Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

2. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

3. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.
B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.
C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.
D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.
E. Il codice viene compilato ed eseguito correttamente.

---

4. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

6. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 l'output di questo codice?

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

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

10. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f)
   {return f;}`
C. `protected void f(int i, float f)
    throws RuntimeException {}`
D. `public final int f()
    {return 9;}`
E. Nessuna delle precedenti.

---

11. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

12. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
```

```
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

13. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

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 Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

---

16. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111

B. 110

C. 101

D. 100

E. 000

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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

Prova n. 2

1. Qual'è l'output di questo codice?

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

   A. 1563Exception in thread "main" MyExc1
   B. Errore a tempo di compilazione
   C. 1563333333... (ciclo infinito)
   D. 15634Exception in thread "main" MyExc1
   E. Nessuna delle precedenti

---

2. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

   A. `n = (Object [] []) b;`
   B. `m = (Exception []) n;`
   C. `n = (Object [] []) m;`
   D. `m = (Exception []) b;`
   E. Nessuno dei precedenti

---

3. Quale delle seguenti affermazioni è vera?

   A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
   B. È possibile usare il metodo System.gc() per forzare la garbage collection.
   C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
   D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
   E. Nessuna delle precedenti 'e vera.

---

4. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
   E. Nessuna delle precedenti.

5. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

6. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

7. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
```

```
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

9. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

10. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      I c = new C();
      c.f();
      ((A) c).f();
      ((C) 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 MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
```

```
        }
    }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

12. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

13. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

14. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.
B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.
C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.
D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.
E. Il codice viene compilato ed eseguito correttamente.

---

15. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
   }
}
```

A. `public void f(int i){}`

B. `static long f(long f) {return f;}`

C. `protected void f(int i, float f) throws RuntimeException {}`

D. `public final int f() {return 9;}`

E. Nessuna delle precedenti.

---

16. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

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

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe `E` non può implementare due interfacce.
B. La classe `E` non ha fornito l'implementazione di tutti i metodi dell'interfaccia `I2`.
C. Vi è un conflitto delle dichiarazioni dei due metodi `m()` nelle interfacce `I1` e `I2`.
D. Vi è un conflitto delle dichiarazioni degli attributi `c1` nelle interfacce `I1` e `I2`.
E. Il codice viene compilato ed eseguito correttamente.

---

2. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

4. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

5. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f) {return f;}`
C. `protected  void f(int i, float f) throws RuntimeException {}`
D. `public final int f() {return 9;}`
E. Nessuna delle precedenti.

---

6. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

7. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

8. Qual'è l'output di questo codice?

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

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)

D. 15634Exception in thread ”main” MyExc1

E. Nessuna delle precedenti

---

9. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.

B. È possibile usare il metodo System.gc() per forzare la garbage collection.

C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.

D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.

E. Nessuna delle precedenti ‘e vera.

---

10. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111

B. 110

C. 101

D. 100

E. 000

---

11. Qual’è l’output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione

B. 45213

C. 4523

D. 451

E. Nessuna delle precedenti

---

12. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*

B. Nell’overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri

C. Una classe non interna può essere dichiarata private

D. Un attributo può essere contemporaneamente static e final

E. Una classe non può essere dichiarata protected

---

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

14. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

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

---

15. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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 Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

3. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.
B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

4. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`

B. `static long f(long f) {return f;}`

C. `protected  void f(int i, float f) throws RuntimeException {}`

D. `public final int f() {return 9;}`

E. Nessuna delle precedenti.

---

5. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

6. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

A. IAC

B. AAA

C. CCC

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

---

7. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*

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

C. Una classe non interna può essere dichiarata private

D. Un attributo può essere contemporaneamente static e final

E. Una classe non può essere dichiarata protected

---

8. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.

B. È possibile usare il metodo System.gc() per forzare la garbage collection.

C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.

D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.

E. Nessuna delle precedenti 'e vera.

---

10. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

---

11. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111

B. 110

C. 101

D. 100

E. 000

12. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

13. Qual'è l'output di questo codice?

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

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

---

14. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

15. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

E. Nessuna delle precedenti.

---

16. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

2. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

3. Qual'è l'output di questo codice?

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

A. 1563Exception in thread ″main″ MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread ″main″ MyExc1

E. Nessuna delle precedenti

---

4. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 {
public 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'interno 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().
E. Nessuna delle precedenti.

---

6. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
```

```
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

8. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f) {return f;}`
C. `protected  void f(int i, float f) throws RuntimeException {}`
D. `public final int f() {return 9;}`
E. Nessuna delle precedenti.

---

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

10. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

11. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.

C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.

D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.

E. Nessuna delle precedenti 'e vera.

---

12. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

---

13. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

14. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

---

15. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*

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

C. Una classe non interna può essere dichiarata private

D. Un attributo può essere contemporaneamente static e final

E. Una classe non può essere dichiarata protected

---

16. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

A. IAC

B. AAA

C. CCC

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      I c = new C();
      c.f();
      ((A) c).f();
      ((C) 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 Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

3. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

---

4. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

E. Nessuna delle precedenti.

---

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3

B. Errore a tempo di compilazione

C. 1423

D. 1Exception in thread "main" MyExc3

E. Nessuna delle precedenti

---

7. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

8. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111

B. 110

C. 101

D. 100

E. 000

---

9. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*

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

C. Una classe non interna può essere dichiarata private

D. Un attributo può essere contemporaneamente static e final

E. Una classe non può essere dichiarata protected

---

10. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

---

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
```

```
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
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 A1 {
   public static void main(String [] argv)
throws Exception {
      try {
         System.out.print(1);
         q();
         System.out.print(2);
      }
      catch( Exception e ) {
         System.out.print(3);
         throw( new MyExc1() );
      }
      finally {
         System.out.print(4);
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc3() );
      }
      finally {
         System.out.print(6);
         throw( new Exception() );
      }
   }
}
```

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

---

13. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe `E` non può implementare due interfacce.
B. La classe `E` non ha fornito l'implementazione di tutti i metodi dell'interfaccia `I2`.
C. Vi è un conflitto delle dichiarazioni dei due metodi `m()` nelle interfacce `I1` e `I2`.
D. Vi è un conflitto delle dichiarazioni degli attributi `c1` nelle interfacce `I1` e `I2`.
E. Il codice viene compilato ed eseguito correttamente.

---

14. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.

B. È possibile usare il metodo System.gc() per forzare la garbage collection.

C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.

D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.

E. Nessuna delle precedenti 'e vera.

---

15. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`

B. `static long f(long f)`
`{return f;}`

C. `protected  void f(int i, float f)`
`throws RuntimeException {}`

D. `public final int f()`
`{return 9;}`

E. Nessuna delle precedenti.

---

16. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`

B. `b = d;`

C. `b = f;`

D. `f = b;`

E. `b = (Integer []) f;`

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

   A. `public void f(int i){}`
   B. `static long f(long f) {return f;}`
   C. `protected  void f(int i, float f) throws RuntimeException {}`
   D. `public final int f() {return 9;}`
   E. Nessuna delle precedenti.

---

2. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

   A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

---

4. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*

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

C. Una classe non interna può essere dichiarata private

D. Un attributo può essere contemporaneamente static e final

E. Una classe non può essere dichiarata protected

---

5. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      I c = new C();
      c.f();
      ((A) c).f();
      ((C) 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 MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3

B. Errore a tempo di compilazione

C. 1423

D. 1Exception in thread "main" MyExc3

E. Nessuna delle precedenti

---

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
```

```
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

8. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
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 A1 {
   public static void main(String [] argv)
throws Exception {
      try {
         System.out.print(1);
         q();
         System.out.print(2);
      }
      catch( Exception e ) {
         System.out.print(3);
         throw( new MyExc1() );
      }
      finally {
         System.out.print(4);
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc3() );
      }
      finally {
         System.out.print(6);
         throw( new Exception() );
      }
   }
}
```

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

---

10. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

11. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

---

12. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.

B. È possibile usare il metodo System.gc() per forzare la garbage collection.

C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.

D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.

E. Nessuna delle precedenti 'e vera.

---

13. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

14. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

---

15. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

16. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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

Prova n. 8

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

2. Qual'è l'output di questo codice?

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

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

---

3. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`

E. `b = (Integer []) f;`

---

4. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

5. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

7. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.
B. Definire private la variabile anni.
C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

8. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111

B. 110

C. 101

D. 100

E. 000

---

9. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

---

10. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.

B. È possibile usare il metodo System.gc() per forzare la garbage collection.

C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.

D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.

E. Nessuna delle precedenti 'e vera.

---

11. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`

B. `static long f(long f)`
`{return f;}`

C.
```
protected  void f(int i, float f)
    throws RuntimeException {}
```

D.
```
public final int f()
    {return 9;}
```

E. Nessuna delle precedenti.

---

12. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

A. IAC

B. AAA

C. CCC

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

---

13. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

E. Nessuna delle precedenti.

---

14. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

15. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*

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

C. Una classe non interna può essere dichiarata private

D. Un attributo può essere contemporaneamente static e final

E. Una classe non può essere dichiarata protected

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

   A. `public void f(int i){}`
   B. `static long f(long f) {return f;}`
   C. `protected  void f(int i, float f) throws RuntimeException {}`
   D. `public final int f() {return 9;}`
   E. Nessuna delle precedenti.

---

2. Qual'è l'output di questo codice?

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

   A. 1563Exception in thread "main" MyExc1
   B. Errore a tempo di compilazione
   C. 1563333333... (ciclo infinito)
   D. 15634Exception in thread "main" MyExc1
   E. Nessuna delle precedenti

---

3. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

   A. `f = (Object [] []) d;`
   B. `b = d;`
   C. `b = f;`
   D. `f = b;`
   E. `b = (Integer []) f;`

---

4. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}
```

```
abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

5. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

---

6. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111

B. 110

C. 101

D. 100

E. 000

---

7. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

8. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}
```

```
class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

10. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213

C. 4523

D. 451

E. Nessuna delle precedenti

---

12. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

A. IAC

B. AAA

C. CCC

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

---

13. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.

B. È possibile usare il metodo System.gc() per forzare la garbage collection.

C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.

D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.

E. Nessuna delle precedenti 'e vera.

---

14. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
```

```
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 {
public 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'interno 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().
E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

2. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      I c = new C();
      c.f();
      ((A) c).f();
      ((C) 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 MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
```

```
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

4. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

5. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

6. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f)`
   `{return f;}`
C. `protected  void f(int i, float f)`
   `throws RuntimeException {}`
D. `public final int f()`
   `{return 9;}`
E. Nessuna delle precedenti.

---

8. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

9. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`

B. `b = d;`

C. `b = f;`

D. `f = b;`

E. `b = (Integer []) f;`

---

10. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111

B. 110

C. 101

D. 100

E. 000

---

11. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
```

```
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

12. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

13. Qual'è l'output di questo codice?

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

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

---

14. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

15. Qual'è l'output di questo codice?

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

```
            throw( new MyExc3() );
         }
      }
      static void n() {
         try {
            throw( new MyExc2() );
         }
         catch( MyExc1 j ) {
            System.out.print(4);
         }
      }
   }
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

16. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

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

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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 l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

   A. 142Exception in thread "main" MyExc3
   B. Errore a tempo di compilazione
   C. 1423
   D. 1Exception in thread "main" MyExc3
   E. Nessuna delle precedenti

---

2. Dire quale delle seguenti affermazioni è vera:

   A. Ai metodi static si applica il *dynamic method dispatch*
   B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   C. Una classe non interna può essere dichiarata private
   D. Un attributo può essere contemporaneamente static e final
   E. Una classe non può essere dichiarata protected

---

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

   A. Errore a tempo di compilazione
   B. 45213
   C. 4523
   D. 451
   E. Nessuna delle precedenti

---

4. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

5. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

---

6. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111

B. 110

C. 101

D. 100

E. 000

---

7. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
```

```
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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

---

8. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

9. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

10. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

11. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

12. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

13. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f) {return f;}`
C. `protected  void f(int i, float f) throws RuntimeException {}`
D. `public final int f() {return 9;}`
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 A1 {
   public static void main(String [] argv)
throws Exception {
      try {
         System.out.print(1);
         q();
         System.out.print(2);
      }
      catch( Exception e ) {
         System.out.print(3);
         throw( new MyExc1() );
      }
      finally {
         System.out.print(4);
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc3() );
      }
      finally {
         System.out.print(6);
         throw( new Exception() );
      }
   }
}
```

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`

D. `f = b;`

E. `b = (Integer []) f;`

---

16. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

A. IAC

B. AAA

C. CCC

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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

Prova n. 12

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

2. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

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

---

3. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

4. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

5. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe `E` non può implementare due interfacce.
B. La classe `E` non ha fornito l'implementazione di tutti i metodi dell'interfaccia `I2`.
C. Vi è un conflitto delle dichiarazioni dei due metodi `m()` nelle interfacce `I1` e `I2`.
D. Vi è un conflitto delle dichiarazioni degli attributi `c1` nelle interfacce `I1` e `I2`.
E. Il codice viene compilato ed eseguito correttamente.

---

6. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

E. Nessuna delle precedenti.

---

8. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`

B. `b = d;`

C. `b = f;`

D. `f = b;`

E. `b = (Integer []) f;`

---

9. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

10. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`

B. `static long f(long f)`
`{return f;}`

C. `protected  void f(int i, float f)`
`throws RuntimeException {}`

D. `public final int f()`
`{return 9;}`

E. Nessuna delle precedenti.

---

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

12. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

14. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

15. Qual'è l'output di questo codice?

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

A. 1563Exception in thread ″main″ MyExc1

B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread ”main” MyExc1
E. Nessuna delle precedenti

---

16. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe `E` non può implementare due interfacce.
B. La classe `E` non ha fornito l'implementazione di tutti i metodi dell'interfaccia `I2`.
C. Vi è un conflitto delle dichiarazioni dei due metodi `m()` nelle interfacce `I1` e `I2`.
D. Vi è un conflitto delle dichiarazioni degli attributi `c1` nelle interfacce `I1` e `I2`.
E. Il codice viene compilato ed eseguito correttamente.

---

2. Qual'è l'output di questo codice?

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

A. 1563Exception in thread ″main″ MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception 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 MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
```

```
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

4. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

5. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

6. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
```

```
        }
    }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

7. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

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

---

8. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

9. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

10. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`

B. `static long f(long f)`
`{return f;}`

C. `protected  void f(int i, float f)`
`throws RuntimeException {}`

D. `public final int f()`
`{return 9;}`

E. Nessuna delle precedenti.

---

11. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*

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

C. Una classe non interna può essere dichiarata private

D. Un attributo può essere contemporaneamente static e final

E. Una classe non può essere dichiarata protected

---

12. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

---

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3

B. Errore a tempo di compilazione

C. 1423

D. 1Exception in thread "main" MyExc3

E. Nessuna delle precedenti

---

14. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

15. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

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

---

16. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

3. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f)`
   `{return f;}`
C. `protected  void f(int i, float f)`
   `throws RuntimeException {}`
D. `public final int f()`
   `{return 9;}`
E. Nessuna delle precedenti.

---

4. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
```

```
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

5. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

6. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe `E` non può implementare due interfacce.
B. La classe `E` non ha fornito l'implementazione di tutti i metodi dell'interfaccia `I2`.
C. Vi è un conflitto delle dichiarazioni dei due metodi `m()` nelle interfacce `I1` e `I2`.
D. Vi è un conflitto delle dichiarazioni degli attributi `c1` nelle interfacce `I1` e `I2`.
E. Il codice viene compilato ed eseguito correttamente.

---

7. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi `static` si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata `private`

D. Un attributo può essere contemporaneamente static e final

E. Una classe non può essere dichiarata protected

---

8. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.

B. È possibile usare il metodo System.gc() per forzare la garbage collection.

C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.

D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.

E. Nessuna delle precedenti 'e vera.

---

10. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

A. IAC

B. AAA

C. CCC

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

---

11. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3

B. Errore a tempo di compilazione

C. 1423

D. 1Exception in thread "main" MyExc3

E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`

B. `b = d;`

C. `b = f;`

D. `f = b;`

E. `b = (Integer []) f;`

---

14. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

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

A. 1563Exception in thread "main" MyExc1

B. Errore a tempo di compilazione

C. 1563333333... (ciclo infinito)

D. 15634Exception in thread ”main” MyExc1

E. Nessuna delle precedenti

---

16. Qual’è l’output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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 MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

2. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

3. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

A. IAC
B. AAA
C. CCC

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

---

4. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

---

5. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

6. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`

B. `b = d;`

C. `b = f;`

D. `f = b;`

E. `b = (Integer []) f;`

---

7. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*

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

C. Una classe non interna può essere dichiarata private

D. Un attributo può essere contemporaneamente static e final

E. Una classe non può essere dichiarata protected

---

8. Qual'è l'output di questo codice?

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

A. 1563Exception in thread "main" MyExc1

B. Errore a tempo di compilazione

C. 1563333333... (ciclo infinito)

D. 15634Exception in thread "main" MyExc1

E. Nessuna delle precedenti

---

9. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f) {return f;}`
C. `protected  void f(int i, float f) throws RuntimeException {}`
D. `public final int f() {return 9;}`
E. Nessuna delle precedenti.

---

11. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
```

```
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

14. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

15. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.
B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.
C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

16. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111

B. 110

C. 101

D. 100

E. 000

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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

Prova n. 16

1. Qual'è l'output di questo codice?

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

   A. 1563Exception in thread ″main″ MyExc1
   B. Errore a tempo di compilazione
   C. 1563333333... (ciclo infinito)
   D. 15634Exception in thread ″main″ MyExc1
   E. Nessuna delle precedenti

---

2. Dire quale delle seguenti affermazioni è vera:

   A. Ai metodi static si applica il *dynamic method dispatch*
   B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   C. Una classe non interna può essere dichiarata private
   D. Un attributo può essere contemporaneamente static e final
   E. Una classe non può essere dichiarata protected

---

3. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

   A. 5 April Easter
   B. Easter Easter
   C. Errore a tempo di compilazione
   D. Errore a tempo di esecuzione
   E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

5. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f)`
   `{return f;}`
C. `protected  void f(int i, float f)`
   `throws RuntimeException {}`
D. `public final int f()`
   `{return 9;}`
E. Nessuna delle precedenti.

---

6. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

7. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

8. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
```

```
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

11. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

E. Nessuna delle precedenti.

---

12. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.

B. È possibile usare il metodo System.gc() per forzare la garbage collection.

C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.

D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.

E. Nessuna delle precedenti 'e vera.

---

13. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

15. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi `c1` nelle interfacce `I1` e `I2`.

E. Il codice viene compilato ed eseguito correttamente.

---

16. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

A. IAC

B. AAA

C. CCC

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

   A. `public void f(int i){}`
   B. `static long f(long f) {return f;}`
   C. `protected  void f(int i, float f) throws RuntimeException {}`
   D. `public final int f() {return 9;}`
   E. Nessuna delle precedenti.

---

2. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

   A. `n = (Object [] []) b;`
   B. `m = (Exception []) n;`
   C. `n = (Object [] []) m;`
   D. `m = (Exception []) b;`
   E. Nessuno dei precedenti

---

3. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

   A. 111
   B. 110
   C. 101
   D. 100
   E. 000

---

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
```

```
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

5. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

6. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

7. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

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

---

8. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

9. Qual'è l'output di questo codice?

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

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

---

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

11. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

12. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}
```

```
abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

13. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

---

14. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

E. Nessuna delle precedenti.

---

15. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
```

```
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

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

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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 delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

   A. `public void f(int i){}`
   B. `static long f(long f) {return f;}`
   C. `protected  void f(int i, float f) throws RuntimeException {}`
   D. `public final int f() {return 9;}`
   E. Nessuna delle precedenti.

---

2. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

   A. La classe `E` non può implementare due interfacce.
   B. La classe `E` non ha fornito l'implementazione di tutti i metodi dell'interfaccia `I2`.
   C. Vi è un conflitto delle dichiarazioni dei due metodi `m()` nelle interfacce `I1` e `I2`.
   D. Vi è un conflitto delle dichiarazioni degli attributi `c1` nelle interfacce `I1` e `I2`.
   E. Il codice viene compilato ed eseguito correttamente.

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

5. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

6. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

7. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

8. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

10. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
```

```
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

11. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

12. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

13. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

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

---

14. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

15. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

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

A. 1563Exception in thread "main" MyExc1

B. Errore a tempo di compilazione

C. 1563333333... (ciclo infinito)

D. 15634Exception in thread "main" MyExc1

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

2. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

4. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
```

```
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.
B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.
C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.
D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.
E. Il codice viene compilato ed eseguito correttamente.

---

6. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

7. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

8. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
```

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

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 G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

10. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f) {return f;}`
C. `protected  void f(int i, float f) throws RuntimeException {}`
D. `public final int f() {return 9;}`
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 A1 {
   public static void main(String [] argv)
throws Exception {
      try {
         System.out.print(1);
         q();
         System.out.print(2);
      }
      catch( Exception e ) {
         System.out.print(3);
         throw( new MyExc1() );
      }
      finally {
         System.out.print(4);
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc3() );
      }
      finally {
         System.out.print(6);
         throw( new Exception() );
      }
   }
}
```

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)

D. 15634Exception in thread ″main″ MyExc1

E. Nessuna delle precedenti

---

12. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

---

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione

B. 45213

C. 4523

D. 451

E. Nessuna delle precedenti

---

15. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*

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

C. Una classe non interna può essere dichiarata private

D. Un attributo può essere contemporaneamente static e final

E. Una classe non può essere dichiarata protected

---

16. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

2. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

3. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

4. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f) {return f;}`
C. `protected  void f(int i, float f) throws RuntimeException {}`
D. `public final int f() {return 9;}`
E. Nessuna delle precedenti.

---

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

7. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

8. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.
B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.
C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.
D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.
E. Il codice viene compilato ed eseguito correttamente.

---

9. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

10. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

11. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

12. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

13. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

14. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

15. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      I c = new C();
      c.f();
      ((A) c).f();
      ((C) 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 A1 {
   public static void main(String [] argv)
throws Exception {
      try {
         System.out.print(1);
         q();
         System.out.print(2);
      }
      catch( Exception e ) {
         System.out.print(3);
         throw( new MyExc1() );
      }
      finally {
         System.out.print(4);
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc3() );
      }
      finally {
         System.out.print(6);
         throw( new Exception() );
      }
   }
}
```

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      I c = new C();
      c.f();
      ((A) c).f();
      ((C) 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 Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String [] argv)
throws Exception {
      try {
```

```
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception e ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(6);
            throw( new Exception() );
        }
    }
}
```

A. 1563Exception in thread ”main” MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread ”main” MyExc1
E. Nessuna delle precedenti

---

4. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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’interno 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().
E. Nessuna delle precedenti.

---

5. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti ‘e vera.

---

6. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.
B. La classe E non ha fornito l’implementazione di tutti i metodi dell’interfaccia I2.
C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.
D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.
E. Il codice viene compilato ed eseguito correttamente.

---

7. Qual’è l’output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
```

```
  abstract void m(A a);
}

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

---

8. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

9. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

10. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

11. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.
B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

12. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`

B. `b = d;`

C. `b = f;`

D. `f = b;`

E. `b = (Integer []) f;`

---

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione

B. 45213

C. 4523

D. 451

E. Nessuna delle precedenti

---

14. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

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 f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`

B. `static long f(long f)`
`{return f;}`

C. `protected  void f(int i, float f)`
`throws RuntimeException {}`

D. `public final int f()`
`{return 9;}`

E. Nessuna delle precedenti.

---

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3

B. Errore a tempo di compilazione

C. 1423

D. 1Exception in thread "main" MyExc3

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Quale delle seguenti affermazioni è vera?

   A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
   B. È possibile usare il metodo System.gc() per forzare la garbage collection.
   C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
   D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
   E. Nessuna delle precedenti 'e vera.

---

2. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

   A. 111
   B. 110
   C. 101
   D. 100
   E. 000

---

3. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

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

4. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

5. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

7. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

8. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

9. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

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

A. 1563Exception in thread "main" MyExc1

B. Errore a tempo di compilazione

C. 1563333333... (ciclo infinito)

D. 15634Exception in thread "main" MyExc1

E. Nessuna delle precedenti

---

11. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

---

12. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

14. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

15. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
```

```
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

16. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`

B. `static long f(long f)`
`{return f;}`

C. `protected  void f(int i, float f)`
`throws RuntimeException {}`

D. `public final int f()`
`{return 9;}`

E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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

Prova n. 23

1. Qual'è l'output di questo codice?

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

   A. 1563Exception in thread ″main″ MyExc1
   B. Errore a tempo di compilazione
   C. 1563333333... (ciclo infinito)
   D. 15634Exception in thread ″main″ MyExc1
   E. Nessuna delle precedenti

---

2. Dire quale delle seguenti affermazioni è vera:

   A. Ai metodi static si applica il *dynamic method dispatch*
   B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   C. Una classe non interna può essere dichiarata private
   D. Un attributo può essere contemporaneamente static e final
   E. Una classe non può essere dichiarata protected

---

3. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

   A. `f = (Object [] []) d;`
   B. `b = d;`
   C. `b = f;`
   D. `f = b;`
   E. `b = (Integer []) f;`

---

4. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

   A. `n = (Object [] []) b;`
   B. `m = (Exception []) n;`
   C. `n = (Object [] []) m;`
   D. `m = (Exception []) b;`
   E. Nessuno dei precedenti

---

5. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f) {return f;}`
C. `protected  void f(int i, float f) throws RuntimeException {}`
D. `public final int f() {return 9;}`
E. Nessuna delle precedenti.

---

7. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

8. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

9. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

10. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione

B. 45213

C. 4523

D. 451

E. Nessuna delle precedenti

---

12. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

E. Nessuna delle precedenti.

---

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3

B. Errore a tempo di compilazione

C. 1423

D. 1Exception in thread "main" MyExc3

E. Nessuna delle precedenti

---

14. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

---

15. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

```
      ((A) c).f();
      ((C) 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 G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Quale delle seguenti affermazioni è vera?

   A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
   B. È possibile usare il metodo System.gc() per forzare la garbage collection.
   C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
   D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
   E. Nessuna delle precedenti 'e vera.

---

2. Qual'è l'output di questo codice?

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

   A. 1563Exception in thread "main" MyExc1
   B. Errore a tempo di compilazione
   C. 1563333333... (ciclo infinito)
   D. 15634Exception in thread "main" MyExc1
   E. Nessuna delle precedenti

---

3. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

   A. 5 April Easter
   B. Easter Easter
   C. Errore a tempo di compilazione
   D. Errore a tempo di esecuzione
   E. Nessuna delle precedenti

4. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.
B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.
C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.
D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.
E. Il codice viene compilato ed eseguito correttamente.

---

5. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

8. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

9. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

10. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

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

11. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

12. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

13. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f)`
   `{return f;}`
C. `protected  void f(int i, float f)`
    `throws RuntimeException {}`
D. `public final int f()`
    `{return 9;}`
E. Nessuna delle precedenti.

---

14. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

15. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
```

```
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111

B. 110

C. 101

D. 100

E. 000

---

16. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

2. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

3. Qual'è l'output di questo codice?

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

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

---

4. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
```

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

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      I c = new C();
      c.f();
      ((A) c).f();
      ((C) 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 Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f) {return f;}`
C. `protected  void f(int i, float f) throws RuntimeException {}`
D. `public final int f() {return 9;}`
E. Nessuna delle precedenti.

---

7. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

8. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

9. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
```

```
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

12. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

14. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.
B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.
C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.
D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.
E. Il codice viene compilato ed eseguito correttamente.

---

15. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

16. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}
```

```
class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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 delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

   A. `public void f(int i){}`
   B. `static long f(long f) {return f;}`
   C. `protected  void f(int i, float f) throws RuntimeException {}`
   D. `public final int f() {return 9;}`
   E. Nessuna delle precedenti.

---

2. Quale delle seguenti affermazioni è vera?

   A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
   B. È possibile usare il metodo System.gc() per forzare la garbage collection.
   C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
   D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
   E. Nessuna delle precedenti 'e vera.

---

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

   A. Errore a tempo di compilazione
   B. 45213
   C. 4523
   D. 451
   E. Nessuna delle precedenti

---

4. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}
```

```
class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

5. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

6. Qual'è l'output di questo codice?

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

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

---

7. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
```

```
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

---

8. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`

B. `b = d;`

C. `b = f;`

D. `f = b;`

E. `b = (Integer []) f;`

---

9. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

E. Nessuna delle precedenti.

---

10. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111

B. 110

C. 101

D. 100

E. 000

---

11. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

12. Dire quale delle seguenti affermazioni è vera:

    A. Ai metodi static si applica il *dynamic method dispatch*
    B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
    C. Una classe non interna può essere dichiarata private
    D. Un attributo può essere contemporaneamente static e final
    E. Una classe non può essere dichiarata protected

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

    A. 142Exception in thread "main" MyExc3
    B. Errore a tempo di compilazione
    C. 1423
    D. 1Exception in thread "main" MyExc3
    E. Nessuna delle precedenti

14. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
```

```
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

16. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

A. IAC

B. AAA

C. CCC

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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

   A. 1563Exception in thread ”main” MyExc1
   B. Errore a tempo di compilazione
   C. 1563333333... (ciclo infinito)
   D. 15634Exception in thread ”main” MyExc1
   E. Nessuna delle precedenti

---

2. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

   A. `n = (Object [] []) b;`
   B. `m = (Exception []) n;`
   C. `n = (Object [] []) m;`
   D. `m = (Exception []) b;`
   E. Nessuno dei precedenti

---

3. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

5. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

6. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
```

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

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

---

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

9. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

10. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

11. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
```

```
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

14. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f)`
   `{return f;}`
C. `protected  void f(int i, float f)`
   `throws RuntimeException {}`
D. `public final int f()`
   `{return 9;}`
E. Nessuna delle precedenti.

---

15. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
```

```
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

16. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

2. Qual'è l'output di questo codice?

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

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

---

3. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
```

```
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

4. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

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

---

5. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

6. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

7. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`

B. `b = d;`

C. `b = f;`

D. `f = b;`

E. `b = (Integer []) f;`

---

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione

B. 45213

C. 4523

D. 451

E. Nessuna delle precedenti

---

9. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
```

```
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

12. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

13. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

14. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

15. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`

B. `static long f(long f)`
`{return f;}`

C. `protected  void f(int i, float f)`
`throws RuntimeException {}`

D. `public final int f()`
`{return 9;}`

E. Nessuna delle precedenti.

---

16. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}
class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe `E` non può implementare due interfacce.

B. La classe `E` non ha fornito l'implementazione di tutti i metodi dell'interfaccia `I2`.

C. Vi è un conflitto delle dichiarazioni dei due metodi `m()` nelle interfacce `I1` e `I2`.

D. Vi è un conflitto delle dichiarazioni degli attributi `c1` nelle interfacce `I1` e `I2`.

E. Il codice viene compilato ed eseguito correttamente.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

2. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      I c = new C();
      c.f();
      ((A) c).f();
      ((C) 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 MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213

C. 4523

D. 451

E. Nessuna delle precedenti

---

4. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

---

5. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`

B. `static long f(long f)`
`{return f;}`

C. `protected  void f(int i, float f)`
`throws RuntimeException {}`

D. `public final int f()`
`{return 9;}`

E. Nessuna delle precedenti.

---

6. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

---

7. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
```

```
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

8. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.
B. Definire private la variabile anni.
C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
D. Definire protected la variabile anni e aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.
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 A1 {
   public static void main(String [] argv)
throws Exception {
      try {
         System.out.print(1);
         q();
         System.out.print(2);
      }
      catch( Exception e ) {
         System.out.print(3);
         throw( new MyExc1() );
      }
      finally {
         System.out.print(4);
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc3() );
      }
      finally {
         System.out.print(6);
         throw( new Exception() );
      }
   }
}
```

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

---

10. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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. Ai metodi static si applica il *dynamic method dispatch*

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

C. Una classe non interna può essere dichiarata private

D. Un attributo può essere contemporaneamente static e final

E. Una classe non può essere dichiarata protected

---

12. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.

B. È possibile usare il metodo System.gc() per forzare la garbage collection.

C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.

D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.

E. Nessuna delle precedenti 'e vera.

---

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3

B. Errore a tempo di compilazione

C. 1423

D. 1Exception in thread "main" MyExc3

E. Nessuna delle precedenti

---

14. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

E. Nessuna delle precedenti.

---

15. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

16. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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

Prova n. 30

1. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

2. Date le seguenti classi:

```
class C {
    String s = "C";
}

class D extends C {
}
```

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

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

---

3. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
```

```
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

5. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

7. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`

B. `static long f(long f)`
`{return f;}`

C. `protected  void f(int i, float f)`
`throws RuntimeException {}`

D. `public final int f()`
`{return 9;}`

E. Nessuna delle precedenti.

---

8. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

E. Nessuna delle precedenti.

---

9. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

10. Qual'è l'output di questo codice?

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

A. 1563Exception in thread "main" MyExc1

B. Errore a tempo di compilazione

C. 1563333333... (ciclo infinito)

D. 15634Exception in thread "main" MyExc1

E. Nessuna delle precedenti

---

11. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

14. Dire quale delle seguenti affermazioni è vera:

    A. Ai metodi static si applica il *dynamic method dispatch*
    B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
    C. Una classe non interna può essere dichiarata private
    D. Un attributo può essere contemporaneamente static e final
    E. Una classe non può essere dichiarata protected

---

15. Quale delle seguenti affermazioni è vera?

    A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
    B. È possibile usare il metodo System.gc() per forzare la garbage collection.
    C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
    D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
    E. Nessuna delle precedenti 'e vera.

---

16. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

    A. `n = (Object [] []) b;`
    B. `m = (Exception []) n;`
    C. `n = (Object [] []) m;`
    D. `m = (Exception []) b;`
    E. Nessuno dei precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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

Prova n. 31

1. Qual'è l'output di questo codice?

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

A. 1563Exception in thread ”main” MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread ”main” MyExc1
E. Nessuna delle precedenti

---

2. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

3. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`

E. `b = (Integer []) f;`

---

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

5. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

6. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

7. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.

D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.

E. Nessuna delle precedenti 'e vera.

---

8. Dire quale delle seguenti affermazioni è vera:

   A. Ai metodi static si applica il *dynamic method dispatch*

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

   C. Una classe non interna può essere dichiarata private

   D. Un attributo può essere contemporaneamente static e final

   E. Una classe non può essere dichiarata protected

---

9. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

   A. `public void f(int i){}`

   B. `static long f(long f)`
   `{return f;}`

   C. `protected  void f(int i, float f)`
   `throws RuntimeException {}`

   D. `public final int f()`
   `{return 9;}`

   E. Nessuna delle precedenti.

---

10. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

   E. Nessuna delle precedenti.

---

11. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

   A. La classe E non può implementare due interfacce.

   B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

   C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

   D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

   E. Il codice viene compilato ed eseguito correttamente.

---

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
```

```
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

13. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

15. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110

C. 101
D. 100
E. 000

---

16. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

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

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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

Prova n. 32

1. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

2. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

3. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

5. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f)`
   `{return f;}`
C. `protected  void f(int i, float f)`
   `throws RuntimeException {}`
D. `public final int f()`
   `{return 9;}`
E. Nessuna delle precedenti.

---

7. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

---

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
```

```
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione

B. 45213

C. 4523

D. 451

E. Nessuna delle precedenti

---

9. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

---

10. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.

B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.

C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.

D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.

E. Il codice viene compilato ed eseguito correttamente.

---

11. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*

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

C. Una classe non interna può essere dichiarata private

D. Un attributo può essere contemporaneamente static e final

E. Una classe non può essere dichiarata protected

---

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
```

```
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

13. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

14. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

15. Qual'è l'output di questo codice?

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

A. 1563Exception in thread ”main” MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread ”main” MyExc1
E. Nessuna delle precedenti

---

16. Qual’è l’output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

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

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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

Prova n. 33

1. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

2. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

3. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe E non può implementare due interfacce.
B. La classe E non ha fornito l'implementazione di tutti i metodi dell'interfaccia I2.
C. Vi è un conflitto delle dichiarazioni dei due metodi m() nelle interfacce I1 e I2.
D. Vi è un conflitto delle dichiarazioni degli attributi c1 nelle interfacce I1 e I2.
E. Il codice viene compilato ed eseguito correttamente.

---

4. Qual'è l'output di questo codice?

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

```
            throw( new MyExc2() );
         }
         catch( MyExc1 j ) {
            System.out.print(4);
         }
      }
}
```

A. 142Exception in thread ”main” MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread ”main” MyExc3
E. Nessuna delle precedenti

---

5. Qual’è l’output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
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 {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
```

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

---

8. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

10. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

11. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      I c = new C();
      c.f();
      ((A) c).f();
      ((C) 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 {
public 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'interno 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().
E. Nessuna delle precedenti.

---

13. Qual'è l'output di questo codice?

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

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

---

14. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f)`
   `{return f;}`

C. `protected void f(int i, float f) throws RuntimeException {}`

D. `public final int f() {return 9;}`

E. Nessuna delle precedenti.

---

15. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`

B. `b = d;`

C. `b = f;`

D. `f = b;`

E. `b = (Integer []) f;`

---

16. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

A. Le classi sono perfettamente incapsulate.

B. Definire private la variabile anni.

C. Aggiungere due metodi, uno che ne modifichi ed uno che ne restituisca il valore.

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

E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Qual'è l'output di questo codice?

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

   A. 1563Exception in thread "main" MyExc1
   B. Errore a tempo di compilazione
   C. 1563333333... (ciclo infinito)
   D. 15634Exception 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 Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

   A. 142Exception in thread "main" MyExc3
   B. Errore a tempo di compilazione
   C. 1423
   D. 1Exception in thread "main" MyExc3
   E. Nessuna delle precedenti

---

3. Dire quale delle seguenti affermazioni è vera:

   A. Ai metodi static si applica il *dynamic method dispatch*
   B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   C. Una classe non interna può essere dichiarata private
   D. Un attributo può essere contemporaneamente static e final
   E. Una classe non può essere dichiarata protected

---

4. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

5. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

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

---

6. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}

class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe `E` non può implementare due interfacce.
B. La classe `E` non ha fornito l'implementazione di tutti i metodi dell'interfaccia `I2`.
C. Vi è un conflitto delle dichiarazioni dei due metodi `m()` nelle interfacce `I1` e `I2`.
D. Vi è un conflitto delle dichiarazioni degli attributi `c1` nelle interfacce `I1` e `I2`.
E. Il codice viene compilato ed eseguito correttamente.

---

7. Qual'è l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
```

```
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

8. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

10. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

11. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

---

12. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

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

```
class Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

---

14. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f)`
   `{return f;}`

C. `protected void f(int i, float f) throws RuntimeException {}`

D. `public final int f() {return 9;}`

E. Nessuna delle precedenti.

---

15. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().

E. Nessuna delle precedenti.

---

16. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`

B. `m = (Exception []) n;`

C. `n = (Object [] []) m;`

D. `m = (Exception []) b;`

E. Nessuno dei precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

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. Date le dichiarazioni:

```
Integer [] b;
Object [] d;
Object [] [] f;
```

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

A. `f = (Object [] []) d;`
B. `b = d;`
C. `b = f;`
D. `f = b;`
E. `b = (Integer []) f;`

---

2. Quale delle seguenti affermazioni è vera?

A. Gli oggetti sono immediatamente deallocati non appena diventano eleggibili per la garbage collection.
B. È possibile usare il metodo System.gc() per forzare la garbage collection.
C. È possibile usare il metodo Runtime.gc() per forzare la garbage collection.
D. La garbage collection non libera mai la memoria occupata da oggetti che sono ancora accessibili dallo stack di attivazione.
E. Nessuna delle precedenti 'e vera.

---

3. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  protected void f() {
      System.out.print(c);
  }
  abstract void m(A a);
}

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

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

---

4. Date le seguenti classi:

```
class C {
    String s = "C";
}
class D extends C {
}
```

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

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

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
   public static void main(String[] argv)
throws Exception {
      try {
         m();
         System.out.print(1);
      }
      catch( Exception c ) {
         System.out.print(2);
      }
      catch( MyExc2 y ) {
      }
      finally {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         System.out.print(4);
         throw( new Exception() );
      }
      catch( MyExc3 b ) {
      }
      catch( MyExc2 w ) {
      }
      finally {
         System.out.print(5);
      }
   }
}
```

A. Errore a tempo di compilazione
B. 45213
C. 4523
D. 451
E. Nessuna delle precedenti

---

6. Dato un file contenente il seguente codice:

```
interface I1{
    int c1 = 3;
    int c2 = 6;
    void m();

}
interface I2{
    int c1 = 1;
    public void f();
    public int g();
    public void m();
}
class D implements I1{
    public void m(){}
}

abstract class E extends D implements I2{
    public void f(){
        System.out.print(c1 + " " +  c2);
    }
}
```

Dire quale delle seguenti affermazioni è vera:

A. La classe `E` non può implementare due interfacce.
B. La classe `E` non ha fornito l'implementazione di tutti i metodi dell'interfaccia `I2`.
C. Vi è un conflitto delle dichiarazioni dei due metodi `m()` nelle interfacce `I1` e `I2`.
D. Vi è un conflitto delle dichiarazioni degli attributi `c1` nelle interfacce `I1` e `I2`.
E. Il codice viene compilato ed eseguito correttamente.

---

7. Date le dichiarazioni:

```
Object [] b;
Exception [] m;
Object [] [] n;
m = new Exception [4];
b = new Object [5] [4];
n = new Object [2] [1];
```

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

A. `n = (Object [] []) b;`
B. `m = (Exception []) n;`
C. `n = (Object [] []) m;`
D. `m = (Exception []) b;`
E. Nessuno dei precedenti

---

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
   public static void main(String[] argv){
      try {
         System.out.print(1);
         n();
         System.out.print(2);
      }
      catch( MyExc3 d ) {
         System.out.print(3);
      }
      catch( Exception b ) {
```

```
      }
      finally {
         throw( new MyExc3() );
      }
   }
   static void n() {
      try {
         throw( new MyExc2() );
      }
      catch( MyExc1 j ) {
         System.out.print(4);
      }
   }
}
```

A. 142Exception in thread "main" MyExc3
B. Errore a tempo di compilazione
C. 1423
D. 1Exception in thread "main" MyExc3
E. Nessuna delle precedenti

---

9. Dire quale delle seguenti affermazioni è vera:

A. Ai metodi static si applica il *dynamic method dispatch*
B. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
C. Una classe non interna può essere dichiarata private
D. Un attributo può essere contemporaneamente static e final
E. Una classe non può essere dichiarata protected

---

10. Quale output si ottiene invocando il metodo m?

```
class G {
   private Float f4;
   private String s1;
   private String [] a2;
   void m() {
      Float f3 = new Float(50.0);
      f4 = f3;
      s1 = "abc";
      p(f3, f4);
   }
   void p(Float f1, Float f2) {
      String [] a1 = new String [8];
      a2 = new String [8];
      if(f2 == f1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s1 == "abc") {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(a1 == a2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 111
B. 110
C. 101
D. 100
E. 000

---

11. Quale delle seguenti non è un sovraccaricamento corretto del metodo f nella classe A?

```
class A {
  protected int f(int a) throws Exception{
      return a;
  }
}
```

A. `public void f(int i){}`
B. `static long f(long f) {return f;}`
C. `protected  void f(int i, float f) throws RuntimeException {}`
D. `public final int f() {return 9;}`
E. Nessuna delle precedenti.

---

12. Dato un file contenente il seguente codice:

```
package pk;

public class C {
public 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'interno 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().
E. Nessuna delle precedenti.

13. Qual'è l'output di questo codice?

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

A. 1563Exception in thread "main" MyExc1
B. Errore a tempo di compilazione
C. 1563333333... (ciclo infinito)
D. 15634Exception in thread "main" MyExc1
E. Nessuna delle precedenti

---

14. Qual'è l'output di questo codice?

```
interface I {
  char c = 'I';
  void f();
}

abstract class A implements I {
  char c = 'A';
  public void f() {}
  abstract void m(A a);
}

class C extends A {
  char c = 'C';
  protected void m(I i) {
      System.out.print(i.c);
  }
  protected void m(A a) {
      System.out.print(a.c);
  }
  protected void m(C c) {
      System.out.print(c.c);
  }
  public static void main(String[] args){
      C c = new C();
      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 l'output di questo codice?

```
class Date {
  String day;
  String month;
  public Date(){}
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.print(day+" "+month+" ");
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n){
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.print(name+" ");
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter

B. Easter Easter
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 Date {
  String day;
  String month;
  public Date(String d, String m) {
    day = d;
    month = m;
    print();
  }
  public void print(){
    System.out.println(day +" "+month);
  }
}

class Holiday extends Date {
  String name;
  public Holiday(String n) {
    name = n;
    print();
  }
  public Holiday(String d,String m,String n) {
    super(d, m);
    name = n;
    print();
  }
  public void print(){
    System.out.println(name);
  }
  public static void main(String[] args) {
    Date x = new Holiday("5","April","Easter");
    Holiday y = new Holiday("Easter");
  }
}
```

A. 5 April Easter
B. Easter Easter
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
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 — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 2

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 3

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 4

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 5

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 6

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 7

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 8

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 9

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 10

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 11

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 12

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 13

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 14

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 15

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 16

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 17

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 18

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 19

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 20

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 21

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 22

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 23

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 24

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 25

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 26

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 27

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 28

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 29

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 30

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 31

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 32

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 33

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 34

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

Nome e Cognome:

Matricola:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 35

Università di Napoli Federico II – Corso di Laurea in Informatica

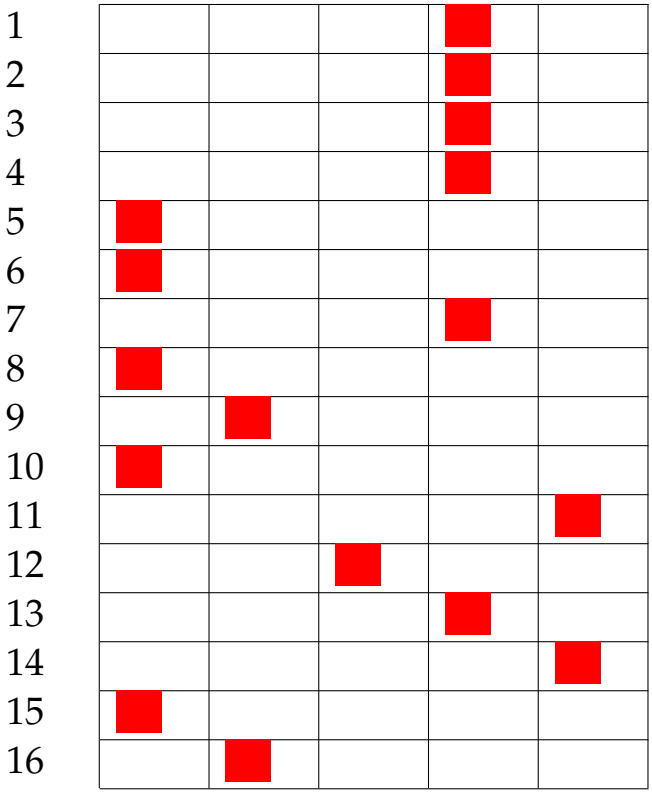
LP1 — Prova d'esame

prof. Piero A. Bonatti — *17-07-2014*

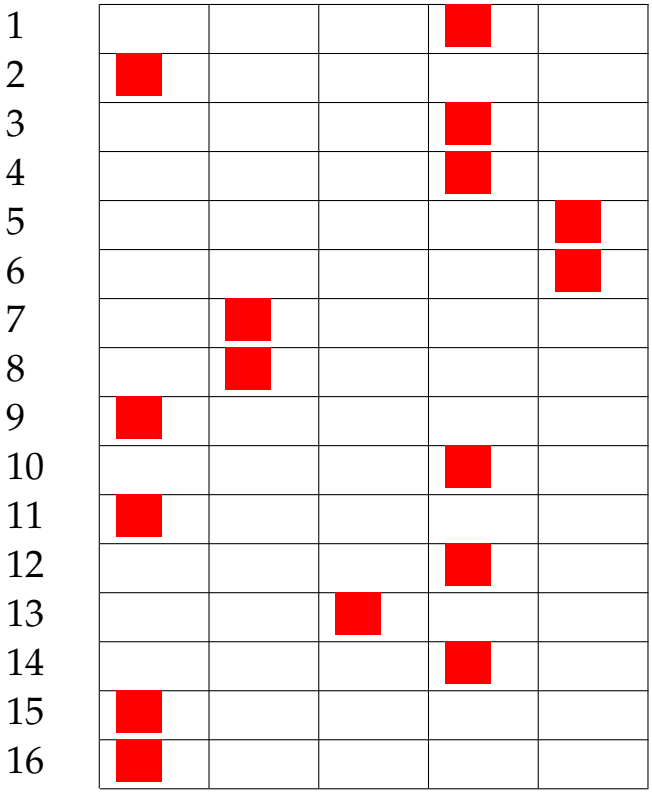
Nome e Cognome:

Matricola:

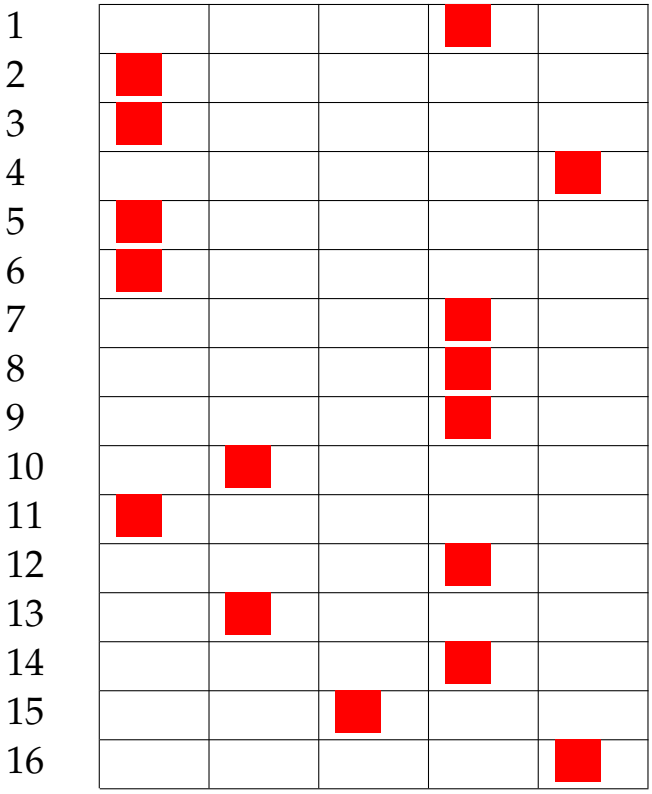
1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E



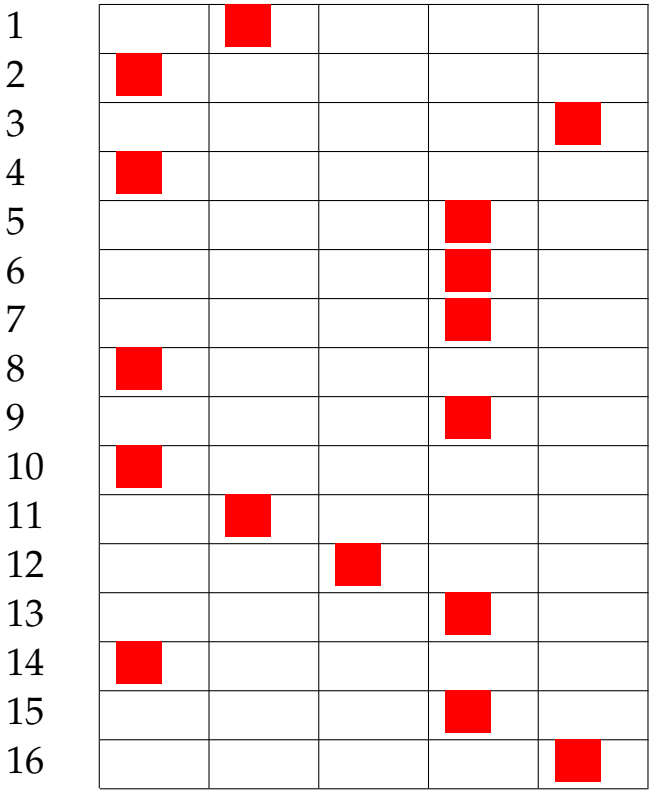
Risultato prova n. 1:



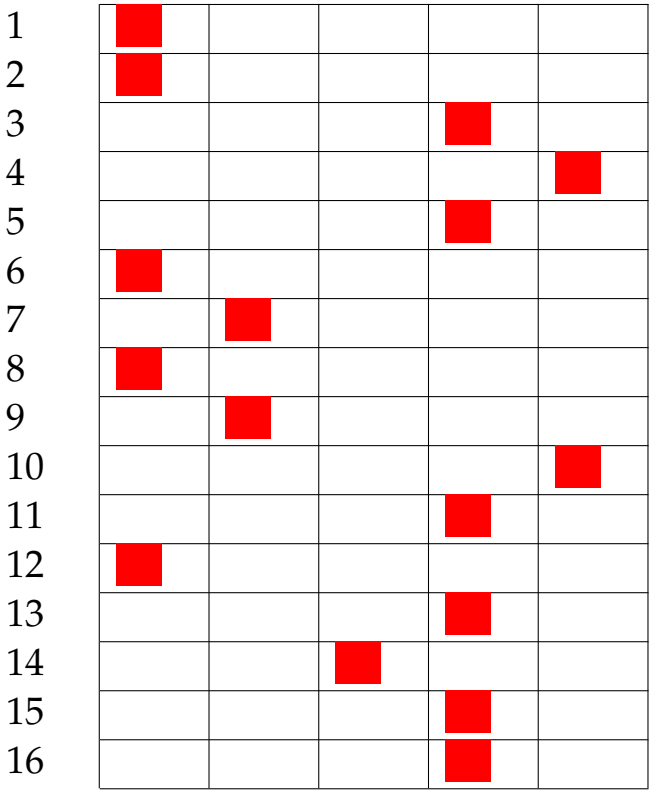
Risultato prova n. 2:



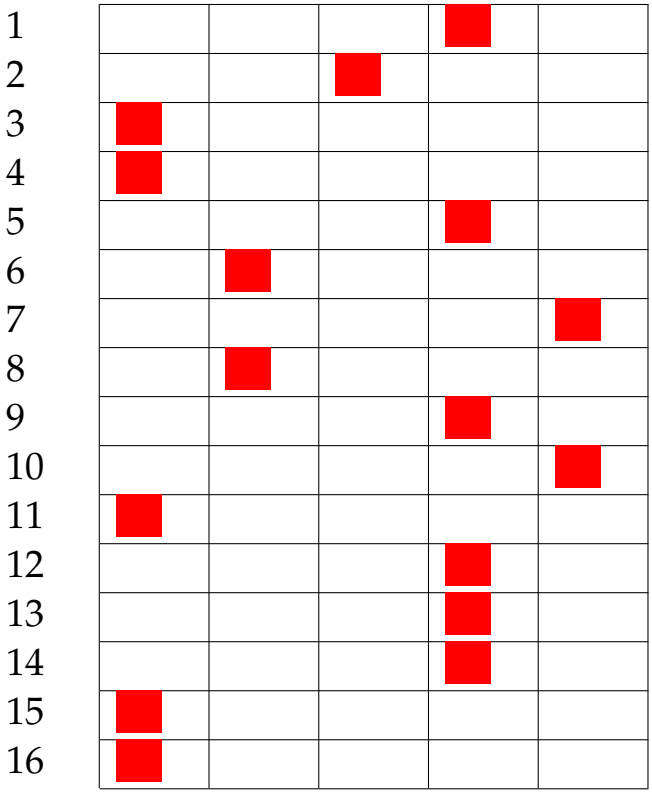
Risultato prova n. 3:



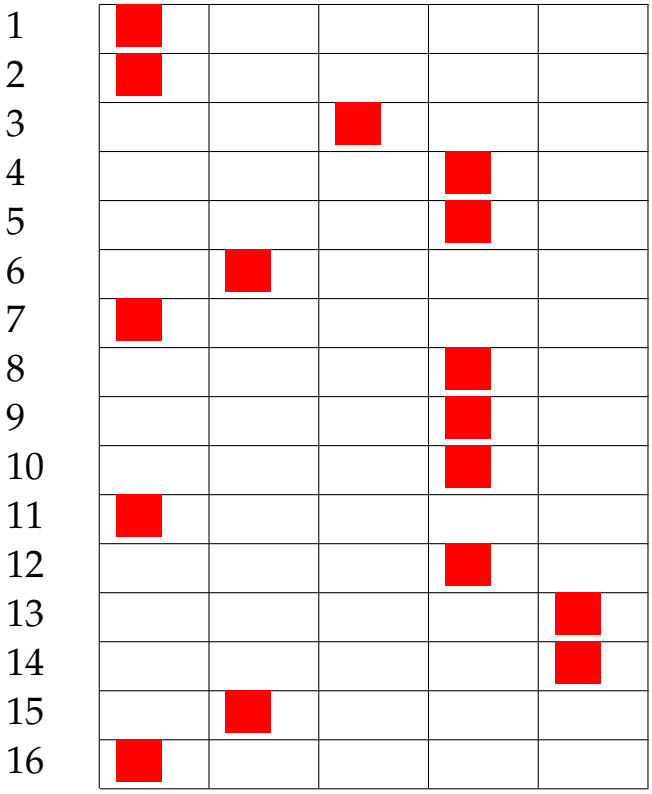
Risultato prova n. 4:



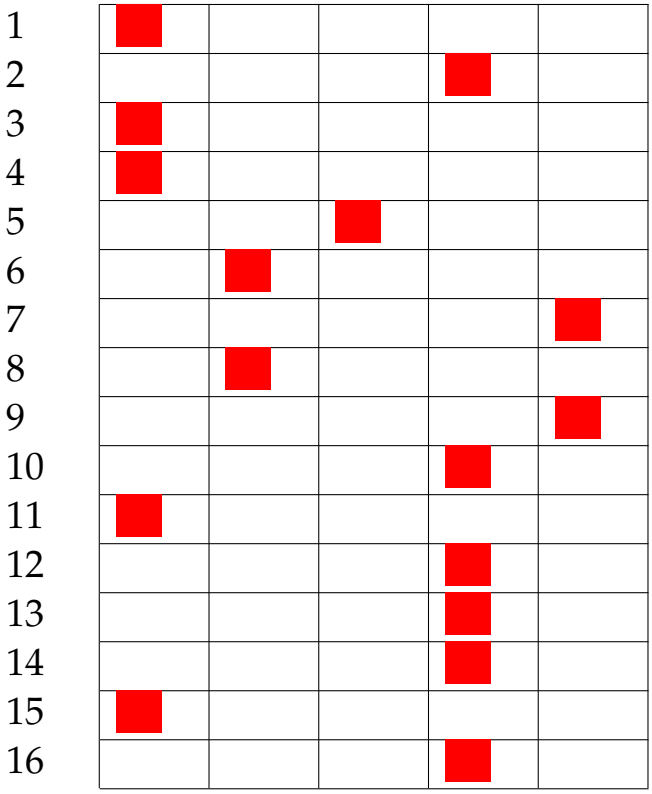
Risultato prova n. 5:



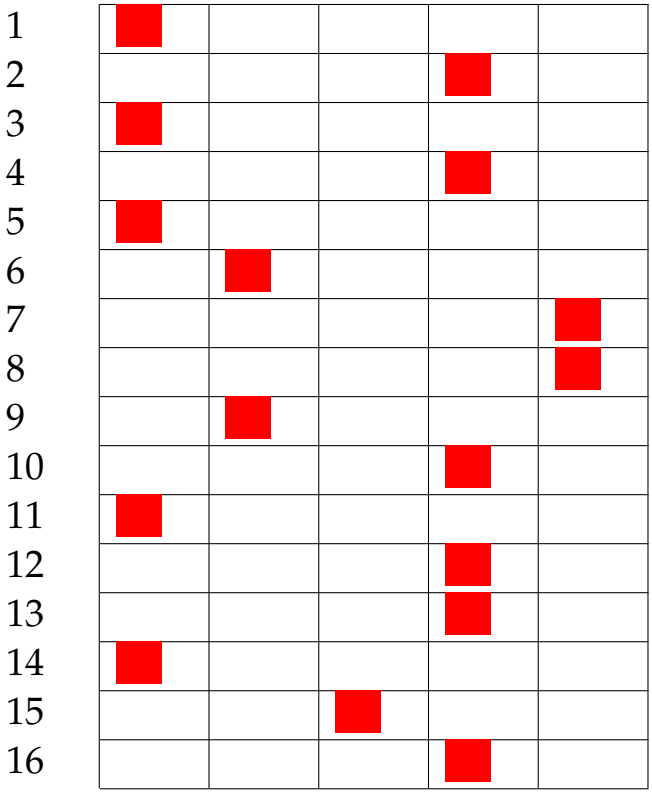
Risultato prova n. 6:



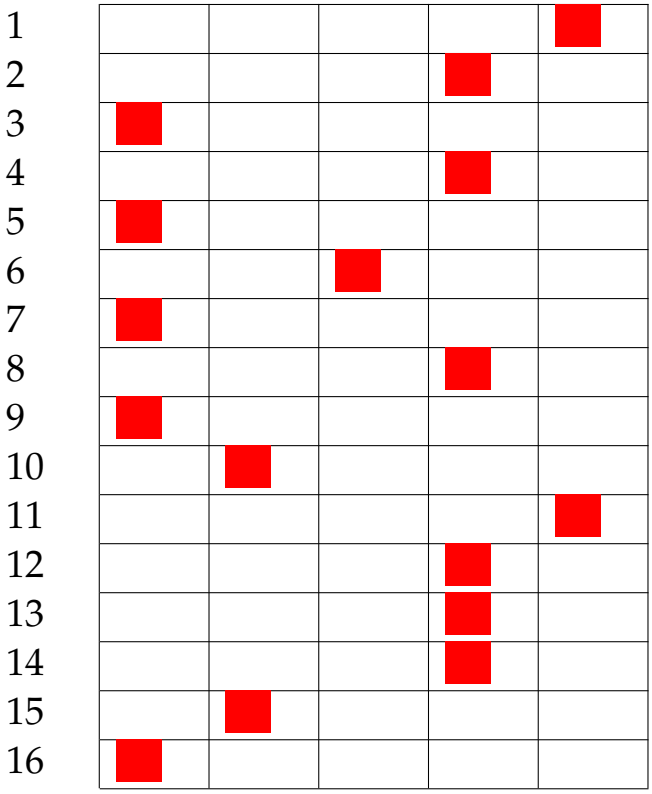
Risultato prova n. 7:



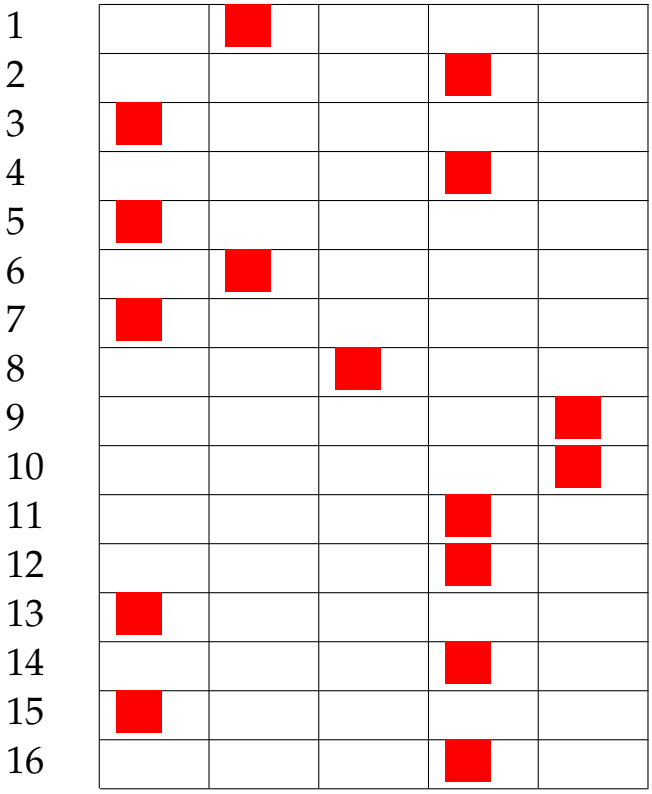
Risultato prova n. 8:



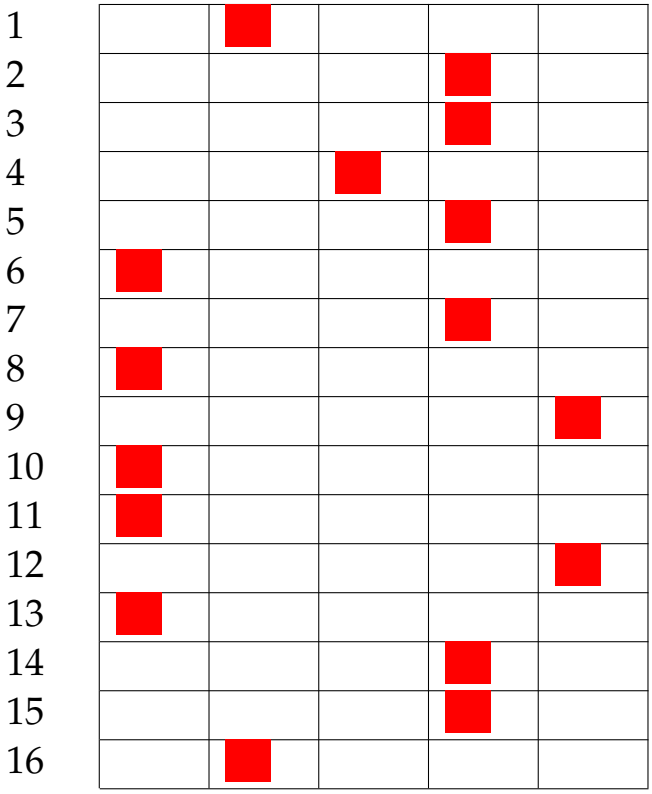
Risultato prova n. 9:



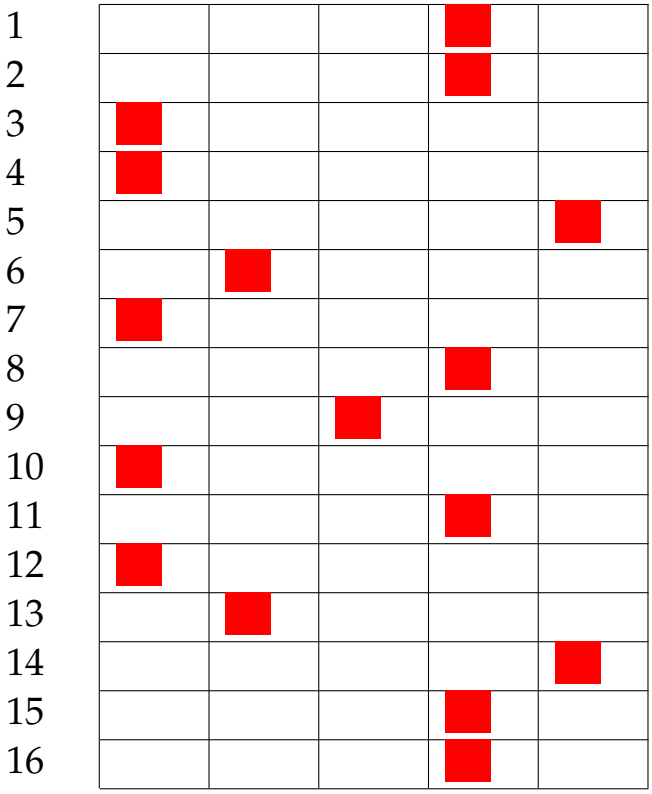
Risultato prova n. 10:



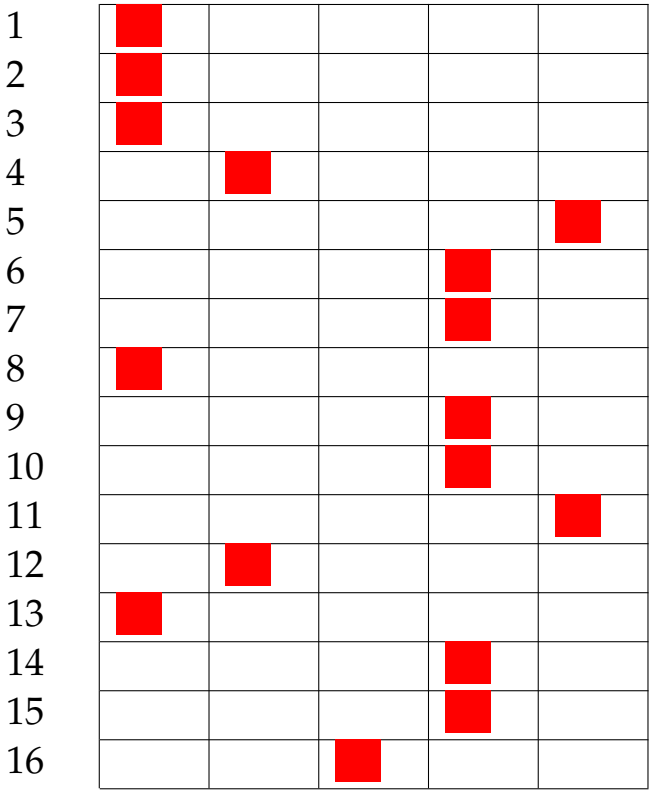
Risultato prova n. 11:



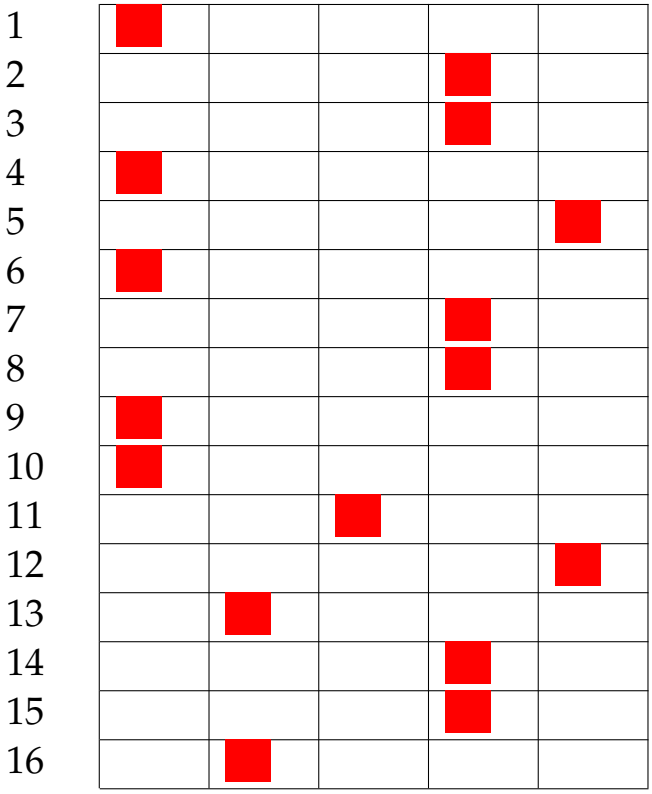
Risultato prova n. 12:



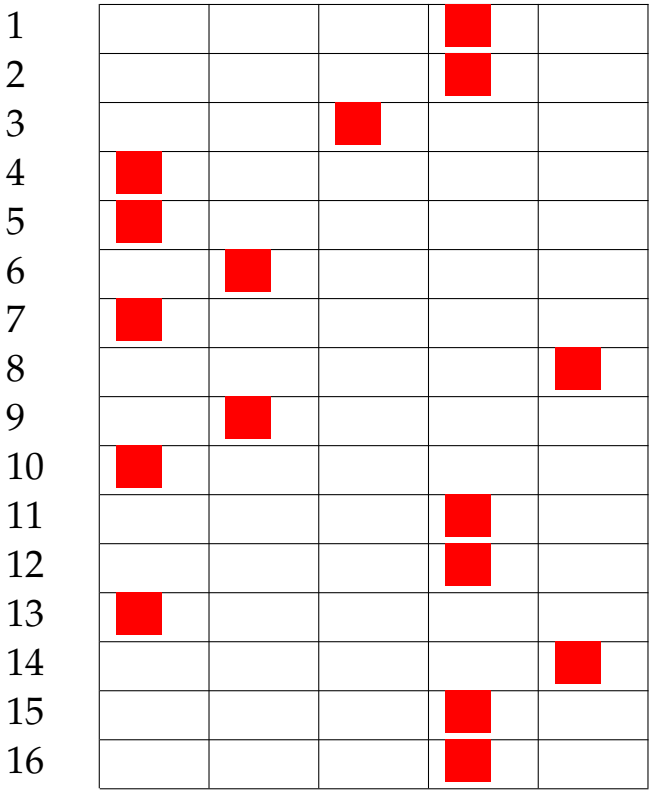
Risultato prova n. 13:



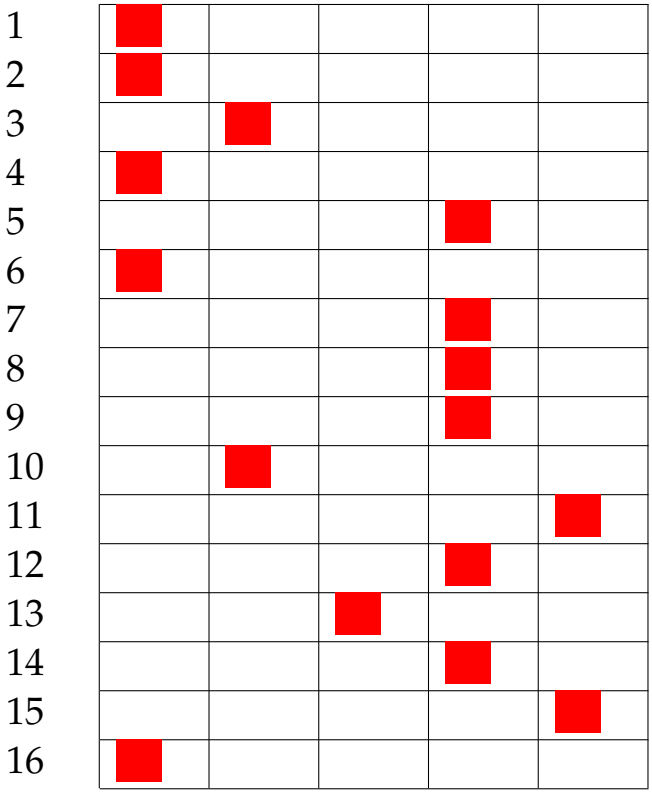
Risultato prova n. 14:



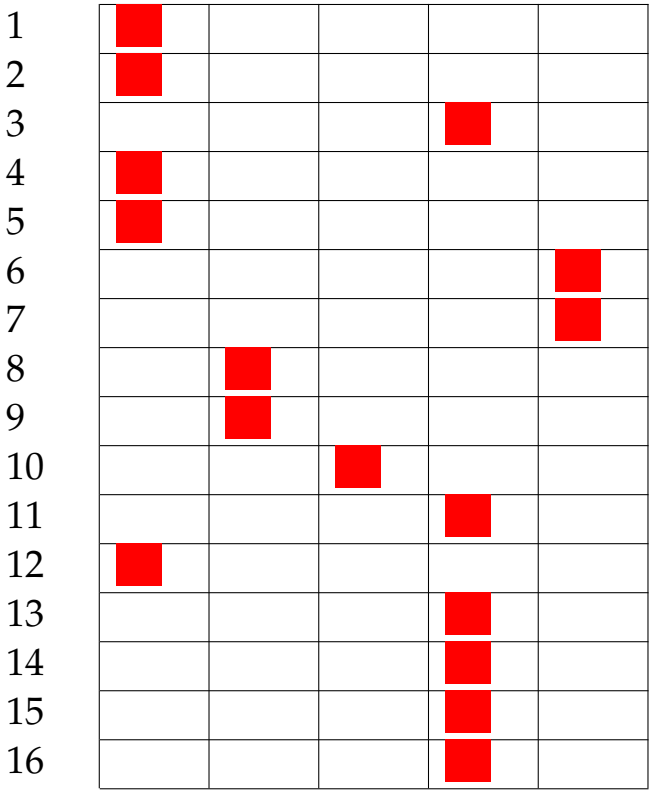
Risultato prova n. 15:



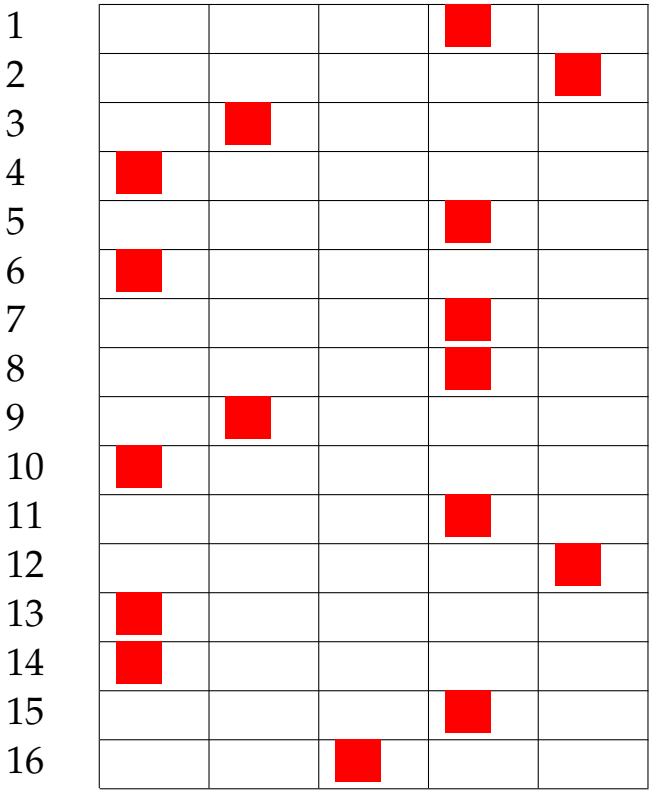
Risultato prova n. 16:



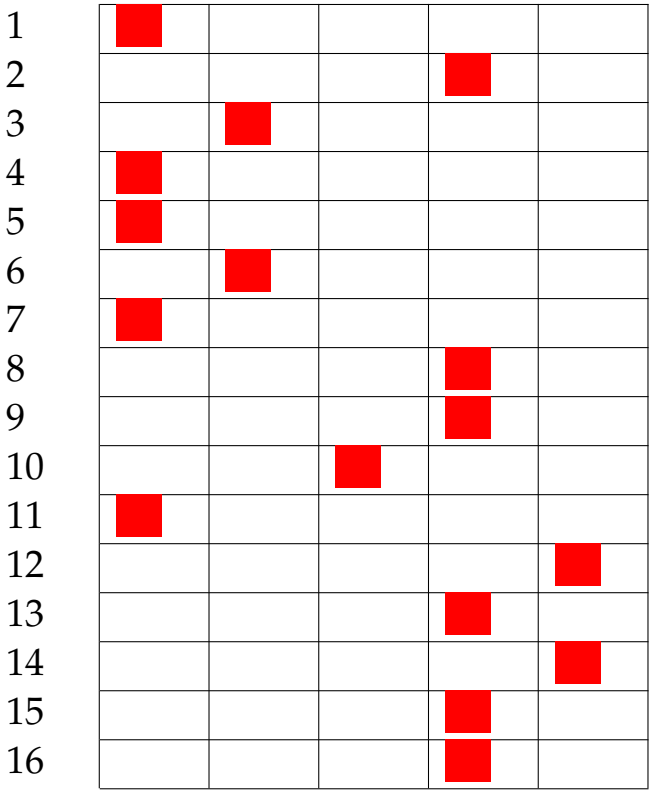
Risultato prova n. 17:



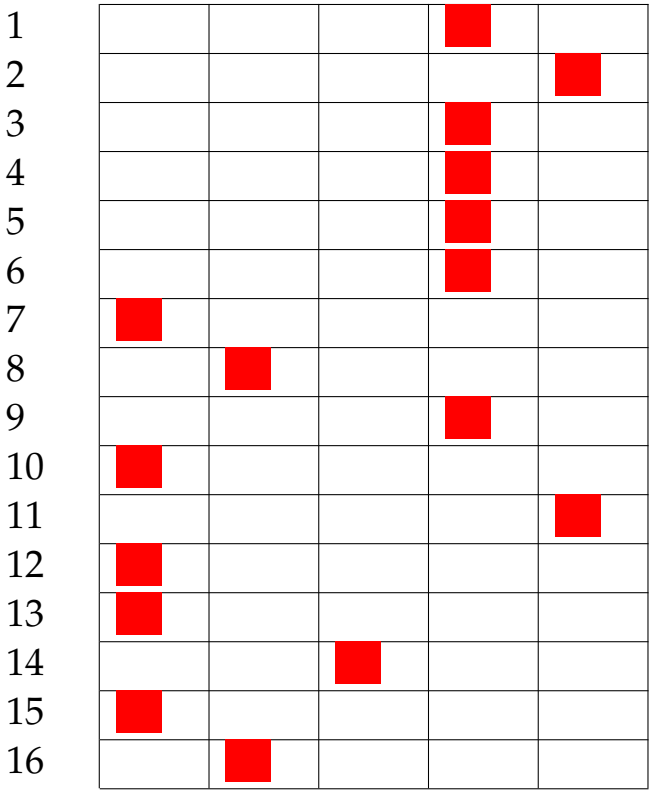
Risultato prova n. 18:



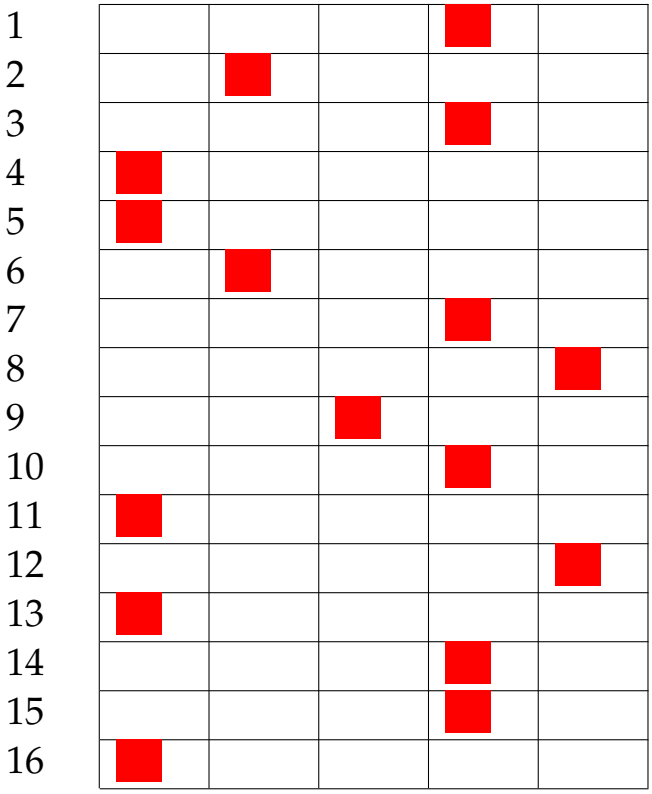
Risultato prova n. 19:



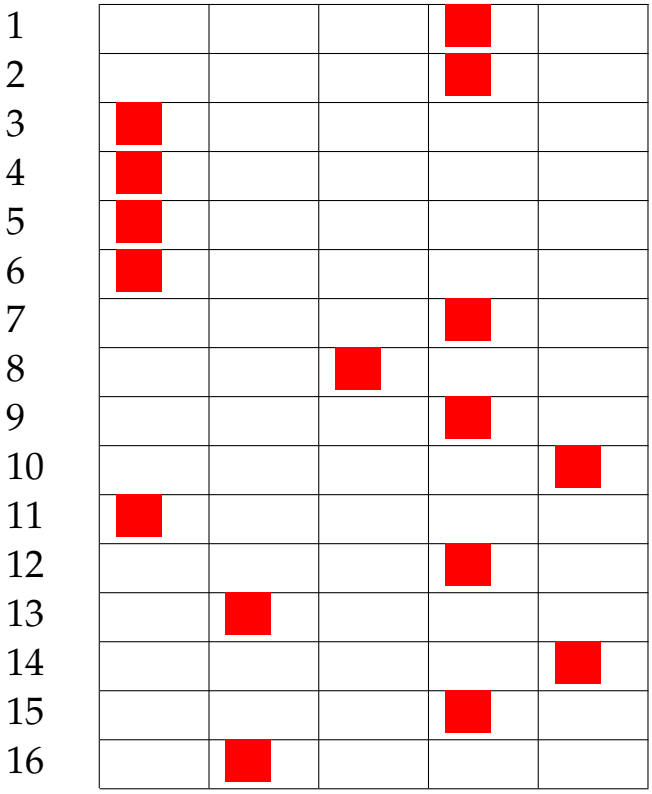
Risultato prova n. 20:



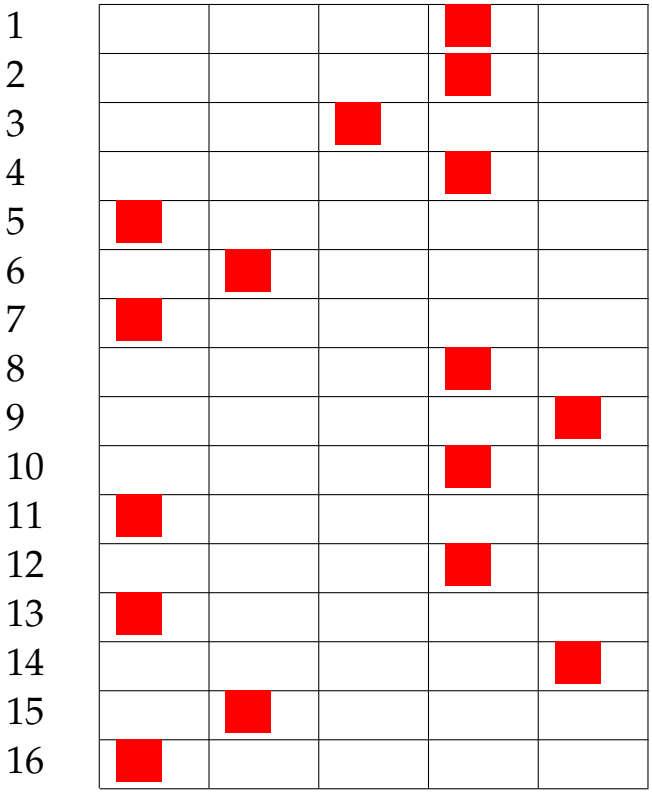
Risultato prova n. 21:



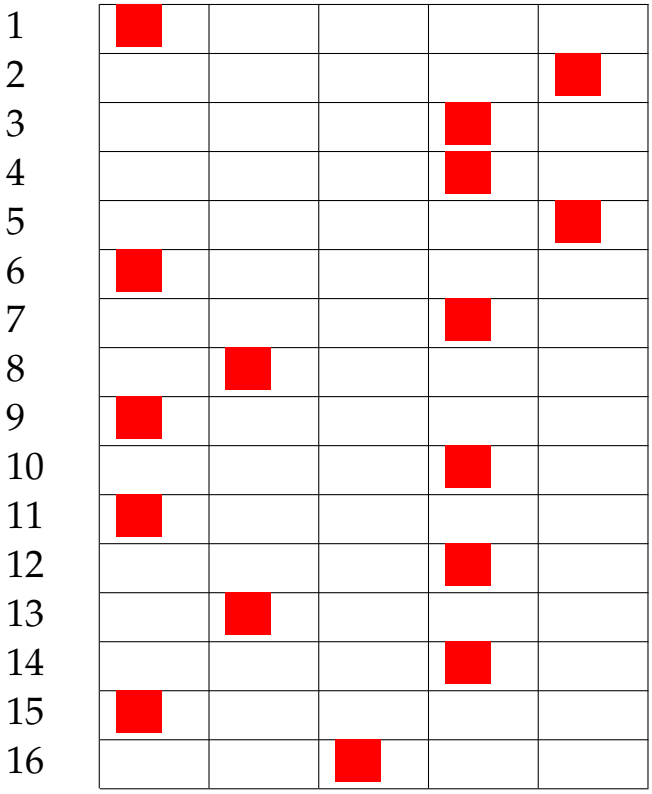
Risultato prova n. 22:



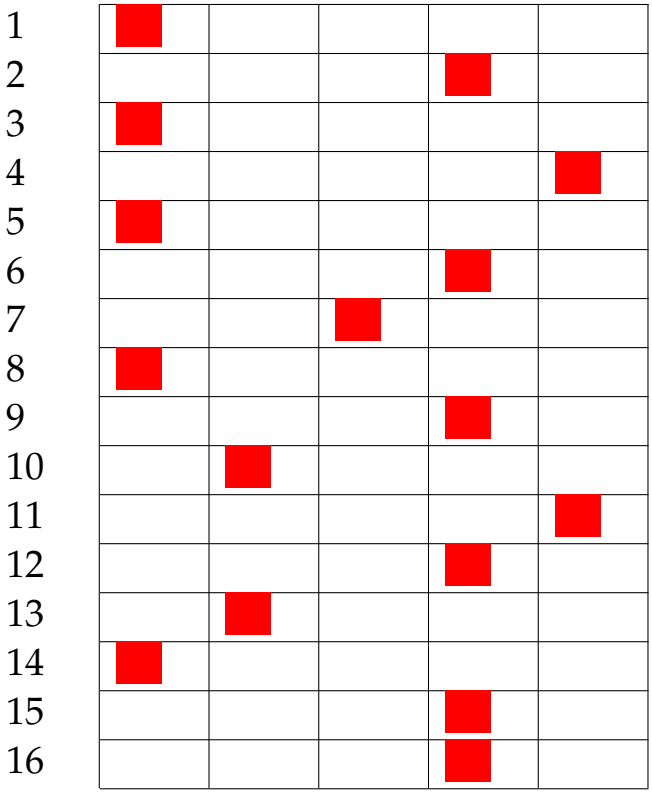
Risultato prova n. 23:



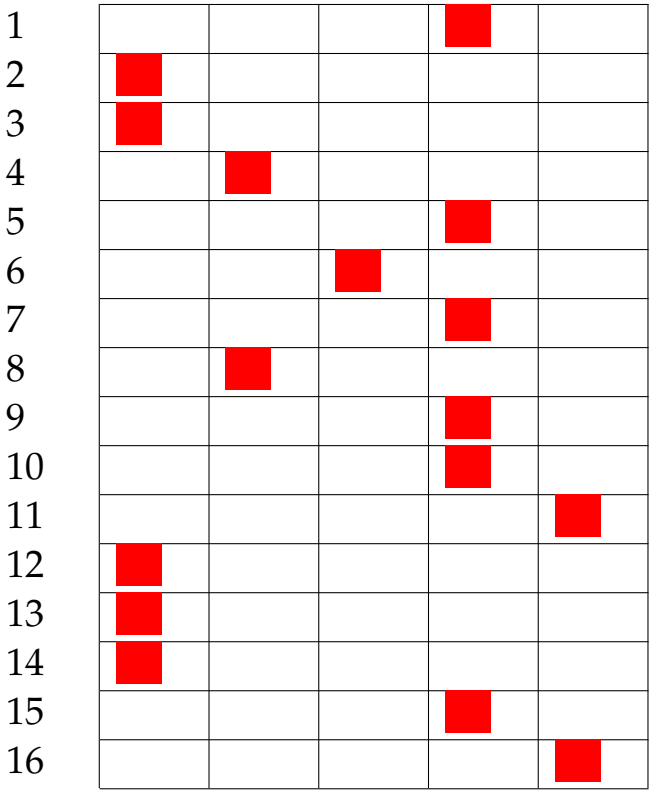
Risultato prova n. 24:



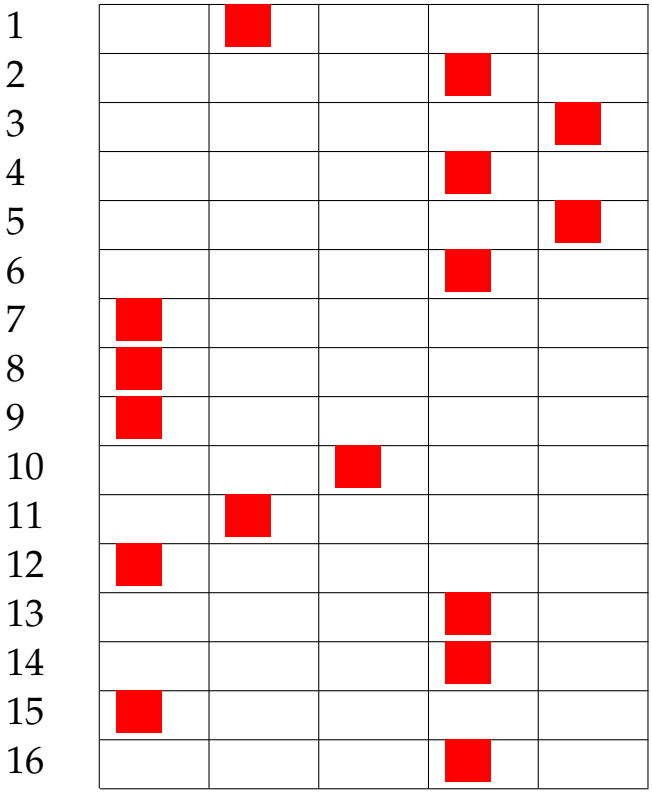
Risultato prova n. 25:



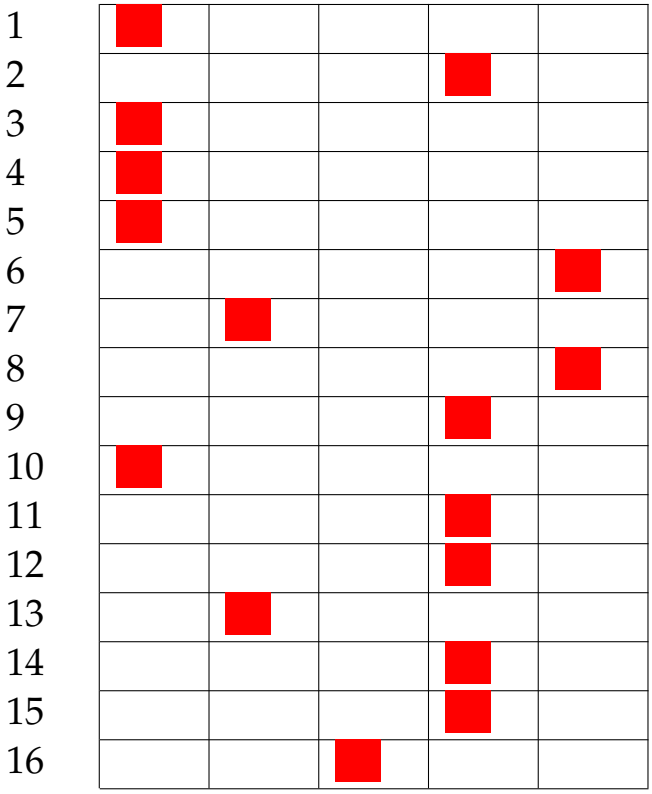
Risultato prova n. 26:



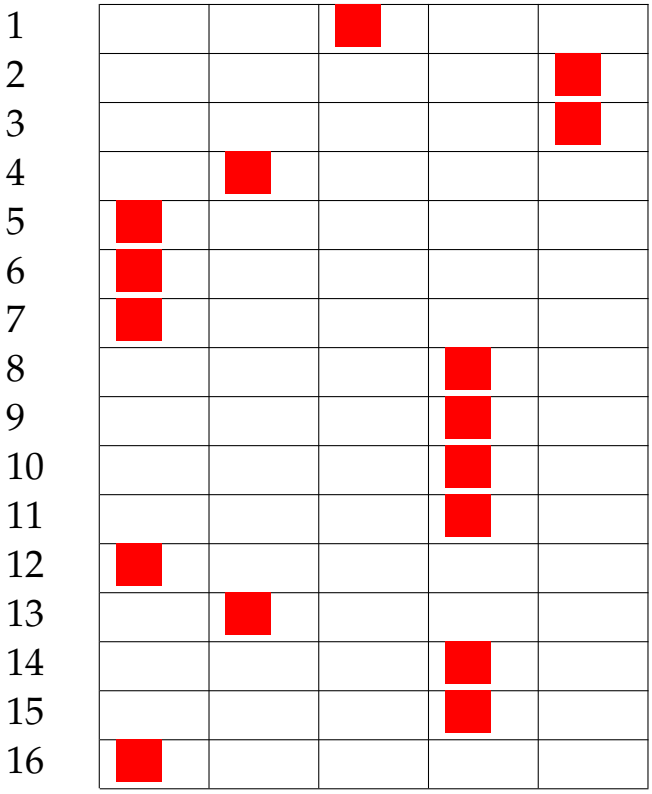
Risultato prova n. 27:



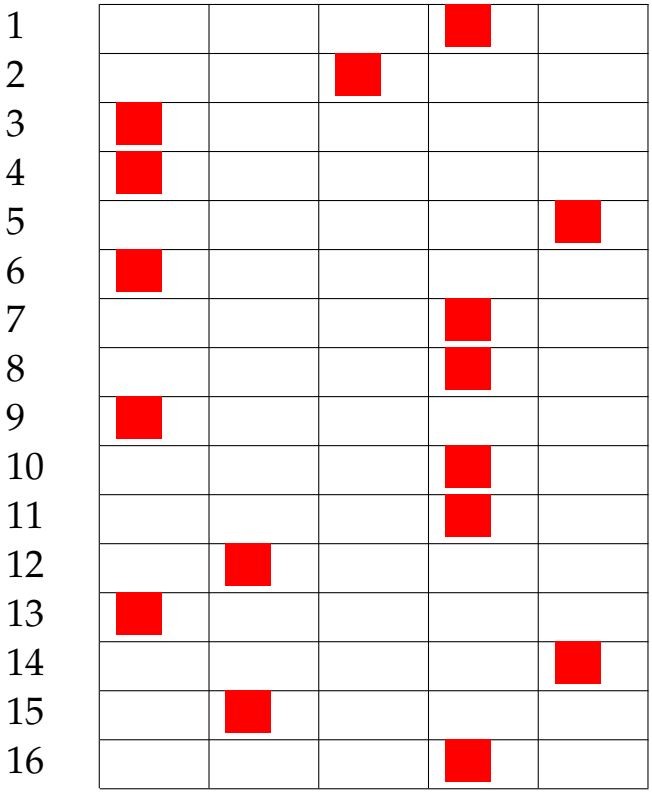
Risultato prova n. 28:



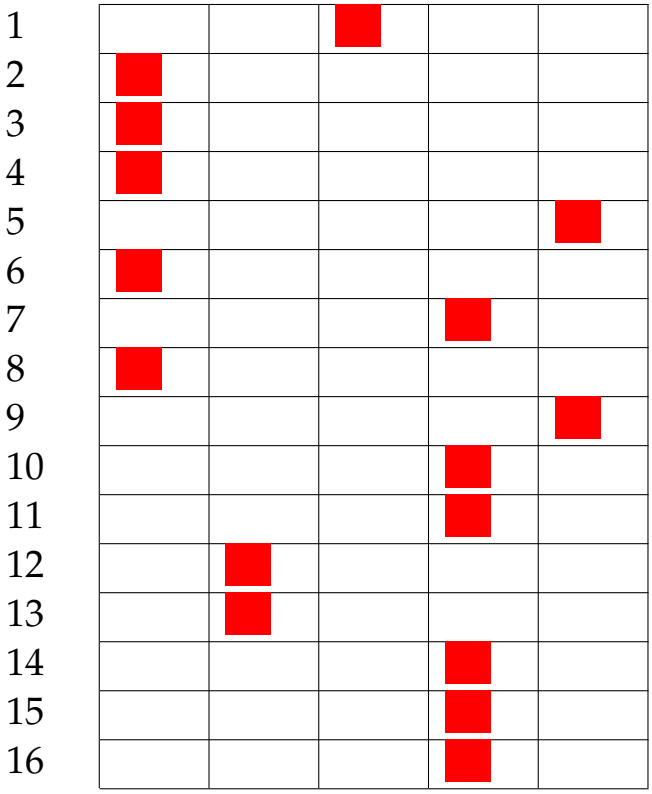
Risultato prova n. 29:



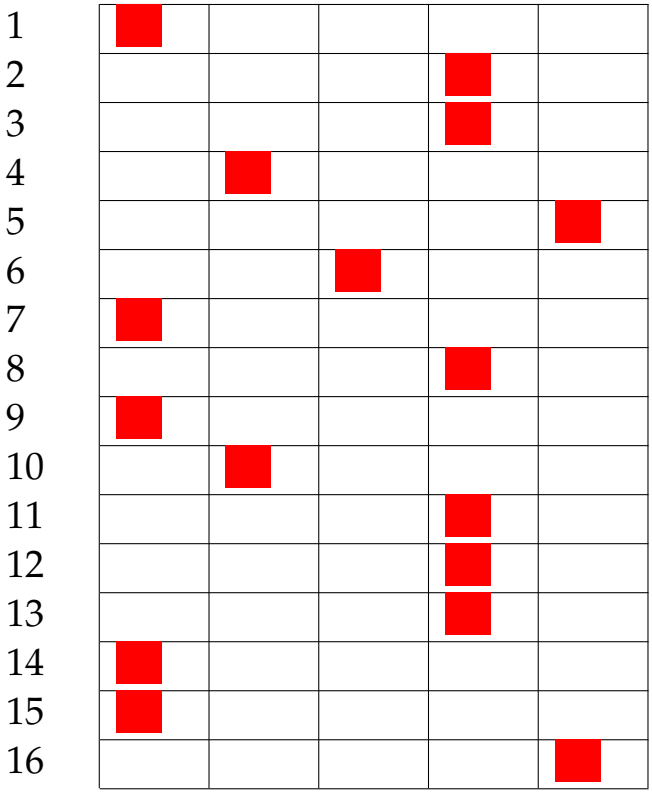
Risultato prova n. 30:



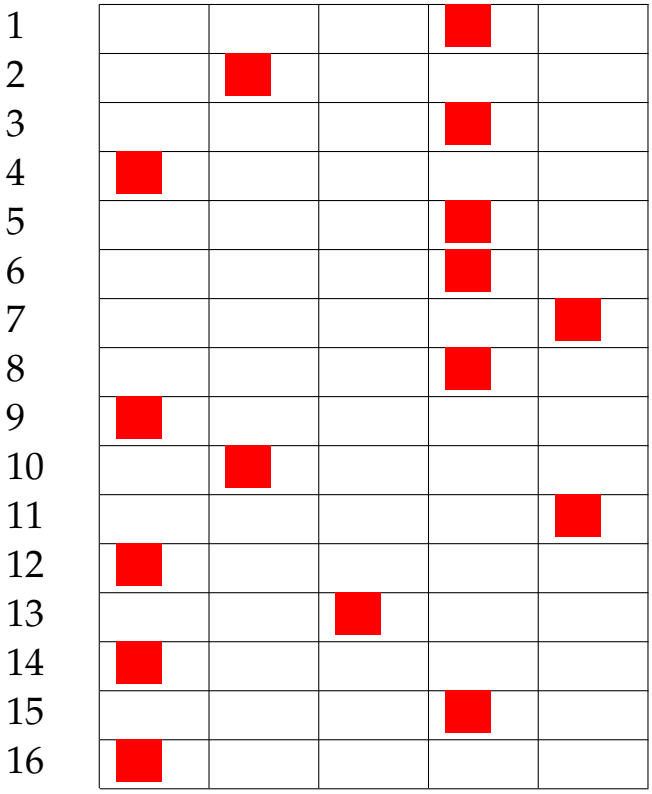
Risultato prova n. 31:



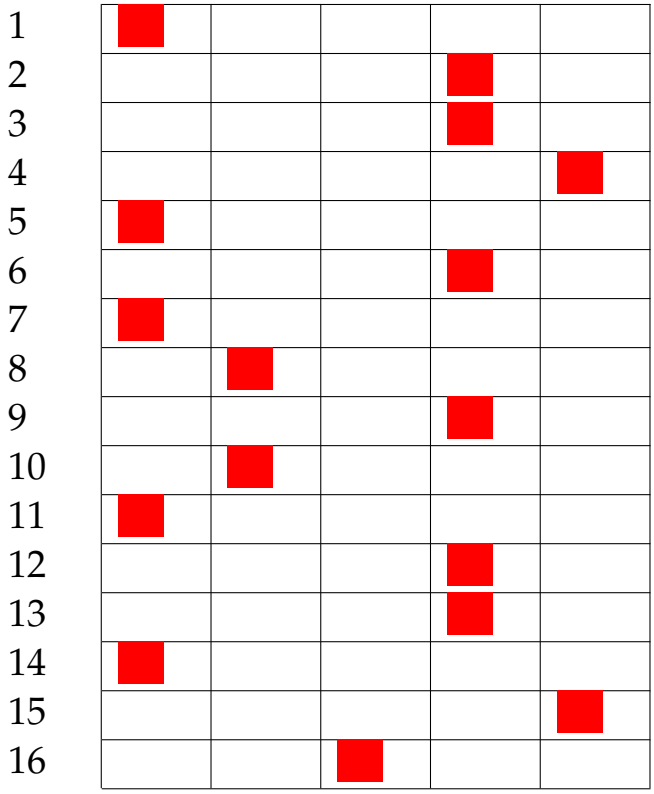
Risultato prova n. 32:



Risultato prova n. 33:



Risultato prova n. 34:



Risultato prova n. 35: