    Currency.getCurrency(50);
    Currency.getCurrency(100);
}

public static void main(String[] argv){
    Currency.m();
}
}
```

- A. 5
 B. 6
 C. 7
 D. 8
 E. Nessuna delle precedenti.

3. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
 B. y = (B1) b;
 C. b = (B2) y;
 D. y = (B1) h;
 E. Nessuno dei precedenti

4. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

5. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.

D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

E. Nessuna delle precedenti.

6. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
}

```

```

}
public B g() {return this;}
public static void main(String[] args){
    new C().m();
}
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

10. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. q: Date le dichiarazioni:

```
Object [] a;  
Error [] m;  
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            p();  
        }  
        catch( MyExc2 e ) {  
            System.out.print(1);  
        }  
        catch( MyExc3 z ) {  
        }  
        catch( MyExc1 k ) {  
        }  
        finally {  
            System.out.print(2);  
        }  
    }  
    static void p() throws Exception {  
        try {  
            throw( new Exception() );  
        }  
        catch( MyExc3 c ) {  
            System.out.print(3);  
        }  
        catch( Exception j ) {  
            System.out.print(4);  
        }  
        finally {  
            throw( new MyExc2() );  
        }  
    }  
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        finally {  
            System.out.print(3);  
            throw( new Exception() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
        }  
        catch( Exception t ) {  
            System.out.print(4);  
        }  
        catch( MyExc2 x ) {  
            System.out.print(5);  
        }  
        finally {  
            throw( new MyExc2() );  
        }  
    }  
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {  
    String name = "A";  
    void m();  
}  
  
abstract class B implements A {  
    String name = "B";  
    abstract B g();  
}  
  
class C extends B {  
    String name = "C";
```

```

void m() {
    System.out.print (
        ((A)this).name +
        ((B)this).name +
        ((C)this).name);
}
B g() {return this;}
public static void main(String[] args){
    new C().m();
}
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {

```

```

            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 14444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 24

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

5. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}
```

```

    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

```

```

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
        }
    }
}

```

```

        q();
    }
    catch( MyExc2 h ) {
        System.out.print(2);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc1() );
    }
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

13. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

14. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

15. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }
}

```

```

public static void m() {
    Currency.getCurrency(50);
    Currency.getCurrency(100);
}

public static void main(String[] argv) {
    Currency.m();
}
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);

```

```

        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 25

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}
```

- A. 5
- B. 6

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

- C. 7
- D. 8
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
}
```

```
void m(String [] a1, String s1) {
    String s3;
    String s4 = "abcd";
    s3 = "abc";
    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s1 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if("abcd" == s4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

8. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

9. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

12. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

```

```

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

```

```

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

15. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi

- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

16. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
 - B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - C. Definire final la variabile i.
 - D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 26

1. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
    }
}
```

```
finally {
    System.out.print(6);
}
}
```

- A. 14444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

3. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}
```



```

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

```

```

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {

```

```

        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

```

```

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

```

```

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

8. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

9. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
```

```
        System.out.print (
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
```

```

String s3;
String s4 = "abcd";
s3 = "abc";
if(a3 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s1 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if("abcd" == s4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

```

}
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

A. 13Exception in thread "main" java.lang.Exception

B. 12

C. 1423Exception in thread "main" java.lang.Exception

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

16. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

C. Definire final la variabile i.

D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 27

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire **private** la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire **final** la variabile *i*.
- D. Definire **protected** la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
    }
}
```

```

        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
    }
}

```

```

        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
}

```

```

    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

8. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {

```

```

try {
}
catch( Exception t ) {
    System.out.print(4);
}
catch( MyExc2 x ) {
    System.out.print(5);
}
finally {
    throw( new MyExc2() );
}
}
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

11. Quale output si ottiene invocando il metodo `p`?


```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

12. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

```
    }  
  }  
}
```

- A. 1444444... (ciclo infinito)
 - B. 1463Exception in thread "main" MyExc1
 - C. Errore a tempo di compilazione
 - D. 143Exception in thread "main" MyExc1
 - E. Nessuna delle precedenti
-

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {  
    String name = "A";  
    void m();  
}  
  
abstract class B implements A {  
    String name = "B";  
    abstract B g();  
}  
  
class C extends B {  
    String name = "C";  
    public void m() {  
        System.out.print(  
            ((A)this).name +  
            ((B)this).name +  
            ((C)this).name);  
    }  
    public B g() {return this;}  
    public static void main(String[] args){  
        new C().m();  
    }  
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {  
    X(){  
        System.out.print("X");  
    }  
    abstract void z();  
}  
  
class Y extends X {  
    void z(){}  
    Y(){  
        System.out.print("Y");  
    }  
  
    public static void main(String[] args){  
        X x = new Y();  
    }  
}
```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 28

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

```
        throw( new MyExc2() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc1() );
    }
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}
```

- A. 14444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
```

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}
```

```

public static void main(String[] args) {
    X x = new Y();
}
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

5. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.

- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Quale output si ottiene invocando il metodo `p`?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
}

```

```

void m(String [] a1, String s1) {
    String s3;
    String s4 = "abcd";
    s3 = "abc";
    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s1 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if("abcd" == s4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

8. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;
}

```

```

Person(String n, int a) {
    name = n;
    age = a;
}

public static String print() {
    return name + " ";
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }
}
```

```
    }
    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

14. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

15. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

16. `q`: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
 - B. `m = a;`
 - C. `m = (Error []) q;`
 - D. `q = a;`
 - E. `q = (Boolean []) m;`
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 29

1. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

2. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7

D. 8

E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
    }
}
```

```

    }
    catch( MyExc2 h ) {
        System.out.print(2);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc1() );
    }
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {

```

```

        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {

```

```

    }
    finally {
        System.out.print(2);
    }
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

13. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

14. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.

D. Definire protected la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

E. Nessuna delle precedenti.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l) {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

A. X

B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

A. X

B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 30

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}
```

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?


```

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
 - B. A S 20 B J 25
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

4. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
 - B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - C. Definire `final` la variabile `i`.
 - D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - E. Nessuna delle precedenti.
-

5. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `A`.
 - B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
 - C. Un'istanza della classe `I` può essere costruita con `sf new A.I()`.
 - D. Un'istanza della classe `I` può essere costruita con `new A().new I()`.
 - E. Nessuna delle precedenti.
-

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
 - B. 12
 - C. 1423Exception in thread "main" java.lang.Exception
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

9. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

12. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

14. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();

```

```

        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
 - B. 4
 - C. 442
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 31

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

2. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;

- D. q = a;
- E. q = (Boolean []) m;

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
    }
}
```

```

    catch( MyExc3 z ) {
    }
    catch( MyExc1 k ) {
    }
    finally {
        System.out.print(2);
    }
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

9. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
```

```
String s4 = "abcd";
s3 = "abc";
if(a3 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s1 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if("abcd" == s4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

10. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

11. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.

- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }
}
```

```
public String print() {
    return name + " ";
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
```

```

        throw( new MyExc2() );
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().

E. Nessuna delle precedenti.

16. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 32

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
```

```
public static void main(String [] argv)
    throws Exception {
    try {
        p();
    }
    catch( MyExc2 e ) {
        System.out.print(1);
    }
    catch( MyExc3 z ) {
    }
    catch( MyExc1 k ) {
    }
    finally {
        System.out.print(2);
    }
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
```

```

}
class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
}

```

```

}
finally {
    System.out.print(3);
    throw( new MyExc1() );
}
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
}
finally {
    System.out.print(6);
}
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

7. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}
```

```
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

10. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

13. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
 - B. A S 20 B J 25
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

16. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
 - B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - C. Definire final la variabile i.
 - D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 33

1. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}
```

```
class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

4. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}
```

```
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }

    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
```

```

        throw( new MyExc2() );
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class Al {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

```

    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

12. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

13. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X

- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 34

1. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}
```

```
class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

4. Qual'è l'output di questo codice?

```
class MyExcl extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
    }
}
```



```

    catch( MyExc3 z ) {
    }
    catch( MyExc1 k ) {
    }
    finally {
        System.out.print(2);
    }
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {

```

```

        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);

```

```

        throw( new MyExc2() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc1() );
    }
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

12. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

13. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Risultato diverso dai precedenti.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 35

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {

```

```

        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

7. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

8. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}
```

- A. ABC

- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}
```

```
class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

14. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().

D. Un'istanza della classe l può essere costruita con new A().new l().

E. Nessuna delle precedenti.

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

```
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

A. 1444444... (ciclo infinito)

B. 1463Exception in thread "main" MyExc1

C. Errore a tempo di compilazione

D. 143Exception in thread "main" MyExc1

E. Nessuna delle precedenti

16. Dire quale delle seguenti affermazioni è vera:

A. La lunghezza di un array può essere variata dopo la sua costruzione

B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi

C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri

D. Un array non ha un'unica superclasse

E. Una classe non può essere dichiarata abstract o final

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 36

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.

- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l) {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }
}
```

```

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {

```

```

            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print (
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }
}

```

```

    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

10. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

11. Qual'è l'output di questo codice?

```

class MyExcl extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
        }
    }
}

```

```

        throw( new Exception() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

13. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
    }
}

```

```

if(a3 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s1 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if("abcd" == s4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

```
    }  
  }  
}
```

- A. 412
 - B. 4
 - C. 442
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

15. Date le dichiarazioni:

```
class B1 extends Super {...}  
class Super extends Object {...}  
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;  
Super h;  
B1 y;  
h = new B2 ();  
b = new B2 ();  
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`

E. Nessuno dei precedenti

16. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {  
    String name = new String();  
    static Integer value = new Integer();  
  
    static void getCurrency(Integer v) {  
        Currency c = new Currency();  
        c.value = v;  
    }  
  
    public static void m() {  
        Currency.getCurrency(50);  
        Currency.getCurrency(100);  
    }  
  
    public static void main(String[] argv) {  
        Currency.m();  
    }  
}
```

- A. 5
 - B. 6
 - C. 7
 - D. 8
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 37

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l) {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
```

```
Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

3. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
}
```

```

static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25

- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

7. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.
- D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

```

```

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
}

```

```

}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011

- C. 100
 - D. 010
 - E. 101
-

15. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
 - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 - C. Un'istanza della classe I può essere costruita con `sf new A.I()`.
 - D. Un'istanza della classe I può essere costruita con `new A().new I()`.
 - E. Nessuna delle precedenti.
-

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 38

1. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

2. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

3. Quale output si ottiene invocando il metodo `p`?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
```

```
a2 = new String [4];
s2 = "abc";
m(a2, s2);
}
void m(String [] a1, String s1) {
    String s3;
    String s4 = "abcd";
    s3 = "abc";
    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s1 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if("abcd" == s4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

4. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

- C. Definire final la variabile i.
- D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

5. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

6. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
        }
    }
}
```

```

        throw( new Exception() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
    }
}

```

```

        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
}

```

```

Y() {
    System.out.print ("Y");
}

public static void main(String[] args) {
    X x = new Y();
}
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +

```

```
        ((B)this).name +
        ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

15. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 39

1. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

2. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

5. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
```

```
public static void main(String [] argv) {
    try {
        System.out.print(1);
        q();
    }
    catch( MyExc2 h ) {
        System.out.print(2);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc1() );
    }
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

7. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
```



```

        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

13. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.

- D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

```

```

}
class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
}

```

```

        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
 - B. 4
 - C. 442
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 40

1. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

4. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
```

```
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata **abstract** o **final**

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}
```

```

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print (
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

9. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print () {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print () {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print (e.print ());
        System.out.print (j.print ());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

-
11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

-
12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}
```

- A. ABC
B. CCC

- C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Risultato diverso dai precedenti.

-
13. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
B. 6
C. 7
D. 8
E. Nessuna delle precedenti.

-
14. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
C. Un'istanza della classe I può essere costruita con sf new A.I().
D. Un'istanza della classe I può essere costruita con new A().new I().
E. Nessuna delle precedenti.

-
15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 1444444... (ciclo infinito)
 - B. 1463Exception in thread "main" MyExc1
 - C. Errore a tempo di compilazione
 - D. 143Exception in thread "main" MyExc1
 - E. Nessuna delle precedenti
-

16. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
 - B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - C. Definire final la variabile i.
 - D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 41

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}
```

- A. 5
- B. 6

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

C. 7

D. 8

E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

A. A S B J 25

B. A S 20 B J 25

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
}
```

```
}
catch( MyExc2 e ) {
    System.out.print(1);
}
catch( MyExc3 z ) {
}
catch( MyExc1 k ) {
}
finally {
    System.out.print(2);
}
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}
```

A. 412

B. 4

C. 442

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

```
}  
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

7. Date le seguenti classi:

```
class A {  
    Integer i = 6;  
}  
class B extends A {  
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
 - B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - C. Definire final la variabile i.
 - D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - E. Nessuna delle precedenti.
-

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {  
    static String name;  
    static int age;  
  
    Person(String n, int a) {  
        name = n;  
        age = a;  
    }  
  
    public static String print() {  
        return name + " ";  
    }  
}  
  
class Student extends Person {  
    Student(String n, int a) {super(n,a);}  
  
    public static String print() {  
        return name + " " + age + " ";  
    }  
}
```

```
class Test {  
    public static void main(String[] args){  
        Person e = new Student("A S", 20);  
        Student j = new Student("B J", 25);  
        System.out.print(e.print());  
        System.out.print(j.print());  
    }  
}
```

- A. A S B J 25
 - B. A S 20 B J 25
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

9. Date le dichiarazioni:

```
class B1 extends Super {...}  
class Super extends Object {...}  
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;  
Super h;  
B1 y;  
h = new B2();  
b = new B2();  
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
 - B. y = (B1) b;
 - C. b = (B2) y;
 - D. y = (B1) h;
 - E. Nessuno dei precedenti
-

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        finally {  
            System.out.print(3);  
            throw( new Exception() );  
        }  
    }  
}
```

```

static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

13. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

14. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

15. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 42

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

2. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.

- B. Definire *private* la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire *final* la variabile *i*.
- D. Definire *protected* la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

3. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
}
```



```

    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

7. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
 - B. 6
 - C. 7
 - D. 8
 - E. Nessuna delle precedenti.
-

8. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
 - B. `y = (B1) b;`
 - C. `b = (B2) y;`
 - D. `y = (B1) h;`
 - E. Nessuno dei precedenti
-

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

10. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
 - D. Un array non ha un'unica superclasse
 - E. Una classe non può essere dichiarata abstract o final
-

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
 - B. A S 20 B J 25
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
```

```
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l) {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {

```

```

        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 43

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione

- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

3. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

4. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}
```

- A. 5

- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args) {
```

```
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l) {
        System.out.print("X");
    }
    abstract void z();
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

```
class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
```



```

try {
    p();
}
catch( MyExc2 e ) {
    System.out.print(1);
}
catch( MyExc3 z ) {
}
catch( MyExc1 k ) {
}
finally {
    System.out.print(2);
}
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;

- D. y = (B1) h;
- E. Nessuno dei precedenti

13. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception

- B. 12
 - C. 1423Exception in thread "main" java.lang.Exception
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

15. Date le seguenti classi:

```
class A {  
    Integer i = 6;  
}  
class B extends A {  
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
 - B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - C. Definire final la variabile i.
 - D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - E. Nessuna delle precedenti.
-

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {  
    String name = "A";  
    void m();  
}  
  
abstract class B implements A {  
    String name = "B";  
    abstract B g();  
}  
  
class C extends B {  
    String name = "C";  
    public void m() {  
        System.out.print(  
            ((A)this).name +  
            ((B)this).name +  
            ((C)this).name);  
    }  
    public B g() {return this;}  
    public static void main(String[] args){  
        new C().m();  
    }  
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 44

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

2. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

3. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
    }
}
```

```

        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {

```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s1 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if("abcd" == s4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

10. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
    }
}

```

```

    }
    catch( MyExc2 h ) {
        System.out.print(2);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc1() );
    }
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

14. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

15. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile `i`.
- D. Definire protected la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
```

```
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
}
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
 - B. 12
 - C. 1423Exception in thread "main" java.lang.Exception
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 45

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

3. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}
```

```
class Y extends X {
    void z(){}
}
```

```

Y() {
    System.out.print ("Y");
}

public static void main(String[] args) {
    X x = new Y();
}
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print (1);
        } else {
            System.out.print (0);
        }
        if(s1 == s3) {
            System.out.print (1);
        } else {
            System.out.print (0);
        }
        if("abcd" == s4) {
            System.out.print (1);
        } else {
            System.out.print (0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print (
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
    }
}

```

```

        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 14444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

11. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print (
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

13. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

14. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 46

1. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`

- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

3. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `A`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
- C. Un'istanza della classe `I` può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe `I` può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

4. `q`: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

5. Qual'è l'output di questo codice?


```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
}

```

```

public static void main(String[] args){
    new C().m();
}
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

9. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }
}

public static void main(String[] args){
    X x = new Y();
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

```

```

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile *i*.
- D. Definire protected la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
    }
}

```

```

    }
    catch( MyExc2 h ) {
        System.out.print(2);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc1() );
    }
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 47

1. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

```
}
}
}
```

- A. 14444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

3. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
        }
    }
}
```

```

        throw( new Exception() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

7. Date le seguenti classi:

```

class A {
    Integer i = 6;
}

class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

8. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

9. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?


```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
}

```

```

public B g() {return this;}
public static void main(String[] args){
    new C().m();
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

15. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`

E. Nessuno dei precedenti

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}
```

```
class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}
```

```
public static void main(String[] args){
    X x = new Y();
}
```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 48

1. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata **abstract** o **final**

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

3. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo codice?

```
class MyExcl extends Exception { }
class MyExc2 extends MyExcl { }
class MyExc3 extends MyExcl { }
public class Al {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExcl() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
```

```

        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

6. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {

```

```

        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {

```

```

        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

11. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.

- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }

    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {

```

```

        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 49

1. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

2. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

3. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

4. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

5. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire **private** la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire **final** la variabile `i`.
- D. Definire **protected** la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

7. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
```

```

        throw( new MyExc2() );
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}
```

```

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
}

```

```

        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
 - B. 4
 - C. 442
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 50

1. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

2. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

3. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
        }
    }
}
```



```

        throw( new MyExc1() );
    }
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}
```

```
class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}
```

```
class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
```

```

        throw( new MyExc2() );
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile `i`.
- D. Definire protected la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

12. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`

E. Nessuno dei precedenti

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
    }
}

```

```

    finally {
        System.out.print(3);
        throw( new Exception() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 13Exception in thread "main" java.lang.Exception**
- B. 12**
- C. 1423Exception in thread "main" java.lang.Exception**
- D. Errore a tempo di compilazione**
- E. Nessuna delle precedenti**

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC**
- B. CCC**
- C. Errore a tempo di compilazione.**
- D. Errore a tempo di esecuzione.**
- E. Risultato diverso dai precedenti.**
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 51

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire **private** la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire **final** la variabile *i*.
- D. Definire **protected** la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

3. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo *m*?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 14444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}
```

```
class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }
}
```

```

    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

10. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {

```



```

        System.out.print(0);
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

11. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}
}

```

```

public static String print() {
    return name + " " + age + " ";
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
        }
    }
}

```

```

        q();
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new Exception() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 13Exception in thread "main" java.lang.Exception
 - B. 12
 - C. 1423Exception in thread "main" java.lang.Exception
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

15. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
 - B. m = a;
 - C. m = (Error []) q;
 - D. q = a;
 - E. q = (Boolean []) m;
-

16. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
 - D. Un array non ha un'unica superclasse
 - E. Una classe non può essere dichiarata abstract o final
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 52

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

2. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.

- B. Definire **private** la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire **final** la variabile `i`.
- D. Definire **protected** la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

3. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
}
```

```

static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

6. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.

E. Nessuna delle precedenti.

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 14444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

8. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }
}

```

```

public static void m() {
    Currency.getCurrency(50);
    Currency.getCurrency(100);
}

public static void main(String[] argv) {
    Currency.m();
}
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l) {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
    }
}
```

```
System.out.print(e.print());
System.out.print(j.print());
}
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 53

1. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print () {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print () {
        return name + " " + age + " ";
    }
}
```

```

    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print (e.print ());
        System.out.print (j.print ());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print (
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC

- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. q: Date le dichiarazioni:

```
Object [] a;  
Error [] m;  
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        finally {  
            System.out.print(3);  
            throw( new Exception() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
        }  
        catch( Exception t ) {  
            System.out.print(4);  
        }  
        catch( MyExc2 x ) {  
            System.out.print(5);  
        }  
        finally {  
            throw( new MyExc2() );  
        }  
    }  
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            p();  
        }  
        catch( MyExc2 e ) {  
            System.out.print(1);  
        }  
        catch( MyExc3 z ) {  
        }  
        catch( MyExc1 k ) {  
        }  
        finally {  
            System.out.print(2);  
        }  
    }  
    static void p() throws Exception {  
        try {  
            throw( new Exception() );  
        }  
        catch( MyExc3 c ) {  
            System.out.print(3);  
        }  
        catch( Exception j ) {  
            System.out.print(4);  
        }  
        finally {  
            throw( new MyExc2() );  
        }  
    }  
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {  
    X(long l){  
        System.out.print("X");  
    }  
    abstract void z();  
}
```

```

class Y extends X {
    void z() {}
    Y() {
        System.out.print ("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print (1);
        } else {
            System.out.print (0);
        }
        if(s1 == s3) {
            System.out.print (1);
        } else {
            System.out.print (0);
        }
        if("abcd" == s4) {
            System.out.print (1);
        } else {
            System.out.print (0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

9. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print (e.print ());
        System.out.print (j.print ());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 14444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

12. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }
}
```

```
}

public static void m() {
    Currency.getCurrency(50);
    Currency.getCurrency(100);
}

public static void main(String[] argv) {
    Currency.m();
}
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}
```

```

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

15. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri

D. Un array non ha un'unica superclasse

E. Una classe non può essere dichiarata abstract o final

16. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.
- D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 54

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
    }
}
```

```
finally {
    System.out.print(3);
    throw( new Exception() );
}
}
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```



```
}
```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

4. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
 - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 - C. Un'istanza della classe I può essere costruita con sf new A.I().
 - D. Un'istanza della classe I può essere costruita con new A().new I().
 - E. Nessuna delle precedenti.
-

5. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
 - B. m = a;
 - C. m = (Error []) q;
 - D. q = a;
 - E. q = (Boolean []) m;
-

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
```

```
try {
    p();
}
catch( MyExc2 e ) {
    System.out.print(1);
}
catch( MyExc3 z ) {
}
catch( MyExc1 k ) {
}
finally {
    System.out.print(2);
}
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}
```

- A. 412
 - B. 4
 - C. 442
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}
```

```

    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

9. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

10. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

11. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;

- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}
```

```
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 1444444... (ciclo infinito)
 - B. 1463Exception in thread "main" MyExc1
 - C. Errore a tempo di compilazione
 - D. 143Exception in thread "main" MyExc1
 - E. Nessuna delle precedenti
-

16. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
 - B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - C. Definire final la variabile i.
 - D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 55

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
```

```
        throw( new MyExc2() );
    }
    finally {
        System.out.print(3);
        throw( new MyExc1() );
    }
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}
```

- A. 14444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
```

```

        System.out.print (
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print ("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print ("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }
}

```

```

    }
    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print (e.print());
        System.out.print (j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        }
    }
}

```

```

    } else {
        System.out.print (0);
    }
}
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X() {
        System.out.print ("X");
    }
    abstract void z ();
}

class Y extends X {
    void z () {}
    Y() {
        System.out.print ("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print (1);
            q();
            System.out.print (2);
        }
        finally {
            System.out.print (3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print (4);
        }
        catch( MyExc2 x ) {
            System.out.print (5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

10. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri

- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

11. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

12. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
```

```

    }
    finally {
        System.out.print (2);
    }
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print (3);
    }
    catch( Exception j ) {
        System.out.print (4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 412
 - B. 4
 - C. 442
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

15. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
 - B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - C. Definire final la variabile i.
 - D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - E. Nessuna delle precedenti.
-

16. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
 - B. m = a;
 - C. m = (Error []) q;
 - D. q = a;
 - E. q = (Boolean []) m;
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 56

1. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

2. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

3. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata **abstract** o **final**

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}
}
```

```
public static String print() {
    return name + " " + age + " ";
}

}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
 - B. A S 20 B J 25
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X() {
        System.out.print ("X");
    }
    abstract void z ();
}

class Y extends X {
    void z () {}
    Y() {
        System.out.print ("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print () {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print () {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print (e.print ());
        System.out.print (j.print ());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print (1);
            q();
            System.out.print (2);
        }
        finally {
            System.out.print (3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print (4);
        }
        catch( MyExc2 x ) {
            System.out.print (5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

12. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p ();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {

```

```

}
        finally {
            System.out.print (2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print (3);
        }
        catch( Exception j ) {
            System.out.print (4);
        }
        finally {
            throw( new MyExc2 () );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `A`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
- C. Un'istanza della classe `I` può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe `I` può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q ();

```

```

}
catch( MyExc2 h ) {
    System.out.print(2);
    throw( new MyExc2() );
}
finally {
    System.out.print(3);
    throw( new MyExc1() );
}
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 14444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

16. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
 - B. 6
 - C. 7
 - D. 8
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 57

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print () {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print () {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
 B. A S 20 B J 25
 C. Errore a tempo di compilazione.
 D. Errore a tempo di esecuzione.
 E. Nessuna delle precedenti.
-
2. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
```

```
static Integer value = new Integer();

static void getCurrency(Integer v) {
    Currency c = new Currency();
    c.value = v;
}

public static void m() {
    Currency.getCurrency(50);
    Currency.getCurrency(100);
}

public static void main(String[] argv){
    Currency.m();
}
}
```

- A. 5
 B. 6
 C. 7
 D. 8
 E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
```

```

}
catch( MyExc3 k ) {
    System.out.print(4);
    throw( new MyExc3() );
}
catch( MyExc1 t ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
}
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

7. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
 - D. Un array non ha un'unica superclasse
 - E. Una classe non può essere dichiarata abstract o final
-

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){ }
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

10. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
 - B. Definire private la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - C. Definire final la variabile `i`.
 - D. Definire protected la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
 - E. Nessuna delle precedenti.
-

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().

E. Nessuna delle precedenti.

13. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

```

```

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
 - B. A S 20 B J 25
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

```

```

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

16. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
 - B. m = a;
 - C. m = (Error []) q;
 - D. q = a;
 - E. q = (Boolean []) m;
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 58

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
 B. A S 20 B J 25
 C. Errore a tempo di compilazione.
 D. Errore a tempo di esecuzione.
 E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

- e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

- indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
 B. y = (B1) b;
 C. b = (B2) y;
 D. y = (B1) h;
 E. Nessuno dei precedenti

3. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
 B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
 D. Un array non ha un'unica superclasse
 E. Una classe non può essere dichiarata **abstract** o **final**

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p ();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
```



```

    }
    catch( MyExc1 k ) {
    }
    finally {
        System.out.print (2);
    }
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print (3);
    }
    catch( Exception j ) {
        System.out.print (4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print (1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print (2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print (3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print (4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {

```

```

        System.out.print (5);
    }
    finally {
        System.out.print (6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X() {
        System.out.print ("X");
    }
    abstract void z ();
}

```

```

class Y extends X {
    void z () {}
    Y() {
        System.out.print ("Y");
    }
}

```

```

public static void main(String[] args) {
    X x = new Y();
}
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {

```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s1 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if("abcd" == s4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
}

```

```

    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

11. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

12. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {

```

```

        throw( new MyExc2() );
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

E. Nessuna delle precedenti.

16. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 59

1. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

2. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

3. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
```

```
Currency c = new Currency();
c.value = v;
}
```

```
public static void m() {
    Currency.getCurrency(50);
    Currency.getCurrency(100);
}
```

```
public static void main(String[] argv){
    Currency.m();
}
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}
```

```
class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}
```

```
public static void main(String[] args){
    X x = new Y();
}
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
```

```
try {
    throw( new Exception() );
}
catch( MyExc3 c ) {
    System.out.print(3);
}
catch( Exception j ) {
    System.out.print(4);
}
finally {
    throw( new MyExc2() );
}
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

8. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

9. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
}

```

```

Y() {
    System.out.print("Y");
}

public static void main(String[] args) {
    X x = new Y();
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
```

```
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

14. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

15. Qual'è l'output di questo codice?

```
class MyExcl extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
            }
```

```

    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }
}

```

```

    }
    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 60

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.

- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

3. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = (Error []) a;`
- B. `m = a;`
- C. `m = (Error []) q;`
- D. `q = a;`
- E. `q = (Boolean []) m;`

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}
```

```

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
    }
}

```

```

}
catch( MyExc2 e ) {
    System.out.print(1);
}
catch( MyExc3 z ) {
}
catch( MyExc1 k ) {
}
finally {
    System.out.print(2);
}
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

```
}  
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        finally {  
            System.out.print(3);  
            throw( new Exception() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
        }  
        catch( Exception t ) {  
            System.out.print(4);  
        }  
        catch( MyExc2 x ) {  
            System.out.print(5);  
        }  
        finally {  
            throw( new MyExc2() );  
        }  
    }  
}
```

- A. 13Exception in thread "main" java.lang.Exception
 - B. 12
 - C. 1423Exception in thread "main" java.lang.Exception
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {  
    String name = "A";  
    void m();  
}  
  
abstract class B implements A {  
    String name = "B";  
    abstract B g();  
}  
  
class C extends B {  
    String name = "C";  
    public void m() {  
        System.out.print(  
            ((A)this).name +  
            ((B)this).name +  
            ((C)this).name);  
    }  
    public B g() {return this;}  
    public static void main(String[] args){  
        new C().m();  
    }  
}
```

- A. ABC
 - B. CCC
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Risultato diverso dai precedenti.
-

10. Dato un file contenente il seguente codice:

```
package pk;  
  
public class A {  
    public final class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
 - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 - C. Un'istanza della classe I può essere costruita con sf new A.I().
 - D. Un'istanza della classe I può essere costruita con new A().new I().
 - E. Nessuna delle precedenti.
-

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {

```

```

try {
    throw( new MyExc3() );
}
catch( MyExc3 k ) {
    System.out.print(4);
    throw( new MyExc3() );
}
catch( MyExc1 t ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
}
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

13. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010

14. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

15. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri

D. Un array non ha un'unica superclasse

- E. Una classe non può essere dichiarata `abstract` o `final`

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 61

1. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

2. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `A`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
- C. Un'istanza della classe `I` può essere costruita con `sf new A.I()`.

- D. Un'istanza della classe `I` può essere costruita con `new A().new I()`.

- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
```

```
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

6. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
```

```
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
 - B. 4
 - C. 442
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}
```

```
    }
    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

13. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
 - B. y = (B1) b;
 - C. b = (B2) y;
 - D. y = (B1) h;
 - E. Nessuno dei precedenti
-

14. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
 - D. Un array non ha un'unica superclasse
 - E. Una classe non può essere dichiarata abstract o final
-

15. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile `i`.
- D. Definire protected la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);

```

```

        q();
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new Exception() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 13Exception in thread "main" java.lang.Exception
 - B. 12
 - C. 1423Exception in thread "main" java.lang.Exception
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 62

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
}
```

```
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

4. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC

- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile `i`.
- D. Definire protected la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

8. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args) {
```

```
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

```

}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public final class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

14. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

16. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];

```

```

void p() {
    String [] a2;
    String s2;
    a2 = new String [4];
    s2 = "abc";
    m(a2, s2);
}

void m(String [] a1, String s1) {
    String s3;
    String s4 = "abcd";
    s3 = "abc";
    if(a3 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s1 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if("abcd" == s4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 111
 - B. 011
 - C. 100
 - D. 010
 - E. 101
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 63

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2 ();
b = new B2 ();
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

3. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
```

```

        System.out.print (0);
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l) {
        System.out.print ("X");
    }
    abstract void z ();
}

class Y extends X {
    void z () {}
    Y () {
        System.out.print ("Y");
    }

    public static void main (String [] args) {
        X x = new Y ();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m ();
}

abstract class B implements A {
    String name = "B";
    abstract B g ();
}

class C extends B {
    String name = "C";
    public void m () {

```

```

        System.out.print (
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g () {return this;}
    public static void main (String [] args) {
        new C ().m ();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

6. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main (String [] argv) {
        try {
            System.out.print (1);
            q ();
        }
        catch ( MyExc2 h ) {
            System.out.print (2);
            throw ( new MyExc2 () );
        }
        finally {
            System.out.print (3);
            throw ( new MyExc1 () );
        }
    }
    static void q () {
        try {
            throw ( new MyExc3 () );
        }
        catch ( MyExc3 k ) {
            System.out.print (4);

```

```

        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

10. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

11. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}

```


- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}
```

```
class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}
```

```
class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
```

```
        throw( new MyExc2() );
    }
}
```

- A. 412
 - B. 4
 - C. 442
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

15. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con `new A.I()`.
- D. Un'istanza della classe I può essere costruita con `new A().new I()`.

E. Nessuna delle precedenti.

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 64

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }
}
```

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public static void main(String[] args){
    X x = new Y();
}
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
}

```

```

B g() {return this;}
public static void main(String[] args){
    new C().m();
}
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

6. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

7. Quale output si ottiene invocando il metodo `p`?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
    if(s1 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if("abcd" == s4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class Bl {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

13. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){
    }
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
 - B. Y
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}
```

```
class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
 - B. A S 20 B J 25
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

16. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
 - B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
 - C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
 - D. Un array non ha un'unica superclasse
 - E. Una classe non può essere dichiarata abstract o final
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 65

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l) {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
```

```
((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

3. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

4. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

5. q: Date le dichiarazioni:

```

Object [] a;
Error [] m;
Boolean [] q;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire **private** la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire **final** la variabile i.

- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `A`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
- C. Un'istanza della classe `I` può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe `I` può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {

```

```

        System.out.print(3);
        throw( new MyExc1() );
    }
}
static void q() {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 k ) {
        System.out.print(4);
        throw( new MyExc3() );
    }
    catch( MyExc1 t ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

```

```

}
class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }
    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 66

1. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire `private` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire `final` la variabile `i`.
- D. Definire `protected` la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}
```

```
class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){
        Y(){
            System.out.print("Y");
        }

        public static void main(String[] args){
            X x = new Y();
        }
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo codice?


```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

10. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita con sf new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

11. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
```

```

        System.out.print(0);
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

13. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }
}

```

```

public static void m() {
    Currency.getCurrency(50);
    Currency.getCurrency(100);
}

```

```

public static void main(String[] argv) {
    Currency.m();
}
}

```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

15. q: Date le dichiarazioni:

```
Object [] a;  
Error [] m;  
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
 - B. m = a;
 - C. m = (Error []) q;
 - D. q = a;
 - E. q = (Boolean []) m;
-

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class A1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            q();  
        }  
        catch( MyExc2 h ) {  
            System.out.print(2);  
            throw( new MyExc2() );  
        }  
    }  
}
```

```
    }  
    finally {  
        System.out.print(3);  
        throw( new MyExc1() );  
    }  
}  
static void q() {  
    try {  
        throw( new MyExc3() );  
    }  
    catch( MyExc3 k ) {  
        System.out.print(4);  
        throw( new MyExc3() );  
    }  
    catch( MyExc1 t ) {  
        System.out.print(5);  
    }  
    finally {  
        System.out.print(6);  
    }  
}  
}
```

- A. 1444444... (ciclo infinito)
 - B. 1463Exception in thread "main" MyExc1
 - C. Errore a tempo di compilazione
 - D. 143Exception in thread "main" MyExc1
 - E. Nessuna delle precedenti
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 67

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

2. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
```

```
String s4 = "abcd";
s3 = "abc";
if(a3 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s1 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if("abcd" == s4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(long l){
        System.out.print("X");
    }
    abstract void z();
}
```

```
class Y extends X {
    void z(){
    Y(){
        System.out.print("Y");
    }
    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

5. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.
- D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

E. Nessuna delle precedenti.

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. Un'istanza della classe l può essere costruita con sf new A.I().
- D. Un'istanza della classe l può essere costruita con new A().new l().
- E. Nessuna delle precedenti.

8. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

9. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
```

```
try {
    p();
}
catch( MyExc2 e ) {
    System.out.print(1);
}
catch( MyExc3 z ) {
}
catch( MyExc1 k ) {
}
finally {
    System.out.print(2);
}
}
static void p() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 c ) {
        System.out.print(3);
    }
    catch( Exception j ) {
        System.out.print(4);
    }
    finally {
        throw( new MyExc2() );
    }
}
}
```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }
}

public static void main(String[] args) {
    X x = new Y();
}
}
```

- A. X

- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri

- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
```

```

    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    static String name;
    static int age;

    Person(String n, int a) {

```

```

        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 68

1. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

2. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X(){
        System.out.print("X");
    }
    abstract void z();
}
```

```
class Y extends X {
    void z(){}
    Y(){
        System.out.print("Y");
    }

    public static void main(String[] args){
        X x = new Y();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse

E. Una classe non può essere dichiarata **abstract** o **final**

5. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
 - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
 - C. Un'istanza della classe I può essere costruita con `sf new A.I()`.
 - D. Un'istanza della classe I può essere costruita con `new A().new I()`.
 - E. Nessuna delle precedenti.
-

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
 - B. A S 20 B J 25
 - C. Errore a tempo di compilazione.
 - D. Errore a tempo di esecuzione.
 - E. Nessuna delle precedenti.
-

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 412
 - B. 4
 - C. 442
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

8. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;  
Super h;  
B1 y;  
h = new B2 ();  
b = new B2 ();  
y = new B1 ();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {  
    X(long l) {  
        System.out.print ("X");  
    }  
    abstract void z ();  
}  
  
class Y extends X {  
    void z () {}  
    Y () {  
        System.out.print ("Y");  
    }  
  
    public static void main (String[] args) {  
        X x = new Y ();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {  
    String name = "A";  
    void m ();  
}  
  
abstract class B implements A {
```

```
    String name = "B";  
    abstract B g ();  
}  
  
class C extends B {  
    String name = "C";  
    public void m () {  
        System.out.print (  
            ((A)this).name +  
            ((B)this).name +  
            ((C)this).name);  
    }  
    public B g () {return this;}  
    public static void main (String[] args) {  
        new C ().m ();  
    }  
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {  
    static String name;  
    static int age;  
  
    Person (String n, int a) {  
        name = n;  
        age = a;  
    }  
  
    public static String print () {  
        return name + " ";  
    }  
}  
  
class Student extends Person {  
    Student (String n, int a) {super (n, a);}  
  
    public static String print () {  
        return name + " " + age + " ";  
    }  
}  
  
class Test {  
    public static void main (String[] args) {  
        Person e = new Student ("A S", 20);  
        Student j = new Student ("B J", 25);  
        System.out.print (e.print ());  
        System.out.print (j.print ());  
    }  
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

13. Date le seguenti classi:

```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire **private** la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

- C. Definire **final** la variabile `i`.
- D. Definire **protected** la variabile `i` e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
```

```

        System.out.print(1);
        q();
        System.out.print(2);
    }
    finally {
        System.out.print(3);
        throw( new Exception() );
    }
}
static void q() throws Exception {
    try {
    }
    catch( Exception t ) {
        System.out.print(4);
    }
    catch( MyExc2 x ) {
        System.out.print(5);
    }
    finally {
        throw( new MyExc2() );
    }
}
}

```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

16. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```

public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv){
        Currency.m();
    }
}

```

- A. 5
 - B. 6
 - C. 7
 - D. 8
 - E. Nessuna delle precedenti.
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 69

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }
}

public static void main(String[] args) {
    X x = new Y();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +

```

```
((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Quale output si ottiene invocando il metodo p?

```
class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
```

```
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

```

    catch( MyExc1 t ) {
        System.out.print (5);
    }
    finally {
        System.out.print (6);
    }
}
}

```

- A. 1444444... (ciclo infinito)
- B. 1463Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 143Exception in thread "main" MyExc1
- E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l){
        System.out.print ("X");
    }
    abstract void z ();
}

class Y extends X {
    void z () {}
    Y () {
        System.out.print ("Y");
    }

    public static void main(String[] args) {
        X x = new Y ();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }
}

```

```

    public String print () {
        return name + " ";
    }
}

class Student extends Person {
    Student (String n, int a) {super (n,a);}

    public String print () {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main (String[] args) {
        Person e = new Student ("A S", 20);
        Student j = new Student ("B J", 25);
        System.out.print (e.print ());
        System.out.print (j.print ());
    }
}

```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Date le seguenti classi:

```

class A {
    Integer i = 6;
}
class B extends A {
}

```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire **private** la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire **final** la variabile *i*.
- D. Definire **protected** la variabile *i* e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

10. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi

- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata `abstract` o `final`

11. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo `m`?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

- A. 5
- B. 6
- C. 7
- D. 8
- E. Nessuna delle precedenti.

12. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `A`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
- C. Un'istanza della classe `I` può essere costruita con `sf new A.I()`.
- D. Un'istanza della classe `I` può essere costruita con `new A().new I()`.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args){
        new C().m();
    }
}
```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

14. Date le dichiarazioni:

```
class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}
```

e le inizializzazioni di variabile:

```
B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (B2) h;`
- B. `y = (B1) b;`
- C. `b = (B2) y;`
- D. `y = (B1) h;`
- E. Nessuno dei precedenti

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
    }
}
```

```
    }
    finally {
        throw( new MyExc2() );
    }
}
}
```

- A. 412
 - B. 4
 - C. 442
 - D. Errore a tempo di compilazione
 - E. Nessuna delle precedenti
-

16. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
 - B. m = a;
 - C. m = (Error []) q;
 - D. q = a;
 - E. q = (Boolean []) m;
-

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 70

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public abstract class X {
    X() {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}
```

- A. X
 B. Y
 C. Errore a tempo di compilazione.
 D. Errore a tempo di esecuzione.
 E. Nessuna delle precedenti.
-
2. Quanti oggetti *al massimo* possono essere deallocati all'uscita del metodo m?

```
public class Currency {
    String name = new String();
    static Integer value = new Integer();

    static void getCurrency(Integer v) {
        Currency c = new Currency();
        c.value = v;
    }

    public static void m() {
        Currency.getCurrency(50);
        Currency.getCurrency(100);
    }

    public static void main(String[] argv) {
        Currency.m();
    }
}
```

```
}
}
```

- A. 5
 B. 6
 C. 7
 D. 8
 E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    void m() {
        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}
```

- A. ABC
 B. CCC
 C. Errore a tempo di compilazione.
 D. Errore a tempo di esecuzione.
 E. Risultato diverso dai precedenti.

4. Date le seguenti classi:


```
class A {
    Integer i = 6;
}
class B extends A {
}
```

Quale delle seguenti proposte di cambiamento del codice precedente è la migliore rispetto all'incapsulamento?

- A. Le classi sono perfettamente incapsulate.
- B. Definire private la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- C. Definire final la variabile i.
- D. Definire protected la variabile i e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
- E. Nessuna delle precedenti.

5. q: Date le dichiarazioni:

```
Object [] a;
Error [] m;
Boolean [] q;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = (Error []) a;
- B. m = a;
- C. m = (Error []) q;
- D. q = a;
- E. q = (Boolean []) m;

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    String name;
    int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}
}
```

```
public String print() {
    return name + " " + age + " ";
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

- A. A S B J 25
- B. A S 20 B J 25
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        finally {
            System.out.print(3);
            throw( new Exception() );
        }
    }
    static void q() throws Exception {
        try {
        }
        catch( Exception t ) {
            System.out.print(4);
        }
        catch( MyExc2 x ) {
            System.out.print(5);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

- A. 13Exception in thread "main" java.lang.Exception
- B. 12
- C. 1423Exception in thread "main" java.lang.Exception

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class A1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc2 h ) {
            System.out.print(2);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 k ) {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc1 t ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}
```

A. 1444444... (ciclo infinito)

B. 1463Exception in thread "main" MyExc1

C. Errore a tempo di compilazione

D. 143Exception in thread "main" MyExc1

E. Nessuna delle precedenti

9. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public final class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. Un'istanza della classe I può essere costruita solo all'interno della classe A.

B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

C. Un'istanza della classe I può essere costruita con sf new A.I().

D. Un'istanza della classe I può essere costruita con new A().new I().

E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Person {
    static String name;
    static int age;

    Person(String n, int a) {
        name = n;
        age = a;
    }

    public static String print() {
        return name + " ";
    }
}

class Student extends Person {
    Student(String n, int a) {super(n,a);}

    public static String print() {
        return name + " " + age + " ";
    }
}

class Test {
    public static void main(String[] args){
        Person e = new Student("A S", 20);
        Student j = new Student("B J", 25);
        System.out.print(e.print());
        System.out.print(j.print());
    }
}
```

A. A S B J 25

B. A S 20 B J 25

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public abstract class X {
    X(long l) {
        System.out.print("X");
    }
    abstract void z();
}

class Y extends X {
    void z() {}
    Y() {
        System.out.print("Y");
    }

    public static void main(String[] args) {
        X x = new Y();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Dire quale delle seguenti affermazioni è vera:

- A. La lunghezza di un array può essere variata dopo la sua costruzione
- B. I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- C. Nell'overloading due metodi possono avere lo stesso nome e lo stesso numero di parametri
- D. Un array non ha un'unica superclasse
- E. Una classe non può essere dichiarata abstract o final

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface A {
    String name = "A";
    void m();
}

abstract class B implements A {
    String name = "B";
    abstract B g();
}

class C extends B {
    String name = "C";
    public void m() {

```

```

        System.out.print(
            ((A)this).name +
            ((B)this).name +
            ((C)this).name);
    }
    public B g() {return this;}
    public static void main(String[] args) {
        new C().m();
    }
}

```

- A. ABC
- B. CCC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Risultato diverso dai precedenti.

14. Quale output si ottiene invocando il metodo p?

```

class G {
    private String [] a3 = new String [4];
    void p() {
        String [] a2;
        String s2;
        a2 = new String [4];
        s2 = "abc";
        m(a2, s2);
    }
    void m(String [] a1, String s1) {
        String s3;
        String s4 = "abcd";
        s3 = "abc";
        if(a3 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s1 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if("abcd" == s4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 011
- C. 100
- D. 010
- E. 101

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            p();
        }
        catch( MyExc2 e ) {
            System.out.print(1);
        }
        catch( MyExc3 z ) {
        }
        catch( MyExc1 k ) {
        }
        finally {
            System.out.print(2);
        }
    }
    static void p() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 c ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            System.out.print(4);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}

```

- A. 412
- B. 4
- C. 442
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

16. Date le dichiarazioni:

```

class B1 extends Super {...}
class Super extends Object {...}
class B2 extends Super {...}

```

e le inizializzazioni di variabile:

```

B2 b;
Super h;
B1 y;
h = new B2();
b = new B2();
y = new B1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (B2) h;
- B. y = (B1) b;
- C. b = (B2) y;
- D. y = (B1) h;
- E. Nessuno dei precedenti

Prova n. 1

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1

Prova d'esame

prof. Piero A. Bonatti

16-07-2015

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

1	■				
2				■	
3			■		
4			■		
5			■		
6		■			
7			■		
8	■				
9				■	
10		■			
11		■			
12		■			
13					■
14	■				
15					■
16	■				

Risultato prova n. 1:

1	■				
2				■	
3	■				
4		■			
5			■		
6			■		
7			■		
8	■				
9				■	
10		■			
11					■
12		■			
13	■				
14					■
15			■		
16		■			

Risultato prova n. 2:

1					■
2	■				
3		■			
4	■				
5	■				
6		■			
7			■		
8			■		
9			■		
10		■			
11		■			
12			■		
13	■				
14				■	
15					■
16				■	

Risultato prova n. 3:

1			■		
2				■	
3					■
4			■		
5		■			
6			■		
7	■				
8			■		
9		■			
10		■			
11		■			
12					■
13				■	
14	■				
15	■				
16	■				

Risultato prova n. 4:

1		■			
2					■
3			■		
4	■				
5				■	
6		■			
7	■				
8			■		
9					■
10			■		
11		■			
12	■				
13		■			
14				■	
15			■		
16	■				

Risultato prova n. 5:

1		■			
2			■		
3			■		
4		■			
5	■				
6			■		
7		■			
8	■				
9				■	
10			■		
11	■				
12	■				
13		■			
14				■	
15					■
16					■

Risultato prova n. 6:

1	■			
2		■		
3		■		
4			■	
5			■	
6				■
7	■			
8			■	
9			■	
10				■
11	■			
12		■		
13		■		
14	■			
15			■	
16			■	

Risultato prova n. 7:

1			■		
2			■		
3		■			
4					■
5	■				
6				■	
7	■				
8				■	
9	■				
10			■		
11		■			
12		■			
13					■
14			■		
15	■				
16		■			

Risultato prova n. 8:

1	■				
2	■				
3		■			
4					■
5		■			
6			■		
7				■	
8			■		
9			■		
10	■				
11	■				
12			■		
13					■
14		■			
15				■	
16		■			

Risultato prova n. 9:

1			■		
2		■			
3		■			
4	■				
5		■			
6		■			
7	■				
8				■	
9			■		
10				■	
11	■				
12			■		
13					■
14			■		
15					■
16	■				

Risultato prova n. 10:

1				■	
2			■		
3			■		
4				■	
5		■			
6	■				
7		■			
8					■
9					■
10		■			
11	■				
12		■			
13			■		
14	■				
15	■				
16			■		

Risultato prova n. 11:

1			■		
2			■		
3		■			
4	■				
5	■				
6					■
7				■	
8	■				
9		■			
10		■			
11		■			
12	■				
13			■		
14				■	
15					■
16			■		

Risultato prova n. 12:

1	■				
2					■
3		■			
4			■		
5	■				
6				■	
7			■		
8		■			
9	■				
10	■				
11			■		
12					■
13				■	
14			■		
15		■			
16		■			

Risultato prova n. 13:

1				■	
2			■		
3			■		
4			■		
5	■				
6		■			
7	■				
8					■
9		■			
10					■
11		■			
12	■				
13		■			
14				■	
15			■		
16	■				

Risultato prova n. 14:

1	■				
2		■			
3					■
4		■			
5			■		
6			■		
7	■				
8		■			
9		■			
10				■	
11					■
12				■	
13	■				
14			■		
15	■				
16			■		

Risultato prova n. 15:

1			■		
2	■				
3		■			
4					■
5			■		
6					■
7		■			
8			■		
9	■				
10		■			
11	■				
12		■			
13			■		
14				■	
15				■	
16	■				

Risultato prova n. 16:

1		■			
2	■				
3	■				
4			■		
5	■				
6					■
7		■			
8			■		
9					■
10			■		
11	■				
12			■		
13		■			
14				■	
15		■			
16				■	

Risultato prova n. 17:

1				■	
2					■
3		■			
4	■				
5	■				
6		■			
7			■		
8	■				
9					■
10				■	
11			■		
12			■		
13			■		
14		■			
15		■			
16	■				

Risultato prova n. 18:

1		■			
2					■
3	■				
4				■	
5					■
6			■		
7	■				
8		■			
9			■		
10			■		
11		■			
12			■		
13		■			
14				■	
15	■				
16	■				

Risultato prova n. 19:

1		■			
2					■
3		■			
4	■				
5			■		
6		■			
7	■				
8				■	
9		■			
10				■	
11	■				
12	■				
13			■		
14					■
15			■		
16			■		

Risultato prova n. 20:

1			■		
2	■				
3			■		
4	■				
5		■			
6		■			
7					■
8				■	
9	■				
10		■			
11	■				
12					■
13			■		
14		■			
15			■		
16				■	

Risultato prova n. 21:

1	■				
2	■				
3			■		
4	■				
5		■			
6					■
7	■				
8					■
9			■		
10		■			
11		■			
12				■	
13			■		
14			■		
15		■			
16				■	

Risultato prova n. 22:

1					■
2		■			
3	■				
4		■			
5		■			
6				■	
7					■
8		■			
9	■				
10			■		
11			■		
12	■				
13	■				
14				■	
15			■		
16			■		

Risultato prova n. 23:

1	■				
2				■	
3			■		
4			■		
5	■				
6					■
7	■				
8		■			
9					■
10		■			
11			■		
12	■				
13		■			
14			■		
15		■			
16				■	

Risultato prova n. 24:

1			■		
2					■
3		■			
4					■
5		■			
6			■		
7			■		
8	■				
9				■	
10				■	
11	■				
12	■				
13	■				
14		■			
15			■		
16		■			

Risultato prova n. 25:

1	■				
2			■		
3				■	
4			■		
5					■
6					■
7		■			
8			■		
9	■				
10	■				
11			■		
12		■			
13		■			
14	■				
15				■	
16		■			

Risultato prova n. 26:

1	■				
2		■			
3					■
4	■				
5		■			
6				■	
7			■		
8	■				
9				■	
10		■			
11		■			
12			■		
13			■		
14			■		
15	■				
16					■

Risultato prova n. 27:

1			■		
2			■		
3			■		
4	■				
5		■			
6				■	
7		■			
8				■	
9					■
10					■
11	■				
12		■			
13	■				
14		■			
15			■		
16	■				

Risultato prova n. 28:

1				■	
2		■			
3	■				
4			■		
5				■	
6	■				
7	■				
8					■
9	■				
10			■		
11		■			
12			■		
13		■			
14		■			
15			■		
16					■

Risultato prova n. 29:

1			■		
2	■				
3					■
4		■			
5				■	
6				■	
7		■			
8	■				
9		■			
10			■		
11		■			
12			■		
13			■		
14	■				
15					■
16	■				

Risultato prova n. 30:

1			■		
2	■				
3					■
4	■				
5			■		
6			■		
7	■				
8	■				
9		■			
10			■		
11		■			
12					■
13		■			
14				■	
15				■	
16		■			

Risultato prova n. 31:

1				■	
2	■				
3			■		
4				■	
5			■		
6			■		
7		■			
8		■			
9	■				
10	■				
11					■
12		■			
13			■		
14	■				
15					■
16		■			

Risultato prova n. 32:

1		■			
2		■			
3		■			
4	■				
5					■
6					■
7	■				
8		■			
9			■		
10			■		
11	■				
12				■	
13			■		
14				■	
15			■		
16	■				

Risultato prova n. 33:

1		■			
2		■			
3	■				
4	■				
5				■	
6			■		
7		■			
8					■
9				■	
10	■				
11			■		
12		■			
13			■		
14			■		
15					■
16	■				

Risultato prova n. 34:

1			■		
2	■				
3	■				
4		■			
5				■	
6		■			
7		■			
8		■			
9			■		
10	■				
11					■
12					■
13	■				
14				■	
15			■		
16			■		

Risultato prova n. 35:

1					■
2		■			
3			■		
4					■
5			■		
6	■				
7		■			
8				■	
9			■		
10			■		
11				■	
12	■				
13		■			
14	■				
15	■				
16		■			

Risultato prova n. 36:

1			■		
2		■			
3			■		
4	■				
5		■			
6	■				
7		■			
8	■				
9					■
10	■				
11					■
12				■	
13			■		
14		■			
15				■	
16			■		

Risultato prova n. 37:

1	■				
2			■		
3		■			
4		■			
5		■			
6				■	
7			■		
8				■	
9	■				
10	■				
11			■		
12					■
13		■			
14			■		
15	■				
16					■

Risultato prova n. 38:

1	■				
2		■			
3		■			
4			■		
5			■		
6			■		
7		■			
8			■		
9					■
10	■				
11				■	
12				■	
13		■			
14					■
15	■				
16	■				

Risultato prova n. 39:

1	■				
2					■
3				■	
4		■			
5	■				
6			■		
7			■		
8			■		
9	■				
10		■			
11					■
12	■				
13		■			
14				■	
15			■		
16		■			

Risultato prova n. 40:

1			■		
2					■
3		■			
4		■			
5	■				
6	■				
7		■			
8					■
9	■				
10				■	
11			■		
12	■				
13				■	
14			■		
15		■			
16			■		

Risultato prova n. 41:

1			■		
2		■			
3	■				
4	■				
5				■	
6		■			
7		■			
8	■				
9					■
10			■		
11					■
12			■		
13	■				
14				■	
15			■		
16		■			

Risultato prova n. 42:

1		■			
2			■		
3	■				
4		■			
5					■
6			■		
7					■
8			■		
9		■			
10			■		
11	■				
12	■				
13				■	
14				■	
15		■			
16	■				

Risultato prova n. 43:

1			■		
2			■		
3	■				
4					■
5		■			
6	■				
7		■			
8			■		
9	■				
10		■			
11			■		
12					■
13	■				
14				■	
15		■			
16				■	

Risultato prova n. 44:

1	■				
2				■	
3	■				
4					■
5		■			
6			■		
7					■
8		■			
9			■		
10			■		
11	■				
12	■				
13		■			
14		■			
15				■	
16			■		

Risultato prova n. 45:

1		■			
2	■				
3				■	
4	■				
5				■	
6			■		
7					■
8	■				
9		■			
10			■		
11					■
12	■				
13		■			
14		■			
15			■		
16			■		

Risultato prova n. 46:

1			■		
2			■		
3				■	
4				■	
5					■
6			■		
7		■			
8		■			
9		■			
10	■				
11					■
12	■				
13		■			
14	■				
15	■				
16			■		

Risultato prova n. 47:

1			■		
2	■				
3				■	
4			■		
5			■		
6		■			
7					■
8	■				
9				■	
10	■				
11		■			
12			■		
13					■
14	■				
15		■			
16		■			

Risultato prova n. 48:

1		■			
2	■				
3	■				
4			■		
5		■			
6			■		
7		■			
8				■	
9					■
10			■		
11					■
12				■	
13		■			
14			■		
15	■				
16	■				

Risultato prova n. 49:

1	■			
2		■		
3			■	
4			■	
5				■
6			■	
7		■		
8	■			
9		■		
10	■			
11		■		
12	■			
13				■
14			■	
15			■	
16			■	

Risultato prova n. 50:

1	■				
2		■			
3		■			
4			■		
5	■				
6					■
7		■			
8				■	
9			■		
10		■			
11	■				
12					■
13			■		
14				■	
15	■				
16			■		

Risultato prova n. 51:

1	■				
2		■			
3	■				
4	■				
5			■		
6				■	
7			■		
8		■			
9			■		
10	■				
11				■	
12					■
13		■			
14		■			
15					■
16			■		

Risultato prova n. 52:

1	■				
2		■			
3	■				
4	■				
5				■	
6	■				
7			■		
8		■			
9				■	
10					■
11			■		
12		■			
13					■
14			■		
15			■		
16		■			

Risultato prova n. 53:

1	■				
2				■	
3			■		
4				■	
5	■				
6	■				
7		■			
8			■		
9		■			
10			■		
11	■				
12					■
13					■
14		■			
15			■		
16		■			

Risultato prova n. 54:

1			■		
2			■		
3	■				
4			■		
5		■			
6		■			
7					■
8				■	
9				■	
10			■		
11		■			
12	■				
13					■
14	■				
15		■			
16	■				

Risultato prova n. 55:

1	■				
2		■			
3			■		
4			■		
5	■				
6					■
7			■		
8					■
9		■			
10				■	
11	■				
12		■			
13	■				
14				■	
15			■		
16		■			

Risultato prova n. 56:

1		■			
2		■			
3			■		
4	■				
5	■				
6	■				
7			■		
8			■		
9			■		
10		■			
11				■	
12				■	
13		■			
14					■
15					■
16	■				

Risultato prova n. 57:

1					■
2	■				
3			■		
4	■				
5			■		
6					■
7		■			
8		■			
9			■		
10			■		
11				■	
12	■				
13	■				
14				■	
15		■			
16		■			

Risultato prova n. 58:

1		■			
2			■		
3		■			
4			■		
5	■				
6	■				
7			■		
8	■				
9				■	
10					■
11			■		
12					■
13		■			
14	■				
15				■	
16		■			

Risultato prova n. 59:

1					■
2		■			
3	■				
4		■			
5		■			
6	■				
7			■		
8				■	
9	■				
10				■	
11					■
12			■		
13		■			
14	■				
15			■		
16			■		

Risultato prova n. 60:

1		■			
2				■	
3			■		
4		■			
5			■		
6	■				
7	■				
8			■		
9					■
10		■			
11	■				
12					■
13	■				
14			■		
15		■			
16				■	

Risultato prova n. 61:

1					■
2			■		
3	■				
4		■			
5			■		
6	■				
7		■			
8			■		
9		■			
10					■
11				■	
12			■		
13				■	
14	■				
15	■				
16		■			

Risultato prova n. 62:

1					■
2	■				
3		■			
4			■		
5	■				
6	■				
7			■		
8				■	
9			■		
10		■			
11		■			
12			■		
13		■			
14	■				
15				■	
16					■

Risultato prova n. 63:

1			■		
2	■				
3			■		
4		■			
5			■		
6		■			
7		■			
8	■				
9	■				
10		■			
11				■	
12				■	
13	■				
14					■
15					■
16			■		

Risultato prova n. 64:

1			■		
2	■				
3	■				
4		■			
5	■				
6	■				
7		■			
8				■	
9				■	
10					■
11		■			
12		■			
13			■		
14			■		
15					■
16			■		

Risultato prova n. 65:

1		■			
2					■
3			■		
4	■				
5			■		
6					■
7	■				
8				■	
9	■				
10				■	
11		■			
12			■		
13		■			
14		■			
15	■				
16			■		

Risultato prova n. 66:

1			■		
2		■			
3			■		
4		■			
5		■			
6				■	
7				■	
8	■				
9	■				
10	■				
11					■
12		■			
13			■		
14			■		
15	■				
16					■

Risultato prova n. 67:

1	■			
2		■		
3				■
4			■	
5				■
6		■		
7	■			
8	■			
9			■	
10	■			
11				■
12			■	
13		■		
14			■	
15				■
16		■		

Risultato prova n. 68:

1					■
2			■		
3					■
4				■	
5		■			
6			■		
7			■		
8		■			
9		■			
10			■		
11		■			
12				■	
13	■				
14	■				
15	■				
16	■				

Risultato prova n. 69:

1					■
2		■			
3			■		
4		■			
5	■				
6		■			
7				■	
8			■		
9				■	
10					■
11			■		
12			■		
13	■				
14		■			
15	■				
16	■				

Risultato prova n. 70: