



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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

2. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. x = (Error) u;

- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

---

5. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

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

---

6. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

---

7. Qual'è l'output di questo codice?

```

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

```

```

    }
    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
}

public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

9. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

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

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

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

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

12. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

13. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

14. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

15. Qual'è l'output di questo codice?

```
String c = "I";
void f();

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC

- B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
```

```
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - 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. 2**

1. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
- E. Un array non possiede dei membri

2. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo `g`

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo `g`.

3. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`
- B. `g = (Exception) u;`
- C. `x = (Error) g;`
- D. `g = (Exception) x;`
- E. Nessuno dei precedenti

4. Quale delle seguenti non è un sovraccaricamento corretto del metodo `m` nella classe `C`?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`



E. Nessuna delle precedenti.

---

5. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

---

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. 134

B. 14

C. Errore a tempo di compilazione

D. 14Exception in thread main MyExc1

E. Nessuna delle precedenti

---

7. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

---

8. Qual'è l'output di questo codice?

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

A. 25Exception in thread main MyExc1

B. 2451

C. 2Exception in thread main MyExc1

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

9. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

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

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
```

```
String s = "Lemon";
public String type() {
    return s;
}

public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}
}
```

- A. Tree
  - B. Lemon
  - 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

12. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

13. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
```

```
protected void m(I i) {
    System.out.print(i.c);
}

protected void m(A a) {
    System.out.print(a.c);
}

protected void m(B c) {
    System.out.print(c.c);
}

public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

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

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
```

```

        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - 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 A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - 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. 3**

1. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

2. Dato un file contenente il seguente codice:

```
package pk;

public class C {
```

```
static class I {
}
}
```

Dire quale delle seguenti affermazioni è vera:

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

3. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`
- B. `g = (Exception) u;`
- C. `x = (Error) g;`
- D. `g = (Exception) x;`
- E. Nessuno dei precedenti

4. Qual'è l'output di questo codice?

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

```

}
static void q() throws Exception {
    try {
        System.out.print(2);
        throw( new Exception() );
    }
    catch( MyExc2 r ) {
        System.out.print(3);
    }
    catch( MyExc1 c ) {
        System.out.print(4);
    }
    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- 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 MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

6. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {

```

```

        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

9. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

10. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

11. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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



- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

12. Quale delle seguenti non è un sovraccaricamento corretto del metodo `m` nella classe `C`?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

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

```
public class A {
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 55
- B. 55.5
- C. 55.5d
- 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) {
    try {
        m();
        System.out.print(1);
    }
    catch( MyExc1 t ) {
        System.out.print(2);
    }
    catch( MyExc3 i ) {
    }
    catch( Exception c ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 55

- B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

16. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

```
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
-



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 l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
```

```
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

3. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

4. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

5. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

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

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

7. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

8. Quale delle seguenti non è un sovraccaricamento corretto del metodo `m` nella classe `C`?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

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

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

```
package pk;
```

```
public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

11. Quale output si ottiene invocando il metodo `m`?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

10. Dato un file contenente il seguente codice:

12. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

---

13. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

---

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
}

```

```

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

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

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

```

public class A {
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - 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. 5**

1. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

- B. 000
- C. 110
- D. 010
- E. 100

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

2. Quale output si ottiene invocando il metodo `m`?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 5 5
- B. 5 5.5

- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}
abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}
class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
    }
}

```

```

    }
    catch( MyExc3 i ) {
    }
    catch( Exception c ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

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

9. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

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

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

12. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

13. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

14. Qual'è l'output di questo codice?

```
String c = "I";
void f();

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
```

```

        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

---

16. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }
}

```

```

void remove() {
    length--;
    players[length] = null;
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.
-



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. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`
- B. `g = (Exception) u;`
- C. `x = (Error) g;`
- D. `g = (Exception) x;`
- E. Nessuno dei precedenti

3. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
```

```
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
```



```

}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

5. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

6. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo `g`

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo `g`.

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

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- 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 MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
}

```

```

}
static void q() throws Exception {
    try {
        System.out.print(2);
        throw( new Exception() );
    }
    catch( MyExc2 r ) {
        System.out.print(3);
    }
    catch( MyExc1 c ) {
        System.out.print(4);
    }
    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
}
void type() {

```

```

        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

11. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {

```

```

        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

---

15. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

---

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

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - 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. 7**

1. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poich'è l'oggetto viene restituito dal metodo g.

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

2. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

- A. 134
- B. 14

- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

4. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
```

```

        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

7. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

8. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
```

```
Tree() {
    type();
}
private void type() {
    System.out.println(s);
}
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

10. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

11. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;



E. u = f;

---

12. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

13. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
```

```
protected void m(I i) {
    System.out.print(i.c);
}
protected void m(A a) {
    System.out.print(a.c);
}
protected void m(B c) {
    System.out.print(c.c);
}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

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

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 55
  - B. 55.5
  - C. 55.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

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

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
```

```

        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - 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 A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - 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. 8**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

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

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

4. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}

- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

5. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
}
```

```
private void type() {
    System.out.println(s);
}
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

7. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.

- D. Al termine della linea 8.
- E. Non si può dire, poichè oggetto viene restituito dal metodo g.

---

8. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
- E. Un array non possiede dei membri

---

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

10. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

---

11. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

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

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 55
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

13. Quale output si ottiene invocando il metodo `m`?

```
class B {  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
    void q(Float f2, char [] a2, Float f5) {  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a1 == a2) {
```

```
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
    if(f5 == f4) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

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

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant{  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

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

15. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
String c = "A";
protected void f() {}
abstract void m(A a);
}

class B extends A {
String c = "B";
protected void m(I i) {
System.out.print(i.c);
}
protected void m(A a) {
System.out.print(a.c);
}
protected void m(B b) {
System.out.print(b.c);
}
public static void main(String[] args) {
B c = new B();
c.m((I) c);
c.m((A) c);
c.m(c);
}
}

```

```

}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

16. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. `u = c;`
  - B. `c = (Integer) u;`
  - C. `c = u;`
  - D. `f = c;`
  - E. `u = f;`
-





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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

2. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. x = (Error) u;

- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

3. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

4. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

5. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

6. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

8. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

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

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

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

```
public class A {
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
```

```
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

12. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}

class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

1. Tree g(String s) {
2. Park p = new Park();
3. Tree t = new Tree(s);

```

4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

13. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {

```

```

        System.out.print(5);
    }
}

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {

```

```

        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- 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. 10**

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

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

2. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

3. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

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) {
```



```

try {
    m();
    System.out.print(1);
}
catch( MyExc1 t ) {
    System.out.print(2);
}
catch( MyExc3 i ) {
}
catch( Exception c ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
    throw( new MyExc3() );
}
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
}

```

```

}
public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}

```

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

6. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

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

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

8. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

9. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

```

```

class B extends A {

```

```

    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

10. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

11. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

12. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

13. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
String c = "A";
protected void f() {}
abstract void m(A a);
}

class B extends A {
String c = "B";
protected void m(I i) {
System.out.print(i.c);
}
protected void m(A a) {
System.out.print(a.c);
}
protected void m(B b) {
System.out.print(b.c);
}
public static void main(String[] args) {
B c = new B();
c.m((I) c);
c.m((A) c);
c.m(c);
}
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

14. Qual'è l'output di questo codice?

```

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

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public interface Plant {
String s = "Plant";
String type();
}
abstract class Tree {
String s = "Tree";
Tree() {
type();
}
private void type() {
System.out.println(s);
}
}
class Lemon extends Tree implements Plant {
String s = "Lemon";
public String type() {
return s;
}
}

```

```

public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}

```

- A. Tree
- B. Lemon
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {

```

```

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

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception 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. 11**

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

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}
abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}
class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}
```

```
}
public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
    c.m(c);
}
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```
public class A {
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

6. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;
    void add(Tree t) {
        trees[length++] = t;
    }
    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.

- D. Al termine della linea 8.
- E. Non si può dire, poichè oggetto viene restituito dal metodo g.

---

7. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

---

8. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA

- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

---

9. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

---

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

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

---

11. Quale output si ottiene invocando il metodo m?



```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
    }
}

```

```

        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

14. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

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

- A. `x = (Error) u;`
  - B. `g = (Exception) u;`
  - C. `x = (Error) g;`
  - D. `g = (Exception) x;`
  - E. Nessuno dei precedenti
- 

15. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `sf new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

16. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. `u = c;`
  - B. `c = (Integer) u;`
  - C. `c = u;`
  - D. `f = c;`
  - E. `u = f;`
-



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. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`
- B. `g = (Exception) u;`
- C. `x = (Error) g;`
- D. `g = (Exception) x;`
- E. Nessuno dei precedenti

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree

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

3. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

4. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.

---

5. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

---

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

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

7. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

---

8. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}

E. Nessuna delle precedenti.

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

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

10. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

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) {  
        try {  
            m();  
            System.out.print(1);  
        }  
        catch( MyExc1 t ) {  
            System.out.print(2);  
        }  
    }  
}
```

```
}  
catch( MyExc3 i ) {  
}  
catch( Exception c ) {  
    System.out.print(3);  
}  
finally {  
    System.out.print(4);  
    throw( new MyExc3() );  
}  
}  
static void m() {  
    try {  
        System.out.print(5);  
        throw( new MyExc3() );  
    }  
    catch( MyExc3 f ) {  
    }  
    catch( MyExc2 c ) {  
    }  
    catch( MyExc1 u ) {  
        System.out.print(6);  
    }  
    }  
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
- B. Lemon
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
```

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

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

15. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

16. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
```

```
protected void m(I i) {
    System.out.print(i.c);
}
protected void m(A a) {
    System.out.print(a.c);
}
protected void m(B c) {
    System.out.print(c.c);
}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-





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. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

3. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

4. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;

E. u = f;

---

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

6. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.
- 

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

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 55
  - B. 55.5
  - C. 55.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

8. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}
```

```

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

9. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101

- B. 000
- C. 110
- D. 010
- E. 100

10. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
}

```

```

    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione

- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

13. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

14. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

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

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 55
- B. 55.5
- C. 55.5d

D. Errore a tempo di compilazione

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

```
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}
```

A. 514

B. 514Exception in thread main MyExc3

C. 51

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. 14**

1. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
  - B. `protected void m(int i, char c) throws RuntimeException {}`
  - C. `public final int m() {return 1;}`
  - D. `public void m(int i){}`
  - E. Nessuna delle precedenti.
- 
2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 
3. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 
4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
```



```

Tree() {
    type();
}
void type() {
    System.out.println(s);
}
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

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

```

public interface Plant {
    String s = "Plant";
    String type();
}

```

```

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

---

7. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poich'è l'oggetto viene restituito dal metodo g.
- 

8. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

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

- A. `u = c;`
  - B. `c = (Integer) u;`
  - C. `c = u;`
  - D. `f = c;`
  - E. `u = f;`
- 

9. Qual'è l'output di questo codice?

```
String c = "I";  
void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    protected void f() {}  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B b) {  
        System.out.print(b.c);  
    }  
    public static void main(String[] args) {  
        B c = new B();  
        c.m((I) c);  
        c.m((A) c);  
        c.m(c);  
    }  
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

11. Dato un file contenente il seguente codice:

```
package pk;  
  
public class C {  
    static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

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

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( MyExc2 t ) {
```

```

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

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

13. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 000

- C. 110
- D. 010
- E. 100

14. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

15. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
    }
}
```

```
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - 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. 15**

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

3. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

4. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
```

```

protected void f() {}
abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

5. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);

```

```

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

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

7. Qual'è l'output di questo codice?

```

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

```

```
}  
}
```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Quale output si ottiene invocando il metodo m?

```
class B {  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
    void q(Float f2, char [] a2, Float f5) {  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a1 == a2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f5 == f4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

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

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";
```

```
Tree() {  
    type();  
}  
private void type() {  
    System.out.println(s);  
}  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
  - B. Lemon
  - 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 interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
  - B. Lemon
  - 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 A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

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

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente

D. Non tutti i tipi di eccezioni estendono la classe RuntimeException

E. Un array non possiede dei membri

---

14. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;  
  
    void add(Tree t) {  
        trees[length++] = t;  
    }  
  
    void remove() {  
        length--;  
        players[length] = null;  
    }  
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

A. Al termine della linea 5.

B. Al termine della linea 6.

C. Al termine della linea 7.

D. Al termine della linea 8.

E. Non si può dire, poich'è l'oggetto viene restituito dal metodo g.

---

15. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

```

```

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

```

```

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

```

}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

16. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. `u = c;`
  - B. `c = (Integer) u;`
  - C. `c = u;`
  - D. `f = c;`
  - E. `u = f;`
-



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. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

2. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

3. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

4. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`
- B. `g = (Exception) u;`
- C. `x = (Error) g;`
- D. `g = (Exception) x;`
- E. Nessuno dei precedenti

---

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

---

6. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
    }
}
```

```
        players[length] = null;
    }
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poich'è l'oggetto viene restituito dal metodo g.

---

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

8. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
```

```
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- 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. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

11. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}

public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
}
```

```

        c.m( (A) c );
        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

14. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

15. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {

```

```

        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

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

```

public class A {
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - 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. 17**

1. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poich'è l'oggetto viene restituito dal metodo g.

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

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

3. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        }
    }
}
```

```

    } else {
        System.out.print(0);
    }
}
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

4. Qual'è l'output di questo codice?

```

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

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
    }
}

```

```

        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
}

```

```

    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

---

10. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

---

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- 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. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
- E. Un array non possiede dei membri

---

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

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

14. Qual'è l'output di questo codice?

```
String c = "I";
void f();

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
```

```

        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

15. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

16. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int 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. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

2. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
```

```
public void f() {
    System.out.print(c);
}

abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.



---

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

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 55
- B. 5 5.5
- C. 55.5d
- 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 MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            q();  
            System.out.print(1);  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(2);  
            throw( new Exception() );  
        }  
        catch( MyExc2 r ) {  
            System.out.print(3);  
        }  
        catch( MyExc1 c ) {  
            System.out.print(4);  
        }  
        catch( Exception g ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

6. Dato un file contenente il seguente codice:

```
package pk;  
  
public class C {  
    static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

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

---

7. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;  
  
    void add(Tree t) {  
        trees[length++] = t;  
    }  
  
    void remove() {  
        length--;  
        players[length] = null;  
    }  
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poich'è l'oggetto viene restituito dal metodo g.

---

8. Date le dichiarazioni:

```
Exception g;  
Object u;  
Error x;  
g = new Exception();  
x = new Error();  
u = new Error();
```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

---

9. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

---

10. Qual'è l'output di questo codice?

```
String c = "I";  
void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    protected void f() {}  
    abstract void m(A a);
```

```
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B b) {  
        System.out.print(b.c);  
    }  
    public static void main(String[] args) {  
        B c = new B();  
        c.m((I) c);  
        c.m((A) c);  
        c.m(c);  
    }  
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

---

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( MyExc2 t ) {  
            throw( new MyExc1() );  
        }  
    }  
    static void p() throws Exception {  
        try {  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 h ) {  
            System.out.print(2);  
        }  
        catch( Exception h ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - 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 interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
```

```
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - 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?

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

---

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

```
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - 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. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

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

```
public class A {
    public static void m(long l, char c) {
        double d = 1/c;
        System.out.print(d);
    }
}
```

```
public static void main(String[] args){
    long l = 11;
    char c = 2;
    System.out.print(l/c + " ");
    m(l, c);
}
```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- 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 C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

---

5. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}

class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poich'è l'oggetto viene restituito dal metodo g.

---

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

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

---

7. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

---

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

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 55
- B. 55.5
- C. 55.5d
- 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) {  
        try {  
            m();  
            System.out.print(1);  
        }  
        catch( MyExc1 t ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 i ) {  
        }  
        catch( Exception c ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc3() );  
        }  
    }  
    static void m() {  
        try {  
            System.out.print(5);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 f ) {  
        }  
        catch( MyExc2 c ) {
```

```
        }  
        catch( MyExc1 u ) {  
            System.out.print(6);  
        }  
    }  
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

10. Qual'è l'output di questo codice?

```
String c = "I";  
void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    protected void f() {}  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B b) {  
        System.out.print(b.c);  
    }  
    public static void main(String[] args) {  
        B c = new B();  
        c.m((I) c);  
        c.m((A) c);  
        c.m(c);  
    }  
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

---

11. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi



- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. u = c;
- B. c = (Integer) u;

- C. c = u;
- D. f = c;
- E. u = f;

14. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

15. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. x = (Error) u;

- B. `g = (Exception) u;`
  - C. `x = (Error) g;`
  - D. `g = (Exception) x;`
  - E. Nessuno dei precedenti
- 

16. Qual'è l'output di questo codice?

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

```
    }
    catch( MyExc2 r ) {
        System.out.print(3);
    }
    catch( MyExc1 c ) {
        System.out.print(4);
    }
    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - 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. 20**

1. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

3. Quale delle seguenti non è un sovraccaricamento corretto del metodo `m` nella classe `C`?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
}
```

```

public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}
}

```

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

---

5. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.

6. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

---

7. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

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

---

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

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d

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

```
((A) c).f();  
((B) c).f();  
}  
}
```

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

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B c) {  
        System.out.print(c.c);  
    }  
    public static void main(String[] args) {  
        I c = new B();  
        c.f();  
    }  
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

11. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {

```

```

        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

14. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

---

15. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`
- B. `g = (Exception) u;`
- C. `x = (Error) g;`
- D. `g = (Exception) x;`
- E. Nessuno dei precedenti

---

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
```

```
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - 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. 21**

1. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

2. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
```

```
a3 = new char [5];
q(f3, a3, f4);
}

void q(Float f2, char [] a2, Float f5) {
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a1 == a2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f5 == f4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

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

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 55
- B. 55.5

- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

4. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

---

5. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

---

6. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

---

7. Qual'è l'output di questo codice?

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

```

        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- 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 MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

9. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

11. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}
abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

```

```

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

12. 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExcl t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {

```

```

        }
        catch( MyExcl u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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?

```

public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
    public static void main(String[] args){

```

```
    long l = 11;
    int i = 2;
    System.out.print(l/i + " ");
    m(l, i);
}
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

15. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`
  - B. `g = (Exception) u;`
  - C. `x = (Error) g;`
  - D. `g = (Exception) x;`
  - E. Nessuno dei precedenti
- 

16. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
-



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. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.

2. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

3. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}



- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

---

4. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`
- B. `g = (Exception) u;`
- C. `x = (Error) g;`
- D. `g = (Exception) x;`
- E. Nessuno dei precedenti

---

6. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
- E. Un array non possiede dei membri

---

5. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

---

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

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

8. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

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

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

10. Qual'è l'output di questo codice?

```
String c = "I";  
void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    protected void f() {}  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B b) {  
        System.out.print(b.c);  
    }  
}
```

```
}  
public static void main(String[] args) {  
    B c = new B();  
    c.m((I) c);  
    c.m((A) c);  
    c.m(c);  
}  
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

11. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con sf new C.I().
- D. Un'istanza della classe I può essere costruita con new C().new I() anche fuori da pk.
- E. Nessuna delle precedenti.

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {

```

```

        throw( new MyExc1() );
    }
    catch( MyExc3 h ) {
        System.out.print(2);
    }
    catch( Exception h ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

---

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

```

public interface Plant {
    String s = "Plant";
    String type();
}

```

```

}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - 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. 23**

1. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

2. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

```
}
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010

4. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

5. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`
- B. `g = (Exception) u;`
- C. `x = (Error) g;`
- D. `g = (Exception) x;`

6. Quale delle seguenti non è un sovraccaricamento corretto del metodo `m` nella classe `C`?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

7. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

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

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- 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) {
        try {
            m();
            System.out.print(1);
        }
    }
}
```

```
catch( MyExc1 t ) {
    System.out.print(2);
}
catch( MyExc3 i ) {
}
catch( Exception c ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
    throw( new MyExc3() );
}
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```



```
}  
}
```

- A. Tree
  - B. Lemon
  - 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 MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            q();  
            System.out.print(1);  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(2);  
            throw( new Exception() );  
        }  
        catch( MyExc2 r ) {  
            System.out.print(3);  
        }  
        catch( MyExc1 c ) {  
            System.out.print(4);  
        }  
        catch( Exception g ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

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

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}
```

```
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
  - B. Lemon
  - 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. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
  - E. Un array non possiede dei membri
- 

15. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;  
    void add(Tree t) {  
        trees[length++] = t;  
    }  
}
```

```
}  
  
void remove() {  
    length--;  
    players[length] = null;  
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

---

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

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - 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. 24**

1. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

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

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
```

```

    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

5. Qual'è l'output di questo codice?

```

    String c = "I";
    void f();
}
abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}
class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

7. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.

---

8. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

---

9. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

---

10. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

---

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - 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 A {  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

13. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c) throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int 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) {  
        try {  
            m();  
            System.out.print(1);  
        }  
        catch( MyExc1 t ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 i ) {
```

```
}  
        catch( Exception c ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc3() );  
        }  
    }  
    static void m() {  
        try {  
            System.out.print(5);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 f ) {  
        }  
        catch( MyExc2 c ) {  
        }  
        catch( MyExc1 u ) {  
            System.out.print(6);  
        }  
    }  
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual'è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}
```

```
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B c) {  
        System.out.print(c.c);  
    }  
    public static void main(String[] args) {  
        I c = new B();
```

```
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

16. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
}
```

```
void q(Float f2, char [] a2, Float f5) {
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a1 == a2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f5 == f4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}
```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
-





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?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

2. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
```

```
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

3. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

4. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

5. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

6. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.

7. Qual'è l'output di questo codice?

```

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

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

8. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

---

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

11. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

---

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

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 55
  - B. 55.5
  - C. 55.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

13. Qual'è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B c) {  
        System.out.print(c.c);  
    }  
    public static void main(String[] args) {  
        I c = new B();  
        c.f();  
        ((A) c).f();  
        ((B) c).f();  
    }  
}
```

}

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

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

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
  - B. Lemon
  - 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 A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;
```

```

        System.out.print(1/i + " ");
        m(1, i);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {

```

```

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

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception 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. 26**

1. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

2. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
```

```
abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}

class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
    }
}
```



```

        players[length] = null;
    }

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.

4. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

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

5. Qual'è l'output di questo codice?

```

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

```

```

        q();
        System.out.print(1);
    }
}
static void q() throws Exception {
    try {
        System.out.print(2);
        throw( new Exception() );
    }
    catch( MyExc2 r ) {
        System.out.print(3);
    }
    catch( MyExc1 c ) {
        System.out.print(4);
    }
    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

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

```

public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 5 5
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

8. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

9. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {

```

```

        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

10. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExcl t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {

```

```

        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon

- 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. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

```

}

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
-



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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
```

```

        throw( new Exception() );
    }
    catch( MyExc2 r ) {
        System.out.print(3);
    }
    catch( MyExc1 c ) {
        System.out.print(4);
    }
    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

5. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

---

6. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

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

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

8. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}

class Park {
    private Tree[] trees = new Tree[100];
    int length;
}

```

```

void add(Tree t) {
    trees[length++] = t;
}

void remove() {
    length--;
    players[length] = null;
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

10. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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

```

public class A {
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
    public static void main(String[] args) {
        long l = 11;
    }
}

```



```

        char c = 2;
        System.out.print(1/c + " ");
        m(1, c);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

12. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

15. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

16. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}
```

```
class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-



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. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.

2. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

3. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
```

```

        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
    }
}

```

```

    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

```

    } else {
        System.out.print(0);
    }
    if(f5 == f4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

8. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

9. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        }
    }
}

```

10. Qual'è l'output di questo codice?

```

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

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

12. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
- B. AAA

- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

13. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

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

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

15. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe l può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe l può essere costruita solo all'interno del pacchetto pk.
- C. All'esterno del pacchetto pk, un'istanza della classe l può essere costruita con sf new C.I().
- D. Un'istanza della classe l può essere costruita con new C().new l() anche fuori da pk.
- E. Nessuna delle precedenti.

---

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
}
```

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

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception 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. 29**

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

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

2. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

3. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

4. Qual'è l'output di questo codice?

```
String c = "I";
void f();

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

---

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

---

6. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con sf new C.I().

D. Un'istanza della classe I può essere costruita con new C().new I() anche fuori da pk.

E. Nessuna delle precedenti.

---

7. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

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

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
```

```

        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

10. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
}

```

```

}
void q(Float f2, char [] a2, Float f5) {
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a1 == a2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f5 == f4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
    }
}

```

```

    }
    catch( MyExcl u ) {
        System.out.print(6);
    }
}
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

13. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

14. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

15. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}

class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

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

```

public interface Plant {
    String s = "Plant";
    String type();
}

```

```

}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

2. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`

- B. `g = (Exception) u;`
- C. `x = (Error) g;`
- D. `g = (Exception) x;`
- E. Nessuno dei precedenti

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

4. Qual'è l'output di questo codice?



```

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

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
}

```

```

public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}

```

- A. Tree
- B. Lemon
- 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 A {
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

7. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

8. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

9. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

10. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
```

```
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

12. 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExcl t ) {
            System.out.print(2);
        }
    }
}
```

```

    catch( MyExc3 i ) {
    }
    catch( Exception c ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}

public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
}

```

```

        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

14. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}

class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

15. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c) throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

16. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
}
```

```
void q(Float f2, char [] a2, Float f5) {
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a1 == a2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f5 == f4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
-



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. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
- E. Un array non possiede dei membri

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. `14Exception` 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 MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

3. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

4. Quale output si ottiene invocando il metodo `m`?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

if(a1 == a2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(f5 == f4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

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

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 55
- B. 55.5
- C. 55.5d
- 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 Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {

```

```

        System.out.print(2);
    }
    catch( MyExc3 i ) {
    }
    catch( Exception c ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

8. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poich'è l'oggetto viene restituito dal metodo g.

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

10. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

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

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
}
```



```

public static void main(String[] args) {
    long l = 11;
    int i = 2;
    System.out.print(l/i + " ");
    m(l, i);
}

```

- A. 5 5
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

12. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
}

```

```

}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

14. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

15. Qual'è l'output di questo codice?

```

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

```

```

    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
String c = "A";
protected void f() {}
abstract void m(A a);
}

```

```

class B extends A {
String c = "B";
protected void m(I i) {
    System.out.print(i.c);
}
protected void m(A a) {
    System.out.print(a.c);
}
protected void m(B b) {
    System.out.print(b.c);
}
public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
    c.m(c);
}
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-



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?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

2. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

3. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

4. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
```

```

String c = "B";
protected void m(I i) {
    System.out.print(i.c);
}
protected void m(A a) {
    System.out.print(a.c);
}
protected void m(B c) {
    System.out.print(c.c);
}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
    }
}

```

```

        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- 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 MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

7. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

8. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

11. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

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

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
```

```
public String type() {
    return s;
}
public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}
}
```

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

14. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - 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. 33**

1. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

2. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
```

```
throws Exception {
    try {
        System.out.print(1);
        p();
    }
    catch( MyExc2 t ) {
        throw( new MyExc1() );
    }
}
static void p() throws Exception {
    try {
        throw( new MyExc1() );
    }
    catch( MyExc3 h ) {
        System.out.print(2);
    }
    catch( Exception h ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
```

```

    }
    catch( Exception c ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

6. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
    }
}

```

```

        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

---

8. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

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

```
public class A {
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

10. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}
abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}
class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

---

11. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException

E. Un array non possiede dei membri

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

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

13. Date le dichiarazioni:

```
Exception g;  
Object u;  
Error x;  
g = new Exception();  
x = new Error();  
u = new Error();
```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

14. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];
```

```
int length;  
  
void add(Tree t) {  
    trees[length++] = t;  
}  
  
void remove() {  
    length--;  
    players[length] = null;  
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poich'è l'oggetto viene restituito dal metodo g.

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

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

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

16. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
```

```
protected void m(I i) {
    System.out.print(i.c);
}
protected void m(A a) {
    System.out.print(a.c);
}
protected void m(B c) {
    System.out.print(c.c);
}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-



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 dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.

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

3. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo `g`

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo `g`.



4. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
  - E. Un array non possiede dei membri
- 

6. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

7. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c) throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

9. Qual'è l'output di questo codice?

```

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

```

```

        System.out.print(4);
    }
    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

11. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;

- D. g = (Exception) x;
- E. Nessuno dei precedenti

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

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

13. Dato un file contenente il seguente codice:

```
package pk;  
  
public class C {  
    static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

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

14. Quale output si ottiene invocando il metodo m?

```
class B {  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
    void q(Float f2, char [] a2, Float f5) {  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a1 == a2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f5 == f4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

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

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
    }  
}
```

```
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
```

```
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - 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. 35**

1. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

2. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}
```

```
abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}
```

```
class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

4. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException

E. Un array non possiede dei membri

---

5. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`
  - B. `g = (Exception) u;`
  - C. `x = (Error) g;`
  - D. `g = (Exception) x;`
  - E. Nessuno dei precedenti
- 

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

7. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. `u = c;`
  - B. `c = (Integer) u;`
  - C. `c = u;`
  - D. `f = c;`
  - E. `u = f;`
- 

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

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

9. 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExcl t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
    }
}
```

```

    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

11. Date le classi

```

class Tree {
    String type;
    Tree(String s) {

```

```

        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.

12. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}

```



```

    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 MyExc2 { }
public class B1 {
    public static void main(String [] argv)

```

```

    throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {

```

```

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

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception 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. 36**

1. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`
- B. `g = (Exception) u;`
- C. `x = (Error) g;`
- D. `g = (Exception) x;`
- E. Nessuno dei precedenti

2. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
```

```
c.f();
((A) c).f();
((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA

- C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

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

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 55
  - B. 55.5
  - C. 55.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

5. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;  
  
    void add(Tree t) {  
        trees[length++] = t;  
    }  
  
    void remove() {  
        length--;  
        players[length] = null;  
    }  
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);
```

```
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.
- 

6. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

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

```
public class A {
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 5 5
- B. 55.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

8. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

---

9. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
```

```
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a1 == a2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f5 == f4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

---

10. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

---

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
```

```

    }
    catch( MyExc3 h ) {
        System.out.print(2);
    }
    catch( Exception h ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

15. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}

- C. `public final int m() {return 1;}`
  - D. `public void m(int i){}`
  - 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
```

```
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - 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. 37**

1. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
```

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
}
```

```

private void type() {
    System.out.println(s);
}
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

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

4. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

5. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

if(a1 == a2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(f5 == f4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

6. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

7. 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExcl t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

```

static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

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

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
}

```

```

public static void main(String[] args){
    long l = 11;
    int i = 2;
    System.out.print(l/i + " ");
    m(l, i);
}
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

10. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}

class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.

11. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

12. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

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

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- 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 MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
```

```
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
}
```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`
  - B. `g = (Exception) u;`
  - C. `x = (Error) g;`
  - D. `g = (Exception) x;`
  - E. Nessuno dei precedenti
- 

16. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
-



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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

2. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con sf new C.I().
- D. Un'istanza della classe I può essere costruita con new C().new I() anche fuori da pk.
- E. Nessuna delle precedenti.

3. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException { }
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
    }
}
```



```

    }
    catch( Exception h ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

5. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

6. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

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

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
}

```

```

public String type() {
    return s;
}
public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}
}

```

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

9. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1
- B. 2451

- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

13. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
    }
}
```

```
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

16. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
```

```
String c = "B";
protected void m(I i) {
    System.out.print(i.c);
}
protected void m(A a) {
    System.out.print(a.c);
}
protected void m(B b) {
    System.out.print(b.c);
}
public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
    c.m(c);
}
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-



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. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c) throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}

public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
    c.m(c);
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- 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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
```

```
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - 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 MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
-

7. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
- E. Un array non possiede dei membri

8. Quale output si ottiene invocando il metodo `m`?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

9. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo `g`

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo `g`.

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

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```



- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

12. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

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

13. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

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

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon

- 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 A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 5 5
- B. 5 5.5

- C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

16. Date le dichiarazioni:

```
Exception g;  
Object u;  
Error x;  
g = new Exception();  
x = new Error();  
u = new Error();
```

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

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - 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. 40**

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

2. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}
abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}
```

```
}
class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Qual'è l'output di questo codice?

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

```

    catch( MyExc2 r ) {
        System.out.print(3);
    }
    catch( MyExc1 c ) {
        System.out.print(4);
    }
    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

5. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];

```

```

        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

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

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

8. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

9. Date le dichiarazioni:

```
Exception g;  
Object u;  
Error x;  
g = new Exception();  
x = new Error();  
u = new Error();
```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

10. Dato un file contenente il seguente codice:

```
package pk;  
  
public class C {  
    static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

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

11. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

12. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. `static long m(long l){return l;}`
  - B. `protected void m(int i, char c) throws RuntimeException {}`
  - C. `public final int m() {return 1;}`
  - D. `public void m(int i){}`
  - E. Nessuna delle precedenti.
- 

### 13. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.
- 

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - 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 Bl {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
    }
}
```

```

    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}

```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

```

```

}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-





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 il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- 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?

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
    public static void main(String[] args){
        long l = 11;
        int i = 2;
```

```
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1

E. Nessuna delle precedenti

---

4. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

5. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
```

```
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

---

6. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC

- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

---

7. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

---

8. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

---

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

10. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

11. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

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

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

14. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

15. Qual'è l'output di questo codice?

```

    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

A. IAC

B. AAA

C. CCC

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

16. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

A. Un'istanza della classe I può essere costruita solo all'interno della classe C.

B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `sf new C.I()`.

D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.

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. 42**

1. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. `x = (Error) u;`
- B. `g = (Exception) u;`
- C. `x = (Error) g;`
- D. `g = (Exception) x;`
- E. Nessuno dei precedenti

2. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
```

```
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451



- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

5. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

6. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

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

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

7. Quale output si ottiene invocando il metodo m?

```
class B {  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
    void q(Float f2, char [] a2, Float f5) {  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a1 == a2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f5 == f4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

8. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;
```

```

void add(Tree t) {
    trees[length++] = t;
}

void remove() {
    length--;
    players[length] = null;
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

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

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 55
- B. 55.5
- C. 55.5d
- 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 MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
    }
}

```

```

    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

12. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

13. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
    }
}

```

```

        c.m(c);
    }
}

```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- 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 C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con sf new C.I().
- D. Un'istanza della classe I può essere costruita con new C().new I() anche fuori da pk.
- 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. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

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

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

3. Qual'è l'output di questo codice?

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

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

5. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

---

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

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    void type() {  
        System.out.println(s);  
    }  
}
```

```
class Lemon extends Tree implements Plant{  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

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

---

7. Qual'è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B c) {  
        System.out.print(c.c);  
    }  
    public static void main(String[] args) {  
        I c = new B();  
        c.f();  
        ((A) c).f();  
        ((B) c).f();  
    }  
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-

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

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

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

9. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

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

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

10. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c) throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

11. Qual'è l'output di questo codice?

```
String c = "I";
void f();

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

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) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

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

```

public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

14. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

15. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;
}

```

```

void add(Tree t) {
    trees[length++] = t;
}

void remove() {
    length--;
    players[length] = null;
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.

D. Al termine della linea 8.

E. Non si può dire, poichè l'oggetto viene restituito dal metodo g.

16. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

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

Prova n. 1

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

---

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Prova n. 2

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

---

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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



Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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



Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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



Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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



Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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



Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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

Studente e matricola:

Ora di inizio:

Ora di consegna:

---

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: