Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

   A. La classe A
   B. La classe B
   C. La classe C
   D. Tutte le classi
   E. Nessuna delle classi

---

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

   A. AB
   B. BB
   C. Errore a tempo di compilazione
   D. ABException in thread main ...
   E. Nessuna delle precedenti

---

3. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

   A. `u = (Integer) c;`
   B. `d = (String) c;`
   C. `d = (String) u;`
   D. `u = (Integer) d;`
   E. Nessuno dei precedenti

---

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`

B. `public String f(byte k)`

C. `protected String f(int k)`

D. `String f(int i) throws Exception`

E. Nessuna delle precedenti.

---

5. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1

B. o2

C. o3

D. o1 e o2

E. Nessuna delle precedenti.

---

6. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341

B. 3... (ciclo infinito)

C. 31

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

10. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

11. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
```

```
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

12. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java
```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java
```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

15. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
```

```
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

16. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

   A. La classe A
   B. La classe B
   C. La classe C
   D. Tutte le classi
   E. Nessuna delle classi

---

2. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

   A. `u = (Integer) c;`
   B. `d = (String) c;`
   C. `d = (String) u;`
   D. `u = (Integer) d;`
   E. Nessuno dei precedenti

---

3. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`
   C. `protected String f(int k)`
   D. `String f(int i) throws Exception`
   E. Nessuna delle precedenti.

---

4. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

   A. `s = (Error []) r;`
   B. `r = (Object [] []) b;`
   C. `r = b;`
   D. `r = (Object [] []) s;`
   E. `r = s;`

---

5. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
```

```
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
```

```
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

10. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

11. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

12. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
```

```
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione

B. Errore a tempo di compilazione

C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

14. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1

B. o2

C. o3

D. o1 e o2

E. Nessuna delle precedenti.

---

15. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
```

```
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. 341

B. 3... (ciclo infinito)

C. 31

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

16. Dire quale delle seguenti affermazioni è falsa:

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

B. Un attributo può essere contemporaneamente static e final

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

D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 3

1. Qual è l'output di questo codice?

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

   A. Errore a tempo di compilazione
   B. 1524Exception in thread main MyExc1
   C. 15
   D. 15243333333... (ciclo infinito)
   E. Nessuna delle precedenti

---

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

   A. AB
   B. BB
   C. Errore a tempo di compilazione
   D. ABException in thread main ...
   E. Nessuna delle precedenti

---

3. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}
```

```
class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

5. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
```

```
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

8. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

9. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

10. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
```

```
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

12. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

14. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

16. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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 programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

   A. AB
   B. BB
   C. Errore a tempo di compilazione
   D. ABException in thread main ...
   E. Nessuna delle precedenti

---

3. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

4. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
```

```
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

5. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

7. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

8. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

9. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C

D. Tutte le classi

E. Nessuna delle classi

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341

B. 3... (ciclo infinito)

C. 31

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

11. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

12. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

15. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

16. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
```

```
    s5 = new String("ab");
    q(s3, b3, b4);
}
void q(String s2, Boolean b1, Boolean b2){
    String s1;
    s1 = new String("abcde");
    if(s2 == s1) {
       System.out.print(1);
    } else {
       System.out.print(0);
    }
    if(b1 == b2) {
       System.out.print(1);
    } else {
       System.out.print(0);
    }
    if(s4 == s5) {
       System.out.print(1);
    } else {
       System.out.print(0);
    }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

   A. o1
   B. o2
   C. o3
   D. o1 e o2
   E. Nessuna delle precedenti.

---

3. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

4. Qual è l'output di questo codice?

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

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

5. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

7. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

9. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

11. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

12. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`

E. Nessuna delle precedenti

---

13. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

14. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

15. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

16. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione

B. Errore a tempo di compilazione

C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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 programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

   A. Nessun errore a tempo di compilazione e di esecuzione
   B. Errore a tempo di compilazione
   C. Errore a tempo di esecuzione alla linea 5 in B.java
   D. Errore a tempo di esecuzione alla linea 6 in B.java
   E. Nessuna delle precedenti

---

3. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

4. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

5. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

6. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

7. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

9. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

10. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

11. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

13. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

14. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

15. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

16. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

   A. o1
   B. o2
   C. o3
   D. o1 e o2
   E. Nessuna delle precedenti.

---

3. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

   A. `private void f( float i)`
   B. `public int f( int j)`
   C. `private void f( int j)`
   D. `public static void f( int j)`
   E. Nessuna delle precedenti

---

4. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

5. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

6. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

7. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

8. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
```

```
        System.out.print(
                super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
            new B().m();
    }
}
```

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

10. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

11. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

12. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
```

```
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

14. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

15. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
```

```
        System.out.print(1);
    }
    catch( MyExc2 e ) {
        throw( new Error() );
    }
    catch( MyExc3 f ) {
        System.out.print(2);
    }
}
static void n() {
    try {
        System.out.print(3);
        throw( new MyExc1() );
    }
    catch( MyExc1 b ) {
        throw( new MyExc1() );
    }
    catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. 341

B. 3... (ciclo infinito)

C. 31

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

2. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

3. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

   A. Nessun errore a tempo di compilazione e di esecuzione
   B. Errore a tempo di compilazione
   C. Errore a tempo di esecuzione alla linea 5 in B.java
   D. Errore a tempo di esecuzione alla linea 6 in B.java
   E. Nessuna delle precedenti

---

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
    }
```

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

5. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

8. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
```

```
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

9. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
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 B1 {
   public static void main(String [] argv)
                       throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

12. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`

B. `r = (Object [] []) b;`

C. `r = b;`

D. `r = (Object [] []) s;`

E. `r = s;`

---

13. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione

B. 1524Exception in thread main MyExc1

C. 15

D. 15243333333... (ciclo infinito)

E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341

B. 3... (ciclo infinito)

C. 31

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
```

```
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args){
        new B().f();
    }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

16. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 9

1. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

   A. o1
   B. o2
   C. o3
   D. o1 e o2
   E. Nessuna delle precedenti.

---

3. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

   A. AB

B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

4. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

5. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

6. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

7. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

9. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

10. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
```

```
        }
        catch( MyExc2 e ) {
           throw( new Error() );
        }
        catch( MyExc3 f ) {
           System.out.print(2);
        }
    }
    static void n() {
        try {
           System.out.print(3);
           throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
           throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
           System.out.print(4);
        }
    }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

12. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

13. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

14. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

15. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
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 B1 {
   public static void main(String [] argv)
                       throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

4. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }
```

B.java

```
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione

B. Errore a tempo di compilazione

C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

5. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A

B. La classe B

C. La classe C

D. Tutte le classi

E. Nessuna delle classi

---

6. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341

B. 3... (ciclo infinito)

C. 31

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

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

```
      }
      catch( MyExc3 j ) {
      }
      catch( MyExc2 x ) {
      }
      finally {
         System.out.print(4);
         throw( new MyExc1() );
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc2() );
      }
      catch( Exception s ) {
      }
   }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

8. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

9. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

11. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

12. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

13. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

14. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

15. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
```

```
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

3. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}
```

```
abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

4. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

5. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

7. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

8. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

9. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

10. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

11. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java
```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java
```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

12. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

14. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
```

```
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
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 B1 {
   public static void main(String [] argv)
                       throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

3. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

5. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

6. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

7. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

8. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

9. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

11. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

12. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

13. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

15. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java
```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java
```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

16. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 13

1. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Qual è l'output di questo codice?

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

   A. 14Exception in thread main MyExc3
   B. Errore a tempo di compilazione
   C. 1Exception in thread main MyExc3
   D. 1... (ciclo infinito)
   E. Nessuna delle precedenti

---

3. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
```

```
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

5. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

6. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

7. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

8. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

10. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

11. Qual è l'output di questo codice?

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

```
            throw( new MyExc1() );
         }
      }
      static void q() throws Exception {
         try {
            System.out.print(5);
            throw( new MyExc2() );
         }
         catch( Exception s ) {
         }
      }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

12. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

13. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

14. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

15. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Qual è l'output di questo codice?

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

   A. Errore a tempo di compilazione
   B. 1524Exception in thread main MyExc1
   C. 15
   D. 15243333333... (ciclo infinito)
   E. Nessuna delle precedenti

---

2. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

   A. La classe A
   B. La classe B
   C. La classe C
   D. Tutte le classi
   E. Nessuna delle classi

---

3. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

   A. `s = (Error []) r;`
   B. `r = (Object [] []) b;`
   C. `r = b;`
   D. `r = (Object [] []) s;`
   E. `r = s;`

---

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

5. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

6. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

7. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
```

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

10. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
```

```
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

14. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

16. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
```

```
private String s4 = new String("ab");
private String s5;
void p() {
    String s3;
    s3 = new String("abcde");
    b4 = b3;
    s5 = new String("ab");
    q(s3, b3, b4);
}
void q(String s2, Boolean b1, Boolean b2){
    String s1;
    s1 = new String("abcde");
    if(s2 == s1) {
       System.out.print(1);
    } else {
       System.out.print(0);
    }
    if(b1 == b2) {
       System.out.print(1);
    } else {
       System.out.print(0);
    }
    if(s4 == s5) {
       System.out.print(1);
    } else {
       System.out.print(0);
    }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

   A. La classe A
   B. La classe B
   C. La classe C
   D. Tutte le classi
   E. Nessuna delle classi

---

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

   A. AB
   B. BB
   C. Errore a tempo di compilazione
   D. ABException in thread main ...
   E. Nessuna delle precedenti

---

3. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

   A. `private void f( float i)`
   B. `public int f( int j)`
   C. `private void f( int j)`
   D. `public static void f( int j)`
   E. Nessuna delle precedenti

---

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`
   C. `protected String f(int k)`
   D. `String f(int i) throws Exception`

E. Nessuna delle precedenti.

---

5. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

6. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

7. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

8. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

10. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione

B. 1524Exception in thread main MyExc1

C. 15

D. 15243333333... (ciclo infinito)

E. Nessuna delle precedenti

---

11. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`

B. `d = (String) c;`

C. `d = (String) u;`

D. `u = (Integer) d;`

E. Nessuno dei precedenti

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
```

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

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

13. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

14. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

16. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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 programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

   A. AB
   B. BB
   C. Errore a tempo di compilazione
   D. ABException in thread main ...
   E. Nessuna delle precedenti

---

2. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`
   C. `protected String f(int k)`
   D. `String f(int i) throws Exception`
   E. Nessuna delle precedenti.

---

3. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

4. Qual è l'output di questo codice?

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

```
      catch( MyExc3 j ) {
      }
      catch( MyExc2 x ) {
      }
      finally {
         System.out.print(4);
         throw( new MyExc1() );
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc2() );
      }
      catch( Exception s ) {
      }
   }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

5. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java
```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java
```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
```

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

10. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

11. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

13. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

14. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

15. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Se nella classe C è dichiarato il metodo:

   ```
   String f(int i)
   ```

   indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`
   C. `protected String f(int k)`
   D. `String f(int i) throws Exception`
   E. Nessuna delle precedenti.

---

2. Qual è l'output di questo codice?

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

   A. Errore a tempo di compilazione
   B. 1524Exception in thread main MyExc1
   C. 15
   D. 15243333333... (ciclo infinito)
   E. Nessuna delle precedenti

---

3. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

   A. Nessun errore a tempo di compilazione e di esecuzione
   B. Errore a tempo di compilazione
   C. Errore a tempo di esecuzione alla linea 5 in B.java
   D. Errore a tempo di esecuzione alla linea 6 in B.java
   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 B1 {
   public static void main(String [] argv)
                       throws Exception {
```

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

5. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

6. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
```

```
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

10. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

11. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

12. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

14. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

15. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
    }
```

```
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
            new B().m();
    }
}
```

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

3. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
```

```
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

4. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

5. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

6. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

7. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`

D. `String f(int i) throws Exception`

E. Nessuna delle precedenti.

---

8. Dire quale delle seguenti affermazioni è falsa:

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

   B. Un attributo può essere contemporaneamente static e final

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

   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

9. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

   A. `u = (Integer) c;`

   B. `d = (String) c;`

   C. `d = (String) u;`

   D. `u = (Integer) d;`

   E. Nessuno dei precedenti

---

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

   A. AB

   B. BB

   C. Errore a tempo di compilazione

   D. ABException in thread main ...

   E. Nessuna delle precedenti

---

11. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

   A. AB

   B. BB

   C. Errore a tempo di compilazione

   D. ABException in thread main ...

   E. Nessuna delle precedenti

---

12. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

13. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

14. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
```

```
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

16. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

2. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

3. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
```

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

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

5. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

6. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}
```

```
abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

8. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

9. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

11. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

12. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

13. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

14. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

15. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

16. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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 programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

   A. AB
   B. BB
   C. Errore a tempo di compilazione
   D. ABException in thread main ...
   E. Nessuna delle precedenti

---

3. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

   A. Nessun errore a tempo di compilazione e di esecuzione
   B. Errore a tempo di compilazione
   C. Errore a tempo di esecuzione alla linea 5 in B.java
   D. Errore a tempo di esecuzione alla linea 6 in B.java
   E. Nessuna delle precedenti

---

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

5. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

6. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
```

```
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
   }
   static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

10. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

11. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`

B. `public String f(byte k)`

C. `protected String f(int k)`

D. `String f(int i) throws Exception`

E. Nessuna delle precedenti.

---

12. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

13. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione

B. 1524Exception in thread main MyExc1

C. 15

D. 15243333333... (ciclo infinito)

E. Nessuna delle precedenti

---

14. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`

B. `public int f( int j)`

C. `private void f( int j)`

D. `public static void f( int j)`

E. Nessuna delle precedenti

---

15. Dire quale delle seguenti affermazioni è falsa:

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

B. Un attributo può essere contemporaneamente static e final

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

D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

16. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

   A. o1
   B. o2
   C. o3
   D. o1 e o2
   E. Nessuna delle precedenti.

---

2. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`
   C. `protected String f(int k)`
   D. `String f(int i) throws Exception`
   E. Nessuna delle precedenti.

---

3. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

   A. La classe A
   B. La classe B
   C. La classe C
   D. Tutte le classi
   E. Nessuna delle classi

---

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
```

```
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

5. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

6. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
```

```
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

10. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

11. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

12. Qual è l'output di questo codice?

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

```
            System.out.print(4);
            throw( new MyExc1() );
         }
      }
      static void q() throws Exception {
         try {
            System.out.print(5);
            throw( new MyExc2() );
         }
         catch( Exception s ) {
         }
      }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

14. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.    private int i = 1;
 4.    int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.    public static void main(String[] args){
 5.      int a = new A().getI();
 6.      int b = new B().getI();
 7.    }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

15. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

16. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

   A. `private void f( float i)`
   B. `public int f( int j)`
   C. `private void f( int j)`
   D. `public static void f( int j)`
   E. Nessuna delle precedenti

---

3. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`
   C. `protected String f(int k)`
   D. `String f(int i) throws Exception`
   E. Nessuna delle precedenti.

---

4. Qual è l'output di questo codice?

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

   A. Errore a tempo di compilazione
   B. 1524Exception in thread main MyExc1
   C. 15
   D. 15243333333... (ciclo infinito)
   E. Nessuna delle precedenti

---

5. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

6. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
```

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
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 B1 {
  public static void main(String [] argv)
                      throws Exception {
     try {
        System.out.print(1);
        m();
        System.out.print(2);
     }
     catch( MyExc2 g ) {
        System.out.print(3);
     }
  }
  static void m() throws Exception {
     try {
        throw( new MyExc3() );
     }
     catch( MyExc3 c ) {
        throw( new MyExc3() );
     }
     finally {
        System.out.print(4);
     }
  }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

11. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

13. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

16. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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 programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

   A. AB
   B. BB
   C. Errore a tempo di compilazione
   D. ABException in thread main ...
   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 B1 {
   public static void main(String [] argv)
                       throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

   A. 14Exception in thread main MyExc3

B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

4. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

5. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

7. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

8. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

9. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
```

```
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

12. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

13. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

14. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java
```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java
```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

15. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
```

```
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341

B. 3... (ciclo infinito)

C. 31

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

16. Dire quale delle seguenti affermazioni è falsa:

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

B. Un attributo può essere contemporaneamente static e final

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

D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

3. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
```

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

5. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

6. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

7. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

8. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

9. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

10. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

11. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

12. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

13. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

15. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}
```

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

16. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

   A. o1
   B. o2
   C. o3
   D. o1 e o2
   E. Nessuna delle precedenti.

---

2. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

3. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`
   C. `protected String f(int k)`
   D. `String f(int i) throws Exception`
   E. Nessuna delle precedenti.

---

4. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

5. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

6. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}
```

```
class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
```

```
            n();
         }
         catch( MyExc1 z ) {
            System.out.print(1);
         }
         catch( MyExc2 e ) {
            throw( new Error() );
         }
         catch( MyExc3 f ) {
            System.out.print(2);
         }
    }
    static void n() {
         try {
            System.out.print(3);
            throw( new MyExc1() );
         }
         catch( MyExc1 b ) {
            throw( new MyExc1() );
         }
         catch( MyExc3 i ) {
         }
         catch( MyExc2 v ) {
         }
         finally {
            System.out.print(4);
         }
    }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

10. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
```

```
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

12. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

14. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

15. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

16. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione

B. Errore a tempo di compilazione

C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

   A. AB
   B. BB
   C. Errore a tempo di compilazione
   D. ABException in thread main ...
   E. Nessuna delle precedenti

---

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

   A. `private void f( float i)`
   B. `public int f( int j)`
   C. `private void f( int j)`
   D. `public static void f( int j)`
   E. Nessuna delle precedenti

---

3. Qual è l'output di questo codice?

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

   A. Errore a tempo di compilazione
   B. 1524Exception in thread main MyExc1
   C. 15
   D. 15243333333... (ciclo infinito)
   E. Nessuna delle precedenti

---

4. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

5. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

6. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

7. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

9. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

10. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
```

```
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

12. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

13. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

14. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

16. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 341

B. 3... (ciclo infinito)

C. 31

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

   A. Errore a tempo di compilazione
   B. 1524Exception in thread main MyExc1
   C. 15
   D. 15243333333... (ciclo infinito)
   E. Nessuna delle precedenti

---

2. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

`A.java`

```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

`B.java`

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

   A. Nessun errore a tempo di compilazione e di esecuzione
   B. Errore a tempo di compilazione
   C. Errore a tempo di esecuzione alla linea 5 in B.java
   D. Errore a tempo di esecuzione alla linea 6 in B.java
   E. Nessuna delle precedenti

---

3. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

   A. `private void f( float i)`
   B. `public int f( int j)`
   C. `private void f( int j)`
   D. `public static void f( int j)`
   E. Nessuna delle precedenti

---

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`

B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

5. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

6. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

7. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

8. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
```

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

A. 001
B. 101
C. 010
D. 000
E. 111

---

9. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

10. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

11. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

12. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

13. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

14. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

15. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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 delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

   A. La classe A
   B. La classe B
   C. La classe C
   D. Tutte le classi
   E. Nessuna delle classi

---

2. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

   A. `u = (Integer) c;`
   B. `d = (String) c;`
   C. `d = (String) u;`
   D. `u = (Integer) d;`
   E. Nessuno dei precedenti

---

3. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

   A. `s = (Error []) r;`
   B. `r = (Object [] []) b;`
   C. `r = b;`
   D. `r = (Object [] []) s;`
   E. `r = s;`

---

4. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
```

```
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

5. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

6. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

7. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

8. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

13. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3

B. Errore a tempo di compilazione

C. 1Exception in thread main MyExc3

D. 1... (ciclo infinito)

E. Nessuna delle precedenti

---

14. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

15. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

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

```
   }
   catch( Exception h ) {
      System.out.print(3);
      throw( new MyExc1() );
   }
   catch( MyExc3 j ) {
   }
   catch( MyExc2 x ) {
   }
   finally {
      System.out.print(4);
      throw( new MyExc1() );
   }
}
static void q() throws Exception {
   try {
      System.out.print(5);
      throw( new MyExc2() );
   }
   catch( Exception s ) {
   }
  }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Se nella classe C è dichiarato il metodo:

   ```
   String f(int i)
   ```

   indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`
   C. `protected String f(int k)`
   D. `String f(int i) throws Exception`
   E. Nessuna delle precedenti.

---

2. Qual è l'output di questo codice?

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

   A. Errore a tempo di compilazione
   B. 1524Exception in thread main MyExc1
   C. 15
   D. 15243333333... (ciclo infinito)
   E. Nessuna delle precedenti

---

3. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

   A. 001
   B. 101
   C. 010
   D. 000

E. 111

---

4. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`

B. `r = (Object [] []) b;`

C. `r = b;`

D. `r = (Object [] []) s;`

E. `r = s;`

---

5. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

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

A. 14Exception in thread main MyExc3

B. Errore a tempo di compilazione

C. 1Exception in thread main MyExc3

D. 1... (ciclo infinito)

E. Nessuna delle precedenti

---

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
```

```
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

8. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

9. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

10. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

11. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

12. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

14. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`

D. `u = (Integer) d;`

E. Nessuno dei precedenti

---

16. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java
```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java
```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione

B. Errore a tempo di compilazione

C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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 programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

   A. AB
   B. BB
   C. Errore a tempo di compilazione
   D. ABException in thread main ...
   E. Nessuna delle precedenti

---

3. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

   A. `u = (Integer) c;`
   B. `d = (String) c;`
   C. `d = (String) u;`
   D. `u = (Integer) d;`
   E. Nessuno dei precedenti

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
```

```
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

5. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

6. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

8. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

9. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

10. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

13. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

14. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

15. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

16. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 001

B. 101

C. 010

D. 000

E. 111

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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 programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`
   C. `protected String f(int k)`
   D. `String f(int i) throws Exception`
   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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

   A. 14Exception in thread main MyExc3
   B. Errore a tempo di compilazione
   C. 1Exception in thread main MyExc3
   D. 1... (ciclo infinito)
   E. Nessuna delle precedenti

---

4. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

   A. `private void f( float i)`
   B. `public int f( int j)`
   C. `private void f( int j)`
   D. `public static void f( int j)`
   E. Nessuna delle precedenti

5. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

6. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
```

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

9. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java
```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java
```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

11. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

12. Qual è l'output di questo codice?

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

```
      catch( Exception h ) {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc3 j ) {
      }
      catch( MyExc2 x ) {
      }
      finally {
         System.out.print(4);
         throw( new MyExc1() );
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc2() );
      }
      catch( Exception s ) {
      }
   }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

13. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

14. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

15. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
```

```
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

3. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

4. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`

D. `public static void f( int j)`

E. Nessuna delle precedenti

---

5. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`

B. `d = (String) c;`

C. `d = (String) u;`

D. `u = (Integer) d;`

E. Nessuno dei precedenti

---

6. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione

B. 1524Exception in thread main MyExc1

C. 15

D. 15243333333... (ciclo infinito)

E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
```

```
                super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
            new B().m();
    }
}
```

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

9. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

10. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

11. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

12. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

13. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`

C. `r = b;`

D. `r = (Object [] []) s;`

E. `r = s;`

---

16. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Dire quale delle seguenti affermazioni è falsa:

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

   B. Un attributo può essere contemporaneamente static e final

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

   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

   A. 341

   B. 3... (ciclo infinito)

   C. 31

   D. Errore a tempo di compilazione

   E. Nessuna delle precedenti

---

3. Qual è l'output di questo codice?

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

```
   }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

4. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

5. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

8. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

9. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

10. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
```

```
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

11. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

12. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

13. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

15. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
```

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

   A. `private void f( float i)`
   B. `public int f( int j)`
   C. `private void f( int j)`
   D. `public static void f( int j)`
   E. Nessuna delle precedenti

---

3. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

   A. 001
   B. 101
   C. 010
   D. 000
   E. 111

---

4. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

   A. `s = (Error []) r;`
   B. `r = (Object [] []) b;`
   C. `r = b;`
   D. `r = (Object [] []) s;`
   E. `r = s;`

---

5. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

8. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
```

```
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

9. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

10. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

12. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
```

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

14. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

16. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione

B. 1524Exception in thread main MyExc1

C. 15

D. 15243333333... (ciclo infinito)

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

2. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
```

```
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
            new B().m();
    }
}
```

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

5. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

6. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

7. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

8. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

9. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

10. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

11. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
```

```
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

12. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

13. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

14. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

15. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

16. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 36

1. Qual è l'output di questo codice?

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

   A. Errore a tempo di compilazione
   B. 1524Exception in thread main MyExc1
   C. 15
   D. 15243333333... (ciclo infinito)
   E. Nessuna delle precedenti

---

2. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

   A. 001
   B. 101
   C. 010
   D. 000
   E. 111

---

3. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

   A. `private void f( float i)`
   B. `public int f( int j)`
   C. `private void f( int j)`
   D. `public static void f( int j)`

E. Nessuna delle precedenti

---

4. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

5. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

6. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
    }
    public void g() {
        super.g();
```

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

9. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

10. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

11. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java
```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java
```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

12. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

14. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

15. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 37

1. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

   A. La classe A
   B. La classe B
   C. La classe C
   D. Tutte le classi
   E. Nessuna delle classi

---

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

   A. AB
   B. BB
   C. Errore a tempo di compilazione
   D. ABException in thread main ...
   E. Nessuna delle precedenti

---

3. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
```

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

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

4. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

5. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

6. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

7. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

8. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
```

```
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

12. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

15. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

16. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 38

1. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

2. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

3. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}
```

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
```

```
        this.s = s;
    }
    public String getS() {
        return s;
    }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

8. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

9. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

10. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
```

```
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

14. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

15. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
```

```
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione

B. Errore a tempo di compilazione

C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

16. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`

B. `public String f(byte k)`

C. `protected String f(int k)`

D. `String f(int i) throws Exception`

E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 39

1. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
```

```
      }
   }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

4. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

5. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

6. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

7. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

9. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}
```

```
class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

10. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

11. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

12. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java
```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java
```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

14. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

15. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 40

1. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

2. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

3. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

4. Dire quale delle seguenti affermazioni è falsa:

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

B. Un attributo può essere contemporaneamente static e final

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

D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

5. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

6. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`

B. `public int f( int j)`

C. `private void f( int j)`

D. `public static void f( int j)`

E. Nessuna delle precedenti

---

7. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A

B. La classe B

C. La classe C

D. Tutte le classi

E. Nessuna delle classi

---

8. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

10. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

11. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

13. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

16. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 41

1. Qual è l'output di questo codice?

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

   A. 14Exception in thread main MyExc3
   B. Errore a tempo di compilazione
   C. 1Exception in thread main MyExc3
   D. 1... (ciclo infinito)
   E. Nessuna delle precedenti

---

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

   A. `private void f( float i)`
   B. `public int f( int j)`
   C. `private void f( int j)`
   D. `public static void f( int j)`
   E. Nessuna delle precedenti

---

3. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

   A. o1
   B. o2
   C. o3
   D. o1 e o2
   E. Nessuna delle precedenti.

---

4. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

5. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

6. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

8. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

9. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

10. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

11. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101

C. 010

D. 000

E. 111

---

12. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`

B. `d = (String) c;`

C. `d = (String) u;`

D. `u = (Integer) d;`

E. Nessuno dei precedenti

---

13. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A

B. La classe B

C. La classe C

D. Tutte le classi

E. Nessuna delle classi

---

14. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

15. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

16. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione

B. Errore a tempo di compilazione

C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 42

1. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

2. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

3. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

4. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

5. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
```

```
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args){
        new B().f();
    }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

8. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

9. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

10. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

11. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

12. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

13. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

15. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java
```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java
```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione

C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

16. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 43

1. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

2. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

3. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

5. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

7. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

8. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

9. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

11. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
```

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

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

14. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

15. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione

B. 1524Exception in thread main MyExc1

C. 15

D. 15243333333... (ciclo infinito)

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 44

1. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`
   C. `protected String f(int k)`
   D. `String f(int i) throws Exception`
   E. Nessuna delle precedenti.

---

3. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

   A. `s = (Error []) r;`
   B. `r = (Object [] []) b;`
   C. `r = b;`
   D. `r = (Object [] []) s;`
   E. `r = s;`

---

4. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

   A. Nessun errore a tempo di compilazione e di esecuzione
   B. Errore a tempo di compilazione
   C. Errore a tempo di esecuzione alla linea 5 in B.java
   D. Errore a tempo di esecuzione alla linea 6 in B.java
   E. Nessuna delle precedenti

---

5. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

6. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

8. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final

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

D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

9. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`

B. `public int f( int j)`

C. `private void f( int j)`

D. `public static void f( int j)`

E. Nessuna delle precedenti

---

10. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1

B. o2

C. o3

D. o1 e o2

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

A. 14Exception in thread main MyExc3

B. Errore a tempo di compilazione

C. 1Exception in thread main MyExc3

D. 1... (ciclo infinito)

E. Nessuna delle precedenti

---

12. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
    }
```

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

13. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

14. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

15. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

16. Qual è l'output di questo codice?

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

```
      System.out.print(2);
   }
   catch( Exception h ) {
      System.out.print(3);
      throw( new MyExc1() );
   }
   catch( MyExc3 j ) {
   }
   catch( MyExc2 x ) {
   }
   finally {
      System.out.print(4);
      throw( new MyExc1() );
   }
}
static void q() throws Exception {
   try {
         System.out.print(5);
         throw( new MyExc2() );
      }
      catch( Exception s ) {
      }
   }
}
```

A. Errore a tempo di compilazione

B. 1524Exception in thread main MyExc1

C. 15

D. 15243333333... (ciclo infinito)

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 45

1. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

2. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

3. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
```

```
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

4. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

5. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1

B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

10. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

11. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
    }
    public void g() {
```

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

12. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

14. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

16. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 46

1. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

3. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

4. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`

B. `r = (Object [] []) b;`

C. `r = b;`

D. `r = (Object [] []) s;`

E. `r = s;`

---

5. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

6. Dire quale delle seguenti affermazioni è falsa:

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

B. Un attributo può essere contemporaneamente static e final

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

D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

8. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
```

```
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

10. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

11. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

12. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java
```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java
```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

15. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

16. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A

B. La classe B

C. La classe C

D. Tutte le classi

E. Nessuna delle classi

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 47

1. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

5. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

6. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

7. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

8. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
```

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

9. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

10. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

11. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

12. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

13. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

14. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

15. Qual è l'output di questo codice?

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

```
        }
        catch( Exception s ) {
        }
    }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

16. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 48

1. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

2. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

3. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
```

```
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

5. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

6. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

7. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

9. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

10. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

11. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

14. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

15. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

16. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 49

1. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

2. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

   A. `s = (Error []) r;`
   B. `r = (Object [] []) b;`
   C. `r = b;`
   D. `r = (Object [] []) s;`
   E. `r = s;`

---

3. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

   A. La classe A
   B. La classe B
   C. La classe C
   D. Tutte le classi
   E. Nessuna delle classi

---

4. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

   A. `u = (Integer) c;`
   B. `d = (String) c;`
   C. `d = (String) u;`
   D. `u = (Integer) d;`
   E. Nessuno dei precedenti

---

5. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`

C. `protected String f(int k)`

D. `String f(int i) throws Exception`

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

A. 14Exception in thread main MyExc3

B. Errore a tempo di compilazione

C. 1Exception in thread main MyExc3

D. 1... (ciclo infinito)

E. Nessuna delle precedenti

---

7. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione

B. Errore a tempo di compilazione

C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

9. Qual è l'output di questo codice?

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

```
         q();
         System.out.print(2);
      }
      catch( Exception h ) {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc3 j ) {
      }
      catch( MyExc2 x ) {
      }
      finally {
         System.out.print(4);
         throw( new MyExc1() );
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc2() );
      }
      catch( Exception s ) {
      }
   }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

11. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

12. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

14. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001

B. 101

C. 010

D. 000

E. 111

---

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1

B. o2

C. o3

D. o1 e o2

E. Nessuna delle precedenti.

---

16. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 50

1. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

   A. Nessun errore a tempo di compilazione e di esecuzione
   B. Errore a tempo di compilazione
   C. Errore a tempo di esecuzione alla linea 5 in B.java
   D. Errore a tempo di esecuzione alla linea 6 in B.java
   E. Nessuna delle precedenti

---

3. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`
   C. `protected String f(int k)`
   D. `String f(int i) throws Exception`
   E. Nessuna delle precedenti.

---

5. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

6. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

8. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

9. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

10. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

11. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

12. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
```

```
        System.out.print(
                super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
            new B().m();
    }
}
```

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

13. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

14. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
```

```
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args){
        new B().f();
    }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

16. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 51

1. Qual è l'output di questo codice?

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

   A. Errore a tempo di compilazione
   B. 1524Exception in thread main MyExc1
   C. 15
   D. 15243333333... (ciclo infinito)
   E. Nessuna delle precedenti

---

2. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

   A. `u = (Integer) c;`
   B. `d = (String) c;`
   C. `d = (String) u;`
   D. `u = (Integer) d;`
   E. Nessuno dei precedenti

---

3. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

   A. o1
   B. o2
   C. o3
   D. o1 e o2

E. Nessuna delle precedenti.

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

5. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

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

```
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

8. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
```

```
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

12. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

13. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

14. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

15. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java
```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java
```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

16. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 52

1. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

   A. La classe A
   B. La classe B
   C. La classe C
   D. Tutte le classi
   E. Nessuna delle classi

---

2. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

3. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
```

```
        System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

4. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
```

```
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1

B. o2

C. o3

D. o1 e o2

E. Nessuna delle precedenti.

---

8. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`

B. `public int f( int j)`

C. `private void f( int j)`

D. `public static void f( int j)`

E. Nessuna delle precedenti

---

9. Dire quale delle seguenti affermazioni è falsa:

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

B. Un attributo può essere contemporaneamente static e final

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

D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

11. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`

B. `d = (String) c;`

C. `d = (String) u;`

D. `u = (Integer) d;`

E. Nessuno dei precedenti

---

12. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}
```

```
abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

14. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

15. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
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 B1 {
   public static void main(String [] argv)
                     throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 53

1. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

   A. Nessun errore a tempo di compilazione e di esecuzione
   B. Errore a tempo di compilazione
   C. Errore a tempo di esecuzione alla linea 5 in B.java
   D. Errore a tempo di esecuzione alla linea 6 in B.java
   E. Nessuna delle precedenti

---

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

   A. 341
   B. 3... (ciclo infinito)
   C. 31
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

3. Qual è l'output di questo codice?

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

```
      catch( MyExc2 x ) {
      }
      finally {
         System.out.print(4);
         throw( new MyExc1() );
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc2() );
      }
      catch( Exception s ) {
      }
   }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

5. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

6. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010

D. 000

E. 111

---

7. Dire quale delle seguenti affermazioni è falsa:

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

   B. Un attributo può essere contemporaneamente static e final

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

   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

8. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

   A. `s = (Error []) r;`

   B. `r = (Object [] []) b;`

   C. `r = b;`

   D. `r = (Object [] []) s;`

   E. `r = s;`

---

9. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`

   B. `public String f(byte k)`

   C. `protected String f(int k)`

   D. `String f(int i) throws Exception`

   E. Nessuna delle precedenti.

---

10. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

   A. o1

   B. o2

   C. o3

   D. o1 e o2

   E. Nessuna delle precedenti.

---

11. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

12. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

14. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

15. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
```

```
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

16. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 54

1. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

2. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

3. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

5. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

6. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

7. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
```

```
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

10. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

11. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

12. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`

B. `d = (String) c;`

C. `d = (String) u;`

D. `u = (Integer) d;`

E. Nessuno dei precedenti

---

13. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione

B. 1524Exception in thread main MyExc1

C. 15

D. 15243333333... (ciclo infinito)

E. Nessuna delle precedenti

---

14. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

15. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A

B. La classe B

C. La classe C

D. Tutte le classi

E. Nessuna delle classi

16. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione

B. Errore a tempo di compilazione

C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 55

1. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

2. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

4. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

5. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

7. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

8. Qual è l'output di questo codice?

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

```
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

9. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

10. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

11. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

12. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
    }
    public void g() {
```

```
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
            new B().m();
    }
}
```

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

13. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

14. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

15. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

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

```
   }
   catch( Exception h ) {
      System.out.print(3);
      throw( new MyExc1() );
   }
   catch( MyExc3 j ) {
   }
   catch( MyExc2 x ) {
   }
   finally {
      System.out.print(4);
      throw( new MyExc1() );
   }
}
static void q() throws Exception {
   try {
      System.out.print(5);
      throw( new MyExc2() );
   }
   catch( Exception s ) {
   }
  }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 56

1. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Qual è l'output di questo codice?

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

   A. 14Exception in thread main MyExc3
   B. Errore a tempo di compilazione
   C. 1Exception in thread main MyExc3
   D. 1... (ciclo infinito)
   E. Nessuna delle precedenti

---

3. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`
   C. `protected String f(int k)`
   D. `String f(int i) throws Exception`

E. Nessuna delle precedenti.

---

5. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

6. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
```

```
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
```

```
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

12. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

14. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

16. Qual è l'output di questo codice?

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

```
   catch( Exception h ) {
      System.out.print(3);
      throw( new MyExc1() );
   }
   catch( MyExc3 j ) {
   }
   catch( MyExc2 x ) {
   }
   finally {
      System.out.print(4);
      throw( new MyExc1() );
   }
}
static void q() throws Exception {
   try {
      System.out.print(5);
         throw( new MyExc2() );
      }
      catch( Exception s ) {
      }
   }
}
```

A. Errore a tempo di compilazione

B. 1524Exception in thread main MyExc1

C. 15

D. 15243333333... (ciclo infinito)

E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 57

1. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

3. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

   A. Nessun errore a tempo di compilazione e di esecuzione
   B. Errore a tempo di compilazione
   C. Errore a tempo di esecuzione alla linea 5 in B.java
   D. Errore a tempo di esecuzione alla linea 6 in B.java
   E. Nessuna delle precedenti

---

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
    }
```

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

5. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

7. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
```

```
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
   }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

8. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

9. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

11. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

12. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

13. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

14. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

16. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 58

1. Qual è l'output di questo codice?

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

   A. Errore a tempo di compilazione
   B. 1524Exception in thread main MyExc1
   C. 15
   D. 15243333333... (ciclo infinito)
   E. Nessuna delle precedenti

---

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

   A. o1
   B. o2
   C. o3
   D. o1 e o2
   E. Nessuna delle precedenti.

---

3. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
```

```
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

4. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

5. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

8. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

9. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

10. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

12. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

16. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 59

1. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

   A. `private void f( float i)`
   B. `public int f( int j)`
   C. `private void f( int j)`
   D. `public static void f( int j)`
   E. Nessuna delle precedenti

---

3. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

4. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

   A. `s = (Error []) r;`
   B. `r = (Object [] []) b;`
   C. `r = b;`
   D. `r = (Object [] []) s;`
   E. `r = s;`

---

5. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
```

```
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

6. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

7. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

8. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

9. Qual è l'output di questo codice?

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

```
         throw( new MyExc1() );
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc2() );
      }
      catch( Exception s ) {
      }
   }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

10. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

11. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

12. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
```

```
            new B().m();
    }
}
```

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

15. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 60

1. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

3. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

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

```
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

4. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

5. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

6. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

8. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1

B. o2

C. o3

D. o1 e o2

E. Nessuna delle precedenti.

---

9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

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

A. 14Exception in thread main MyExc3

B. Errore a tempo di compilazione

C. 1Exception in thread main MyExc3

D. 1... (ciclo infinito)

E. Nessuna delle precedenti

---

11. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

12. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

13. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

14. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

15. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
```

```
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

16. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 61

1. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

3. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

4. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

5. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
```

```
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

11. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1

B. o2

C. o3

D. o1 e o2

E. Nessuna delle precedenti.

---

13. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001

B. 101

C. 010

D. 000

E. 111

---

14. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`

B. `r = (Object [] []) b;`

C. `r = b;`

D. `r = (Object [] []) s;`

E. `r = s;`

---

15. Dire quale delle seguenti affermazioni è falsa:

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

B. Un attributo può essere contemporaneamente static e final

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

D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

16. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 62

1. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

3. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
```

```
5.      int a = new A().getI();
6.      int b = new B().getI();
7.    }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione

B. Errore a tempo di compilazione

C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

5. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`

B. `public String f(byte k)`

C. `protected String f(int k)`

D. `String f(int i) throws Exception`

E. Nessuna delle precedenti.

---

6. Dire quale delle seguenti affermazioni è falsa:

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

B. Un attributo può essere contemporaneamente static e final

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

D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

7. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

9. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

10. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

11. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}
```

```
class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

12. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

14. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

15. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
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 B1 {
   public static void main(String [] argv)
                     throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 63

1. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

2. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

3. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

4. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
```

```
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

5. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

7. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

8. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`

B. `d = (String) c;`

C. `d = (String) u;`

D. `u = (Integer) d;`

E. Nessuno dei precedenti

---

9. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001

B. 101

C. 010

D. 000

E. 111

---

10. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A

B. La classe B

C. La classe C

D. Tutte le classi

E. Nessuna delle classi

---

11. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione

B. Errore a tempo di compilazione

C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341

B. 3... (ciclo infinito)

C. 31

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

13. Dire quale delle seguenti affermazioni è falsa:

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

B. Un attributo può essere contemporaneamente static e final

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

D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

14. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
```

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 64

1. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

2. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
```

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

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

5. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

6. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

11. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java
```
1. package pkgA;
2. public class A {
3.   private int i = 1;
4.   int getI() { return i; }
5. }
```

B.java
```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.   public static void main(String[] args){
5.     int a = new A().getI();
6.     int b = new B().getI();
7.   }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

12. Qual è l'output di questo codice?

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

```
         q();
         System.out.print(2);
      }
      catch( Exception h ) {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc3 j ) {
      }
      catch( MyExc2 x ) {
      }
      finally {
         System.out.print(4);
         throw( new MyExc1() );
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc2() );
      }
      catch( Exception s ) {
      }
   }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
         }
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

14. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

15. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
```

```
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

16. Dire quale delle seguenti affermazioni è falsa:

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

B. Un attributo può essere contemporaneamente static e final

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

D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 65

1. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

2. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

3. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

4. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

5. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

6. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

7. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

8. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`

D. `u = (Integer) d;`

E. Nessuno dei precedenti

---

10. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001

B. 101

C. 010

D. 000

E. 111

---

11. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1

B. o2

C. o3

D. o1 e o2

E. Nessuna delle precedenti.

---

12. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A

B. La classe B

C. La classe C

D. Tutte le classi

E. Nessuna delle classi

---

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

15. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

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

```
   }
   catch( Exception h ) {
      System.out.print(3);
      throw( new MyExc1() );
   }
   catch( MyExc3 j ) {
   }
   catch( MyExc2 x ) {
   }
   finally {
      System.out.print(4);
      throw( new MyExc1() );
   }
}
static void q() throws Exception {
   try {
      System.out.print(5);
      throw( new MyExc2() );
   }
   catch( Exception s ) {
   }
  }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 66

1. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

   A. La classe A
   B. La classe B
   C. La classe C
   D. Tutte le classi
   E. Nessuna delle classi

---

2. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

   A. `s = (Error []) r;`
   B. `r = (Object [] []) b;`
   C. `r = b;`
   D. `r = (Object [] []) s;`
   E. `r = s;`

---

3. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

4. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

5. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

6. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

8. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
```

```
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

11. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

12. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

14. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

15. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

16. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
```

```
void q(String s2, Boolean b1, Boolean b2){
    String s1;
    s1 = new String("abcde");
    if(s2 == s1) {
       System.out.print(1);
    } else {
       System.out.print(0);
    }
    if(b1 == b2) {
       System.out.print(1);
    } else {
       System.out.print(0);
    }
    if(s4 == s5) {
       System.out.print(1);
    } else {
       System.out.print(0);
    }
  }
}
```

A. 001

B. 101

C. 010

D. 000

E. 111

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 67

1. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

2. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

3. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

4. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
```

```
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

5. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

6. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

7. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
                super.s + s);
        g();
    }
```

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

9. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

10. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

11. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A

B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

13. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

14. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 68

1. Dire quale delle seguenti affermazioni è falsa:

   A. Ai metodi static non si applica il *dynamic method dispatch*
   B. Un attributo può essere contemporaneamente static e final
   C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
   D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
   E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

2. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

   A. `s = (Error []) r;`
   B. `r = (Object [] []) b;`
   C. `r = b;`
   D. `r = (Object [] []) s;`
   E. `r = s;`

---

3. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

   indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

   A. `public void f(int j)`
   B. `public String f(byte k)`
   C. `protected String f(int k)`
   D. `String f(int i) throws Exception`
   E. Nessuna delle precedenti.

---

4. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

   A. A
   B. ACD
   C. ABD
   D. Errore a tempo di compilazione
   E. Nessuna delle precedenti

---

5. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
```

```
5.      int a = new A().getI();
6.      int b = new B().getI();
7.    }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

9. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

10. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

11. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

12. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
```

```
            System.out.print(1);
         } else {
            System.out.print(0);
         }
      }
   }
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

13. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
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 B1 {
   public static void main(String [] argv)
                       throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 69

1. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

   A. o1
   B. o2
   C. o3
   D. o1 e o2
   E. Nessuna delle precedenti.

---

2. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }
```

```
B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

   A. Nessun errore a tempo di compilazione e di esecuzione
   B. Errore a tempo di compilazione
   C. Errore a tempo di esecuzione alla linea 5 in B.java
   D. Errore a tempo di esecuzione alla linea 6 in B.java
   E. Nessuna delle precedenti

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
```

```
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

5. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

6. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

7. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

10. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

11. Qual è l'output di questo codice?

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

```
      finally {
         System.out.print(4);
         throw( new MyExc1() );
      }
   }
   static void q() throws Exception {
      try {
         System.out.print(5);
         throw( new MyExc2() );
      }
      catch( Exception s ) {
      }
   }
}
```

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

12. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
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 B1 {
   public static void main(String [] argv)
                        throws Exception {
      try {
         System.out.print(1);
         m();
         System.out.print(2);
      }
      catch( MyExc2 g ) {
         System.out.print(3);
      }
   }
   static void m() throws Exception {
      try {
         throw( new MyExc3() );
      }
      catch( MyExc3 c ) {
         throw( new MyExc3() );
      }
      finally {
         System.out.print(4);
      }
   }
}
```

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

14. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

15. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`

B. `r = (Object [] []) b;`

C. `r = b;`

D. `r = (Object [] []) s;`

E. `r = s;`

---

16. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`

B. `d = (String) c;`

C. `d = (String) u;`

D. `u = (Integer) d;`

E. Nessuno dei precedenti

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

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

Prova n. 70

1. Qual è l'output di questo codice?

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

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

---

2. Quale output si ottiene invocando il metodo p?

```
class C {
  private Boolean b3 = new Boolean(true);
  private Boolean b4;
  private String s4 = new String("ab");
  private String s5;
  void p() {
      String s3;
      s3 = new String("abcde");
      b4 = b3;
      s5 = new String("ab");
      q(s3, b3, b4);
  }
  void q(String s2, Boolean b1, Boolean b2){
      String s1;
      s1 = new String("abcde");
      if(s2 == s1) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(b1 == b2) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
      if(s4 == s5) {
         System.out.print(1);
      } else {
         System.out.print(0);
      }
  }
}
```

A. 001
B. 101
C. 010
D. 000
E. 111

---

3. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
```

```
        System.out.print(
                super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
            new B().m();
    }
}
```

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

---

4. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

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

A. `u = (Integer) c;`
B. `d = (String) c;`
C. `d = (String) u;`
D. `u = (Integer) d;`
E. Nessuno dei precedenti

---

5. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. `public void f(int j)`
B. `public String f(byte k)`
C. `protected String f(int k)`
D. `String f(int i) throws Exception`
E. Nessuna delle precedenti.

---

6. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
  private String s;
}

class Y {
  private String s;
  private void setS(String s) {
      this.s = s;
  }
  private String getS() {
      return s;
  }
}

class Z {
  private String s;
  public void setS(String s) {
      this.s = s;
  }
  public String getS() {
      return s;
  }
}
```

A. La classe A
B. La classe B
C. La classe C
D. Tutte le classi
E. Nessuna delle classi

---

7. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

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

A. `s = (Error []) r;`
B. `r = (Object [] []) b;`
C. `r = b;`
D. `r = (Object [] []) s;`
E. `r = s;`

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
  public static void main(String[] argv){
      try {
         n();
      }
      catch( MyExc1 z ) {
         System.out.print(1);
      }
```

```
      catch( MyExc2 e ) {
         throw( new Error() );
      }
      catch( MyExc3 f ) {
         System.out.print(2);
      }
  }
  static void n() {
      try {
         System.out.print(3);
         throw( new MyExc1() );
      }
      catch( MyExc1 b ) {
         throw( new MyExc1() );
      }
      catch( MyExc3 i ) {
      }
      catch( MyExc2 v ) {
      }
      finally {
         System.out.print(4);
      }
  }
}
```

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

9. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. `private void f( float i)`
B. `public int f( int j)`
C. `private void f( int j)`
D. `public static void f( int j)`
E. Nessuna delle precedenti

---

10. Qual è l'output di questo programma?

```
abstract class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree() {
      super("C");
      System.out.print("D");
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

11. Qual è l'output di questo programma?

```
class Plant {
  Plant() {
      System.out.print("B");
  }
  Plant(String s) {
      System.out.print(s);
  }
}
class Tree extends Plant{
  Tree(String s) {
      super("C");
      System.out.print(s);
  }
}
class Palm extends Tree {
  public static void main(String argv[]){
      System.out.print("A");
      Palm p = new Palm();
  }
}
```

A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

---

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
  A ref;
  static Object o;
  A(Object o) {
      this.o = o;
  }
  void setRef(A a) {ref = a;}
```

```
}

class B {
  A a1 = new A("o1");
  A a2 = new A("o2");
  A a3 = new A("o3");
  void f() {
      a1.setRef(a2);
      a2.setRef(a3);
      a3.setRef(a1);
      a1 = a3; a2 = a3;
      m();
  }
  void m() { /* do something */ }
  public static void main (String[] args){
      new B().f();
  }
}
```

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.

---

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
 1. package pkgA;
 2. public class A {
 3.   private int i = 1;
 4.   int getI() { return i; }
 5. }

B.java
 1. package pkgB;
 2. import pkgA.A;
 3. public class B extends A {
 4.   public static void main(String[] args){
 5.     int a = new A().getI();
 6.     int b = new B().getI();
 7.   }
 8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

---

14. Dire quale delle seguenti affermazioni è falsa:

A. Ai metodi static non si applica il *dynamic method dispatch*
B. Un attributo può essere contemporaneamente static e final
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

15. Qual è l'output di questo codice?

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

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 15243333333... (ciclo infinito)
E. Nessuna delle precedenti

---

16. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

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

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

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 — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 2

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 3

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 4

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 5

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 6

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 7

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 8

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 9

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 10

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 11

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 12

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 13

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 14

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 15

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 16

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 17

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 18

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 19

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 20

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 21

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 22

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 23

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 24

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 25

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 26

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 27

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 28

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 29

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 30

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 31

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 32

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 33

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 34

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 35

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 36

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 37

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 38

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 39

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 40

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 41

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 42

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 43

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 44

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 45

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 46

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 47

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 48

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 49

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 50

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 51

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 52

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 53

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 54

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 55

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 56

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 57

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 58

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 59

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 60

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 61

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 62

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 63

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 64

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 65

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 66

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 67

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 68

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 69

Università di Napoli Federico II – Corso di Laurea in Informatica

LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

Studente e matricola:

Ora di inizio: Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

Prova n. 70

Università di Napoli Federico II – Corso di Laurea in Informatica

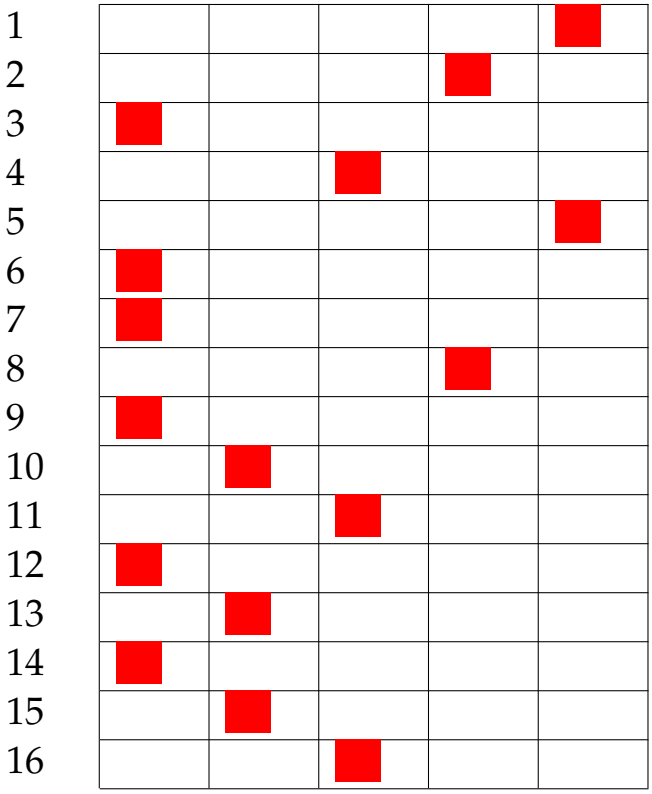
LP1 — Prova d'esame

prof. Piero A. Bonatti — 14 giugno 2016

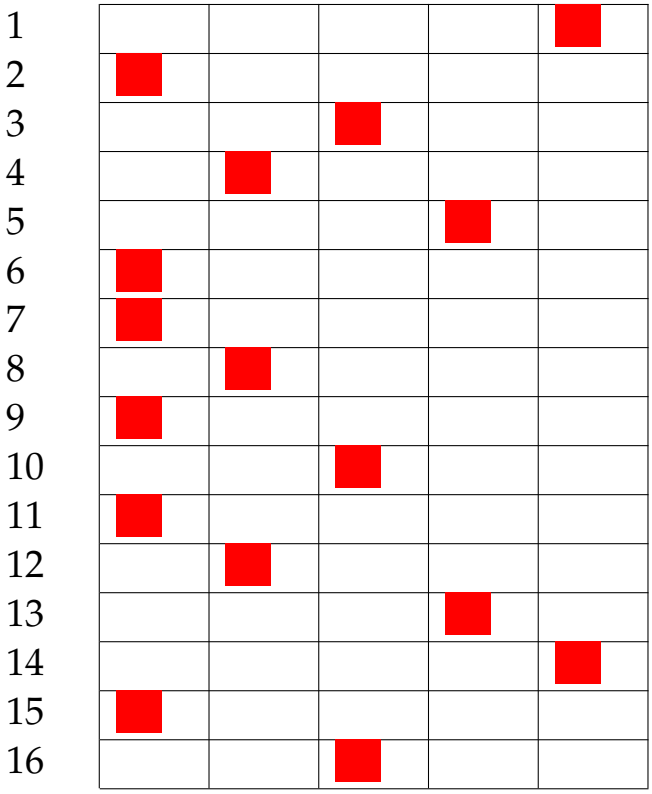
Studente e matricola:

Ora di inizio: Ora di consegna:

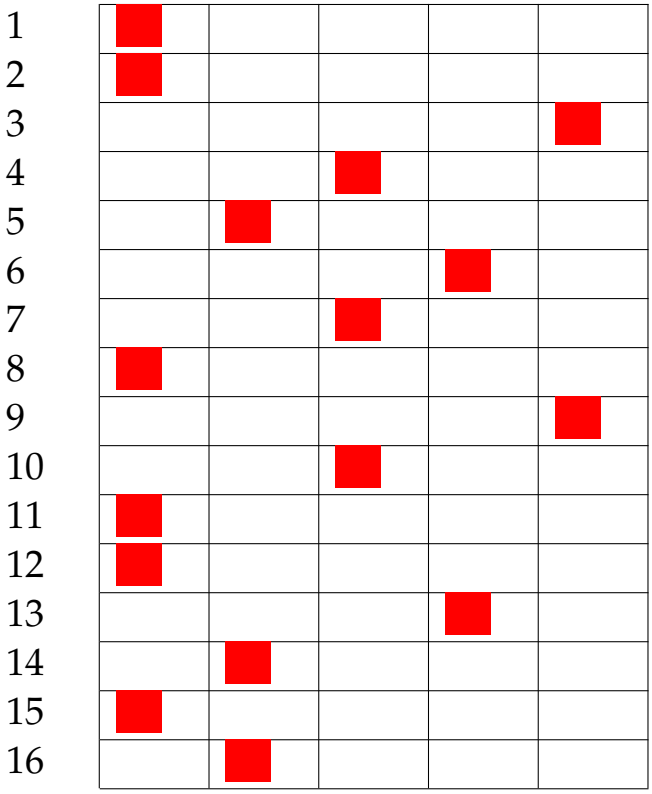
1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E



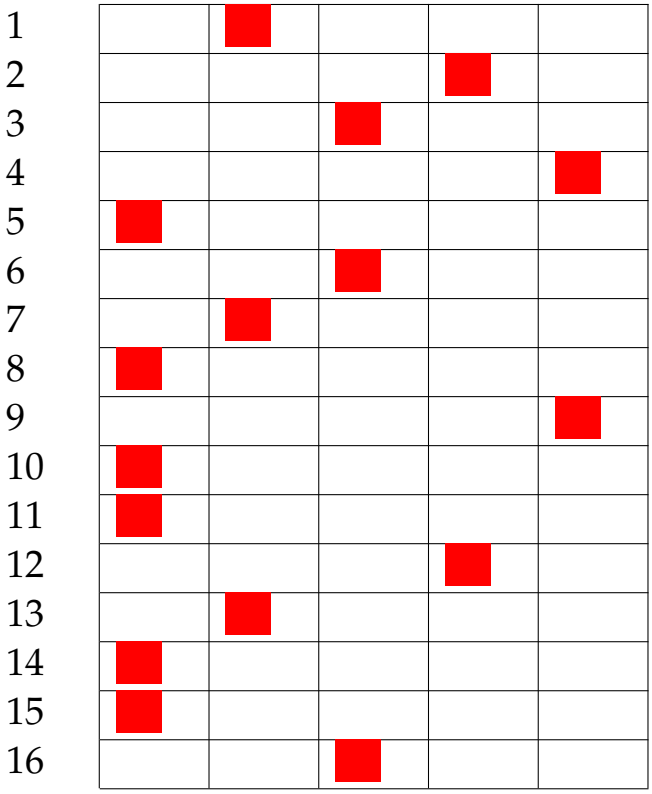
Risultato prova n. 1:



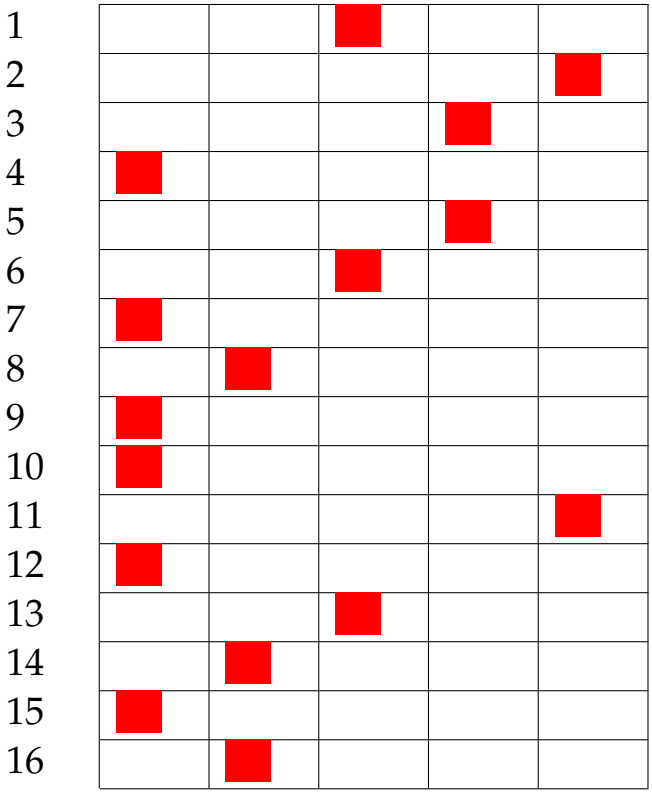
Risultato prova n. 2:



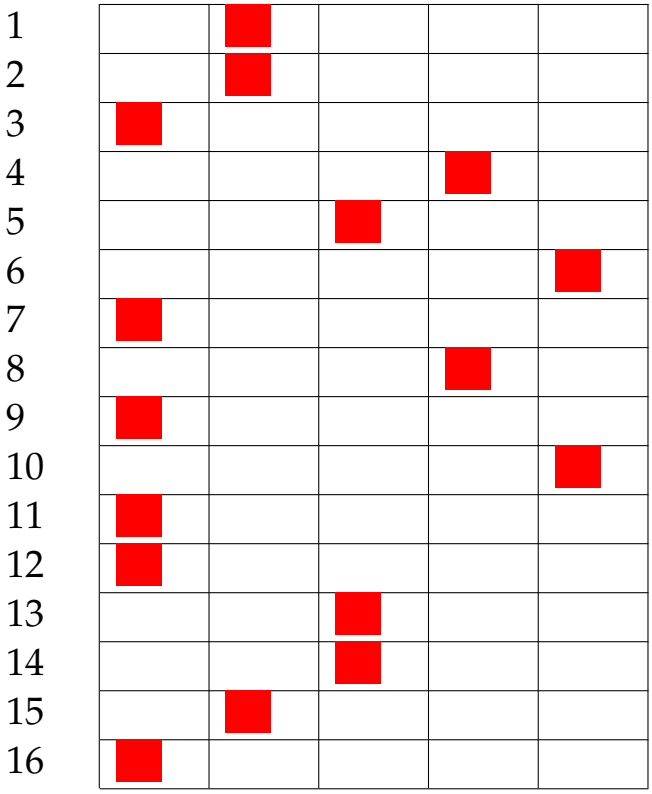
Risultato prova n. 3:



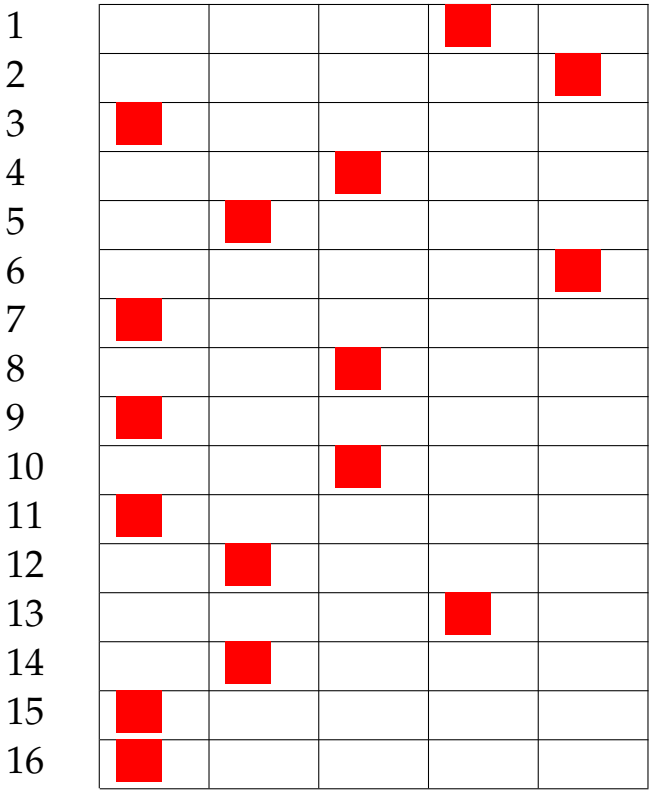
Risultato prova n. 4:



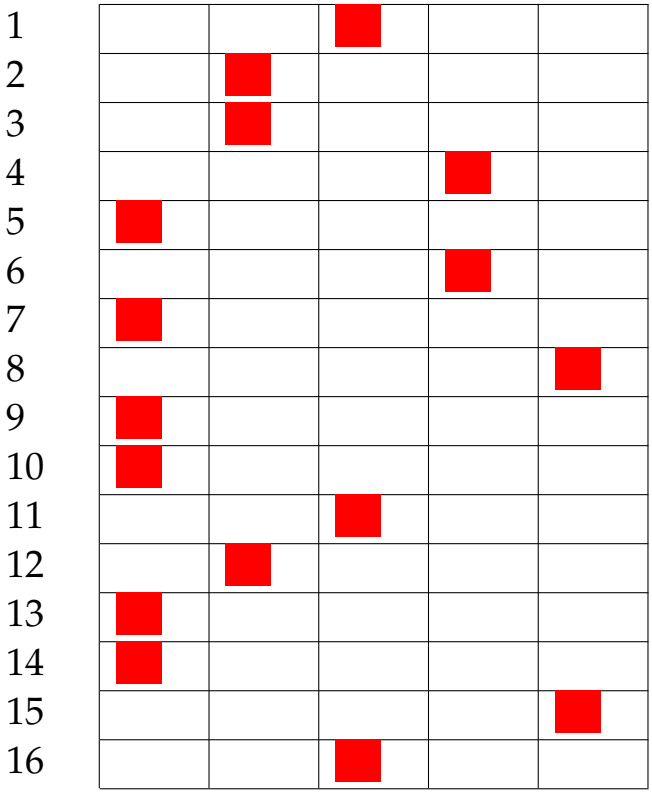
Risultato prova n. 5:



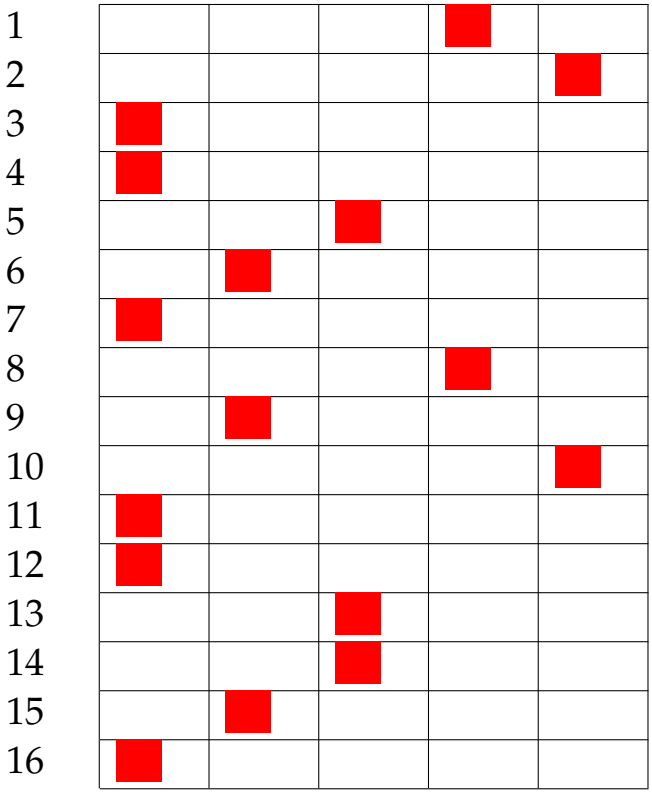
Risultato prova n. 6:



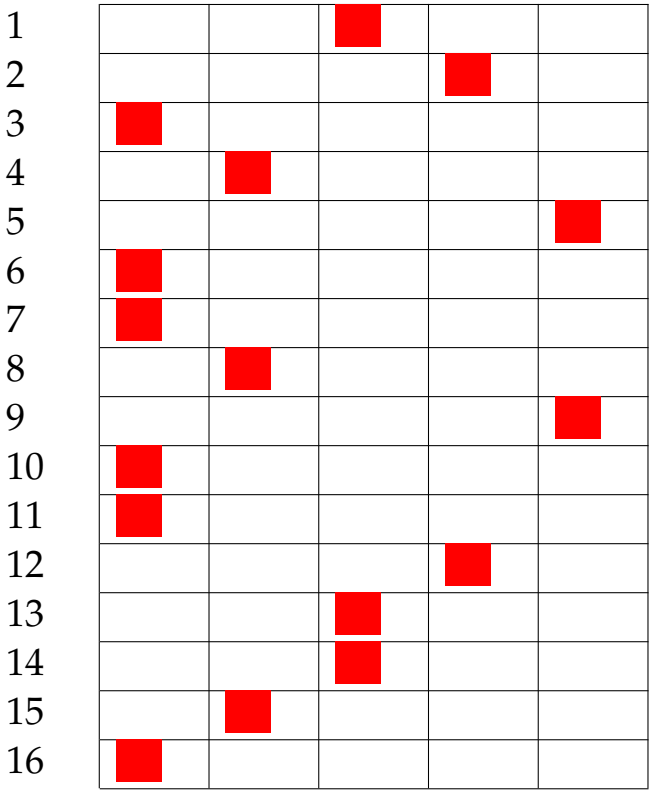
Risultato prova n. 7:



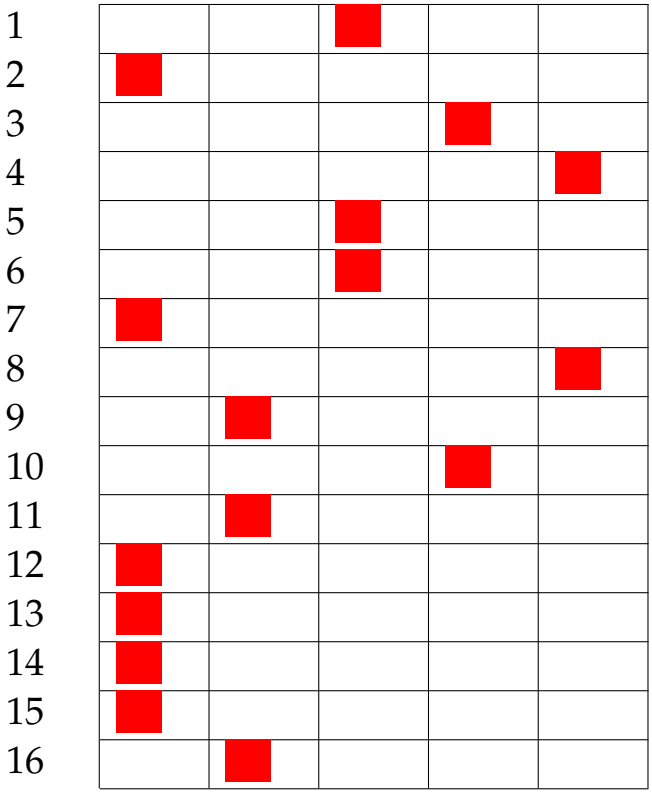
Risultato prova n. 8:



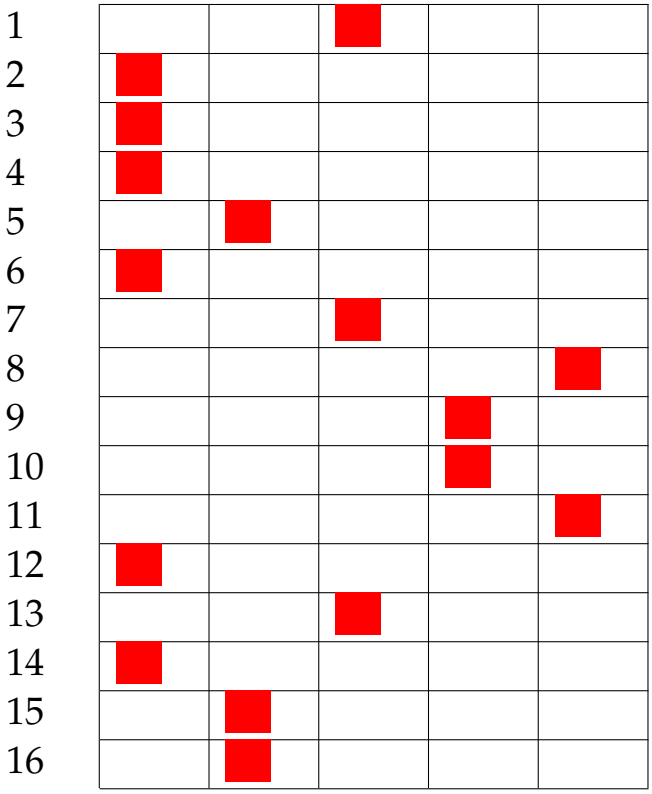
Risultato prova n. 9:



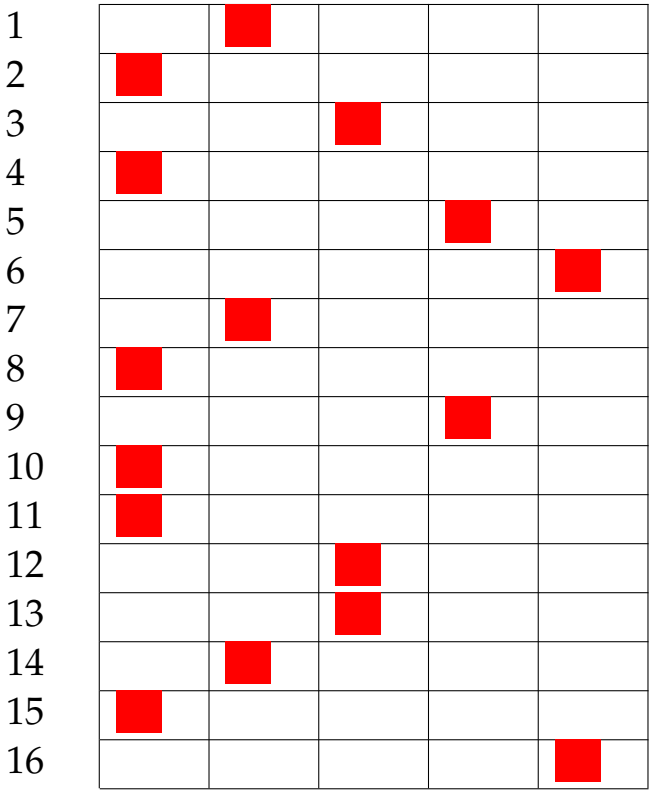
Risultato prova n. 10:



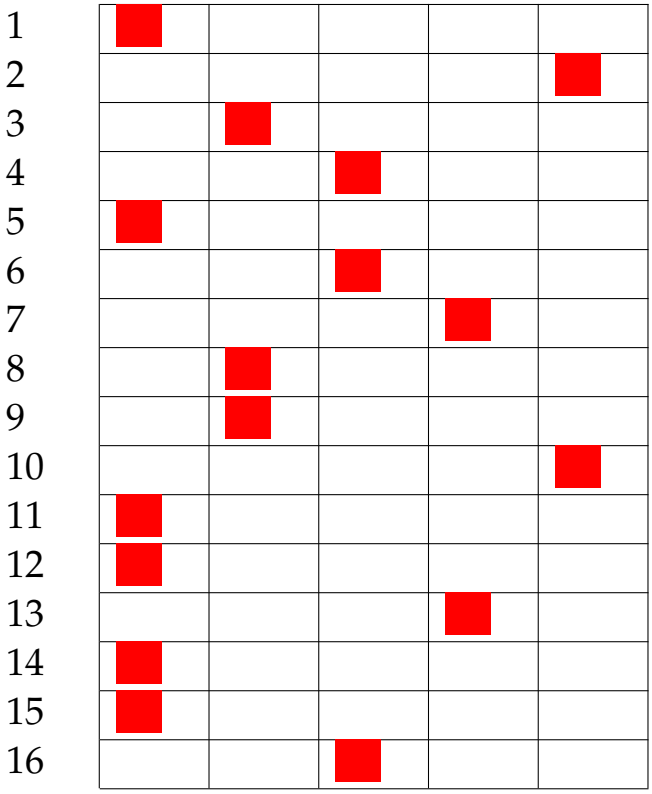
Risultato prova n. 11:



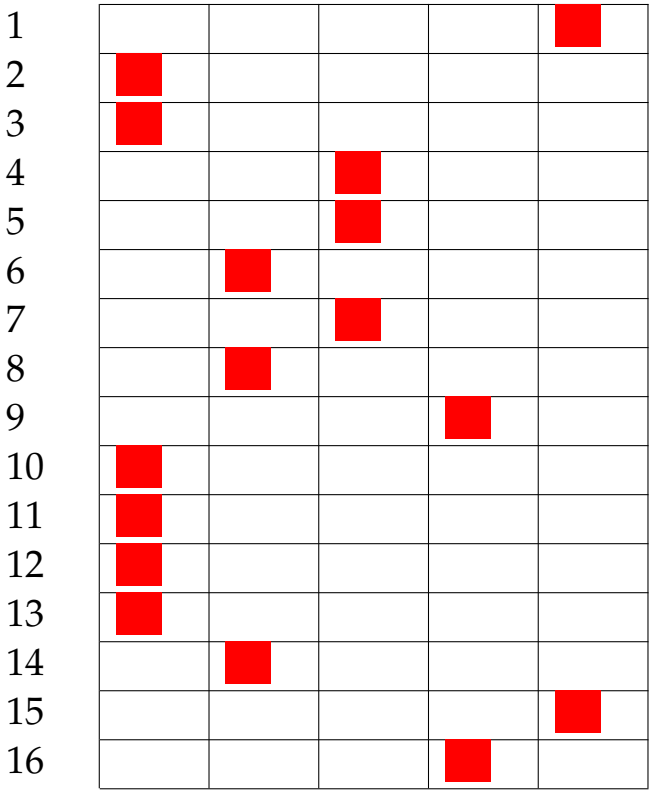
Risultato prova n. 12:



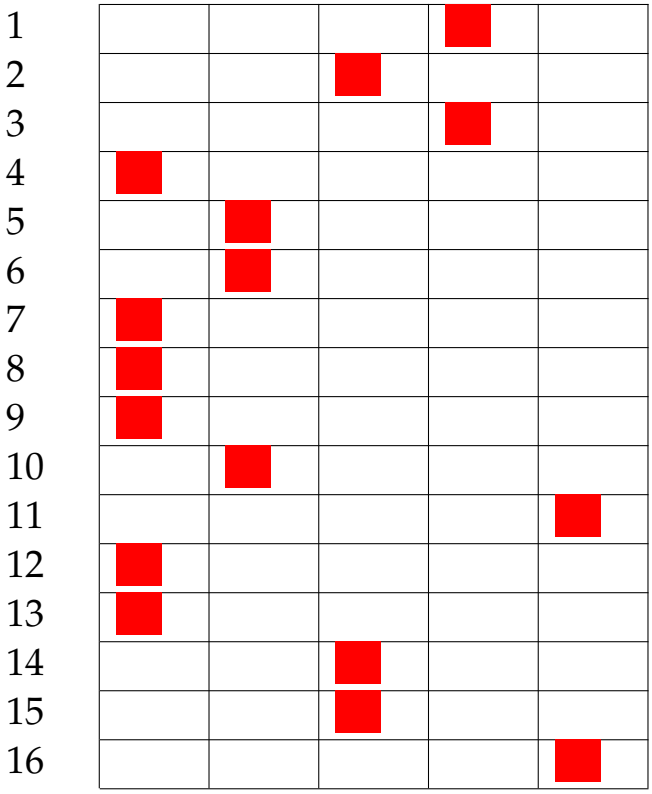
Risultato prova n. 13:



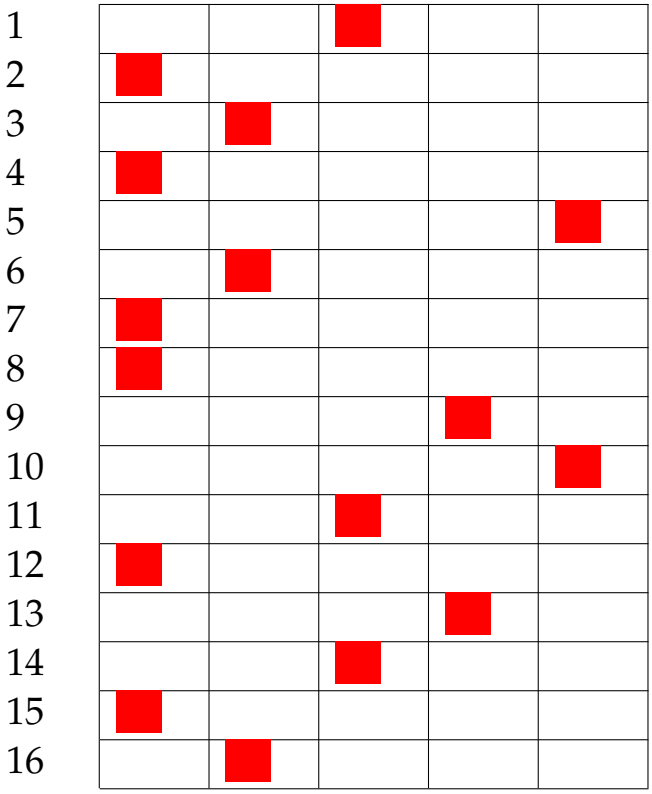
Risultato prova n. 14:



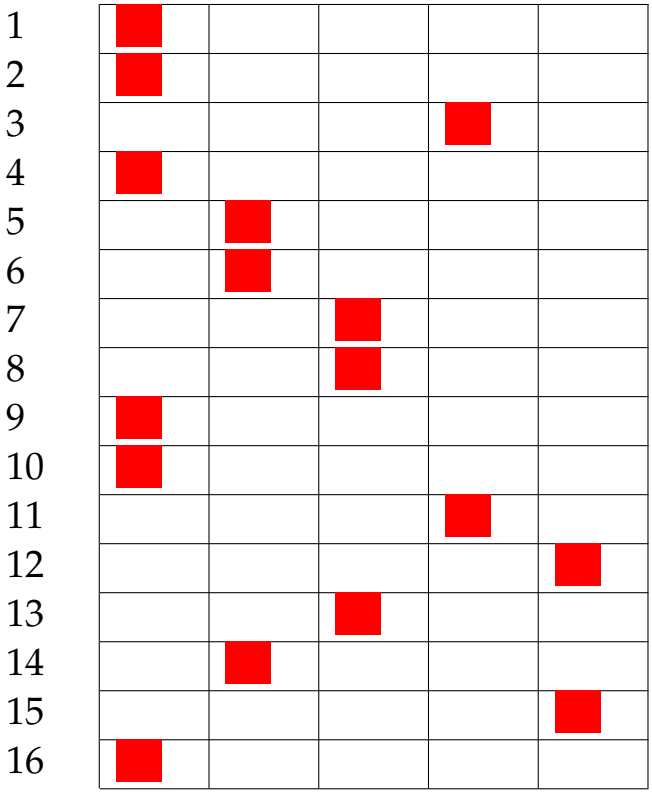
Risultato prova n. 15:



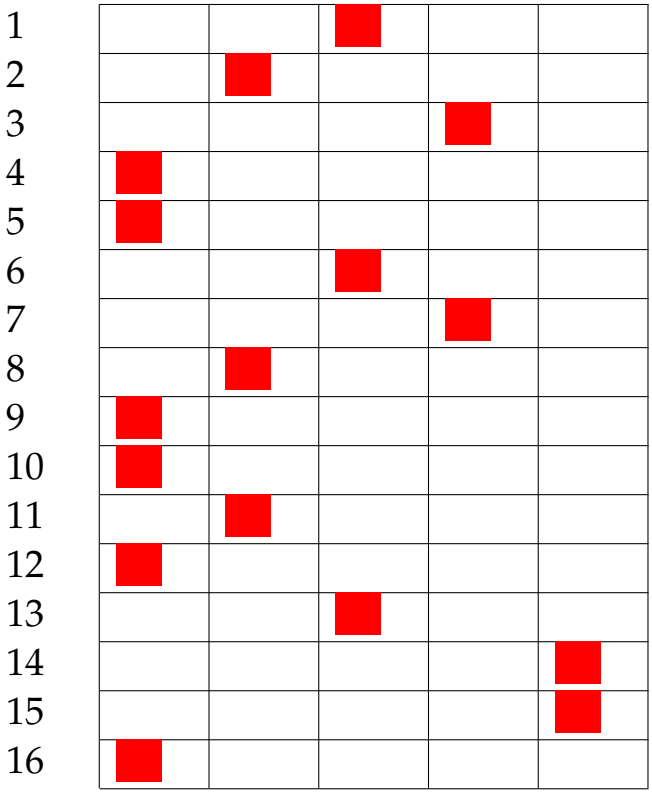
Risultato prova n. 16:



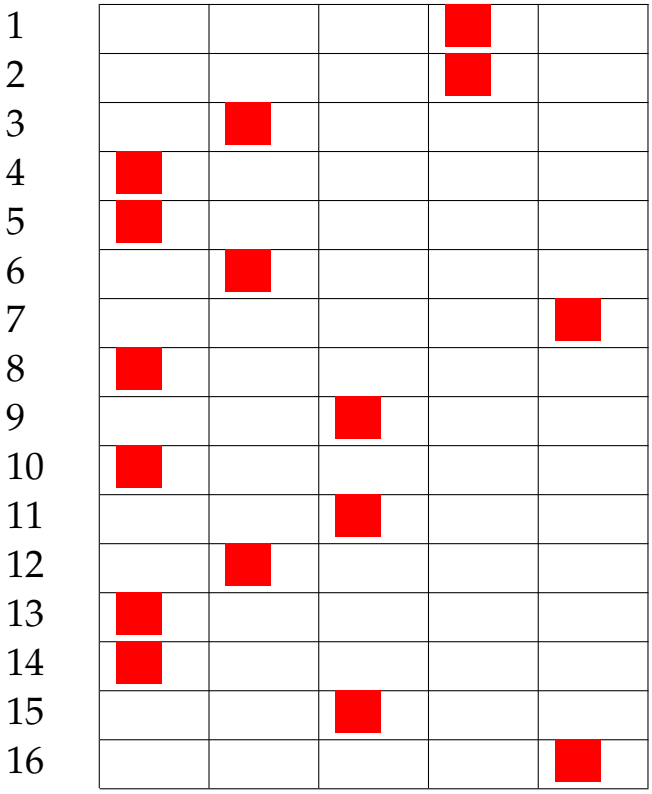
Risultato prova n. 17:



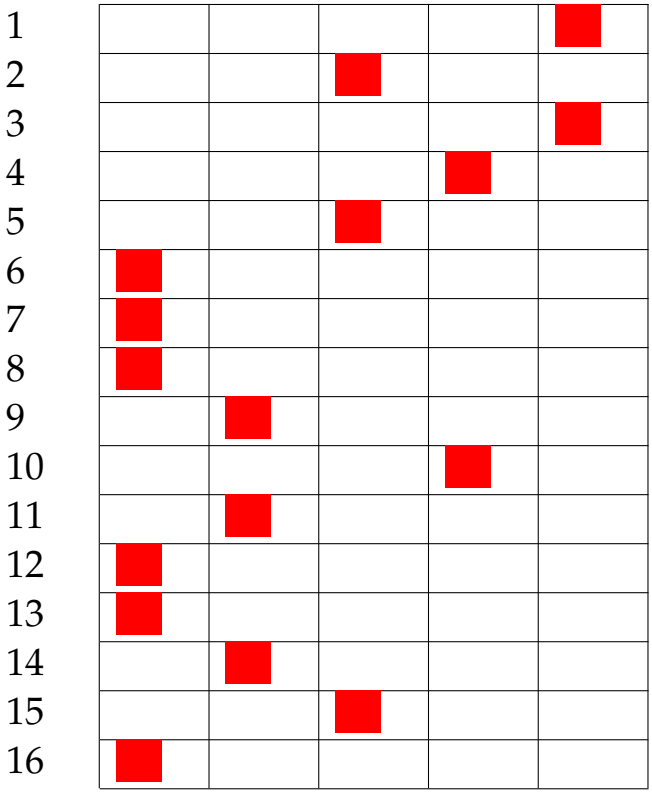
Risultato prova n. 18:



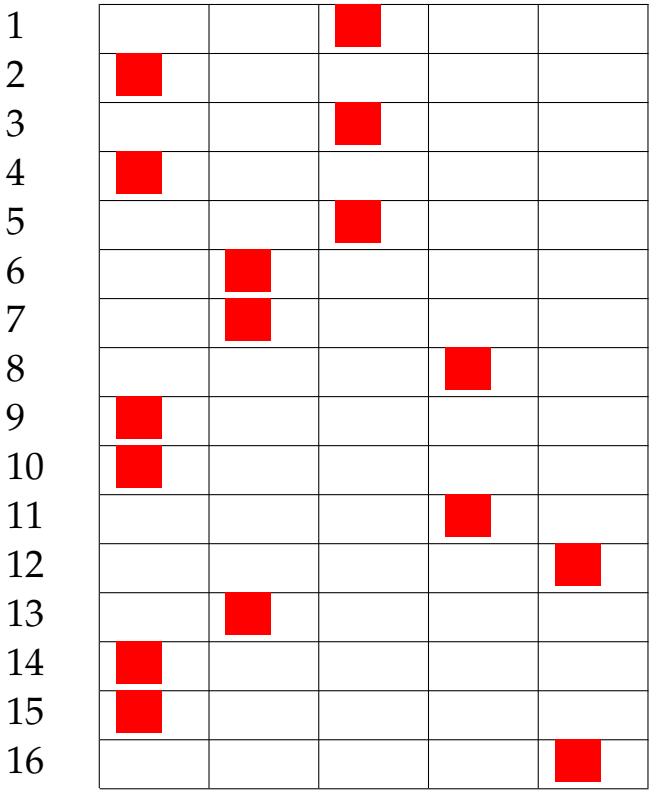
Risultato prova n. 19:



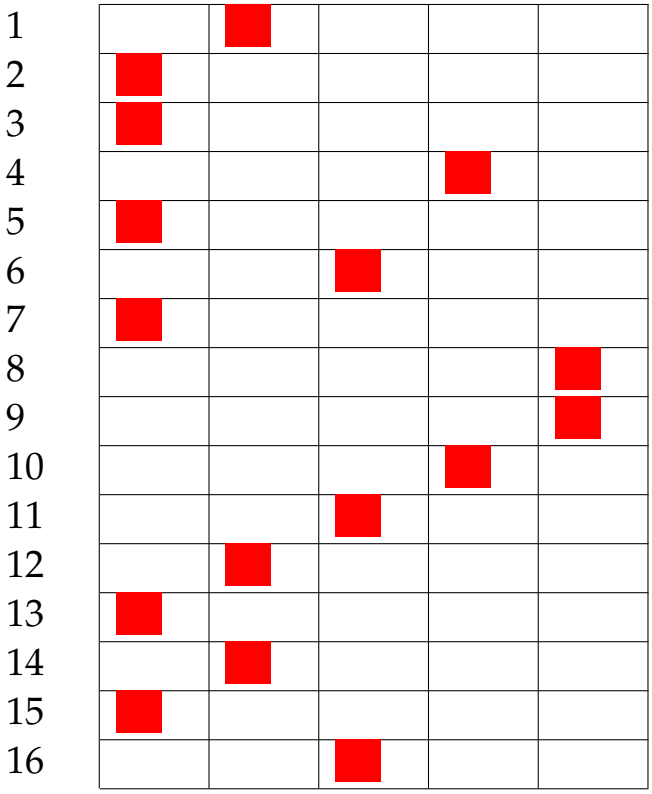
Risultato prova n. 20:



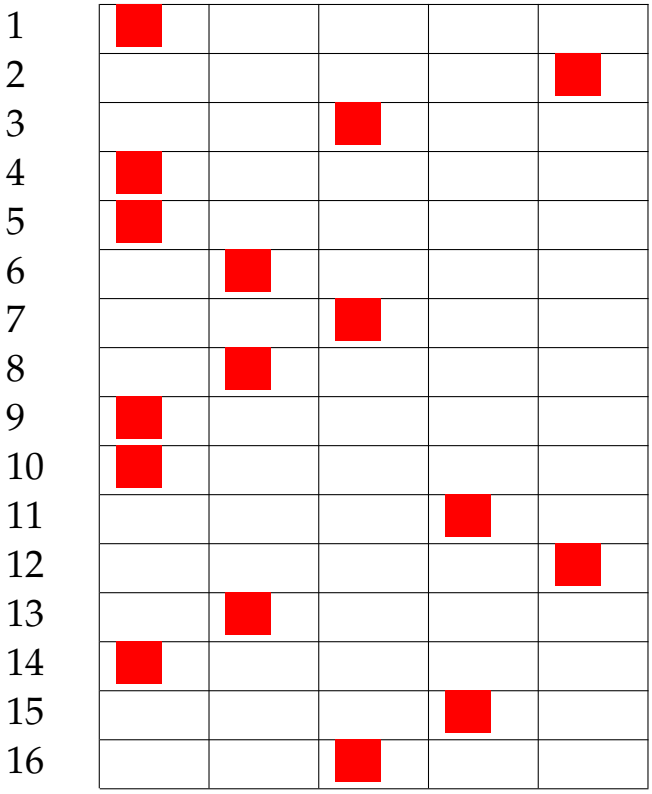
Risultato prova n. 21:



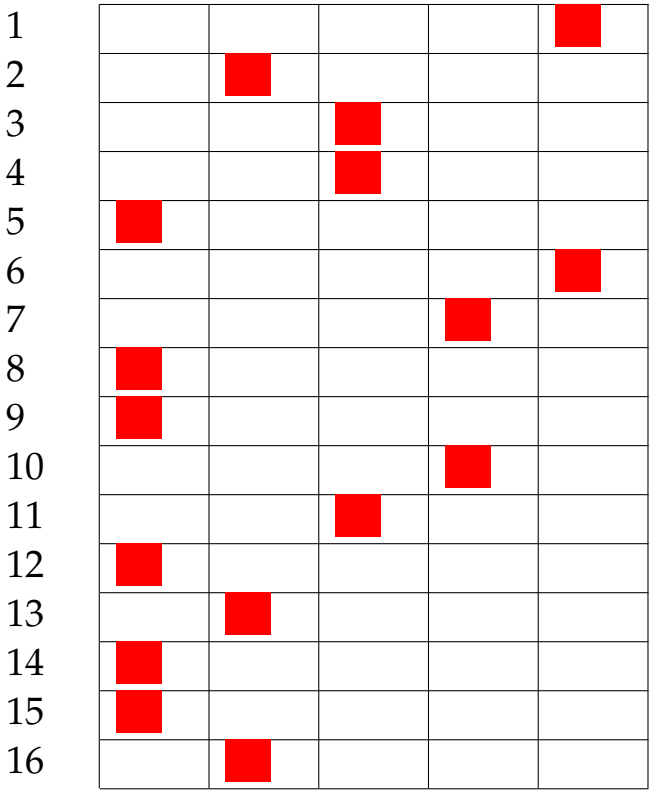
Risultato prova n. 22:



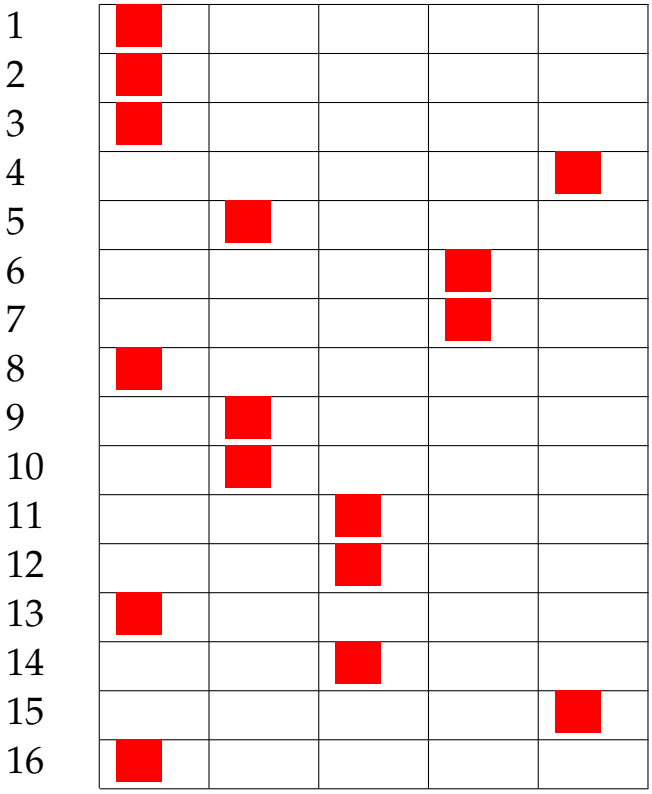
Risultato prova n. 23:



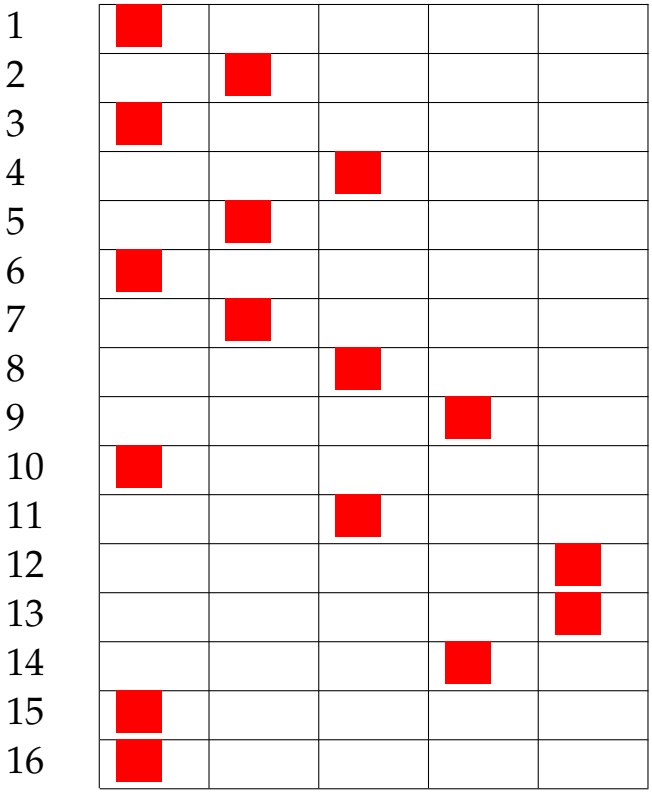
Risultato prova n. 24:



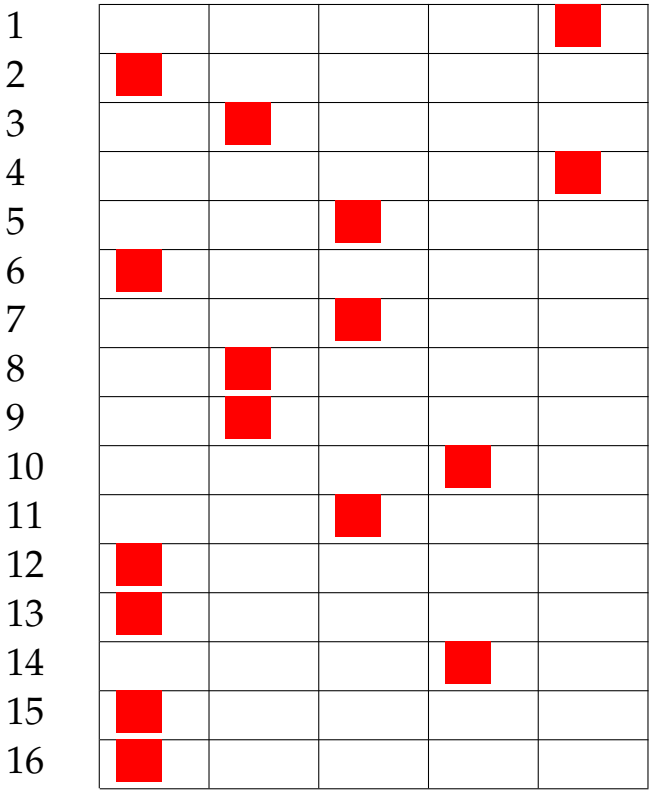
Risultato prova n. 25:



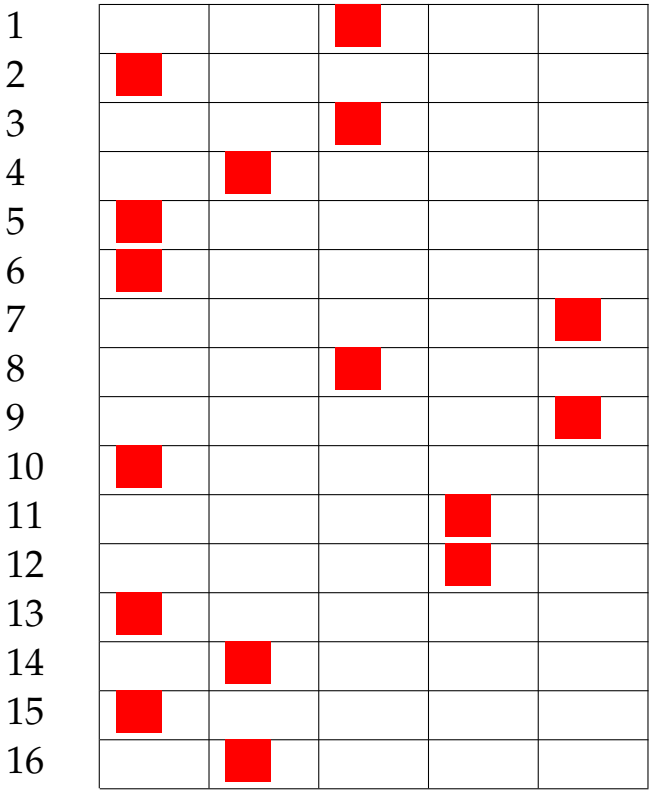
Risultato prova n. 26:



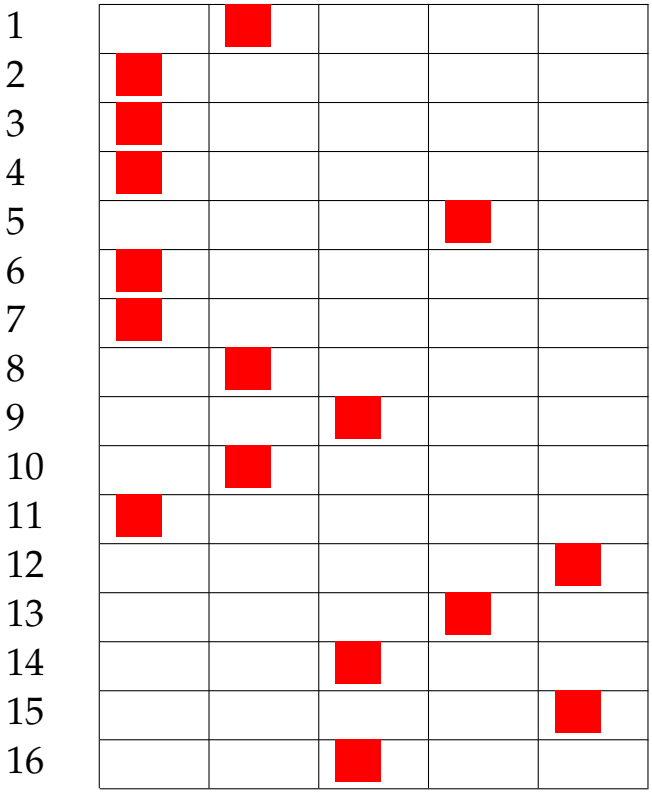
Risultato prova n. 27:



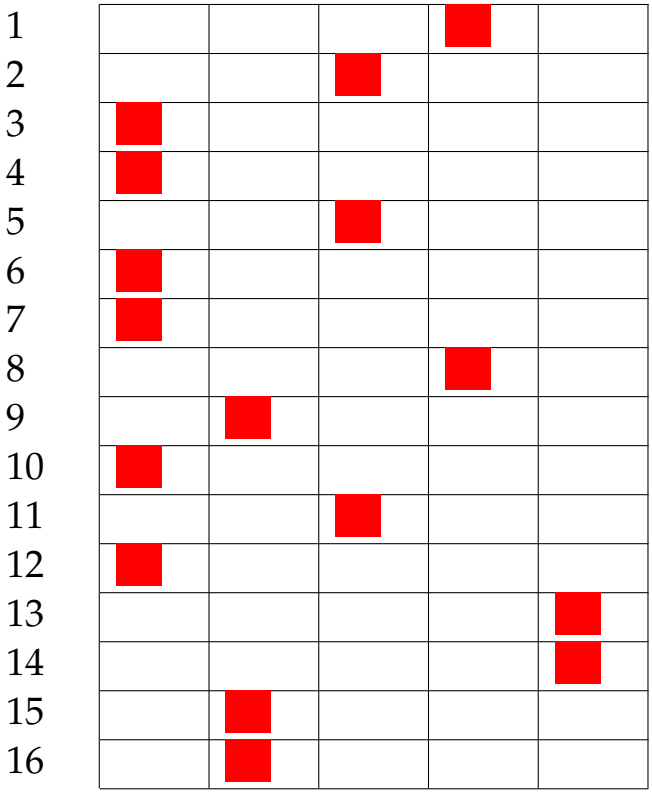
Risultato prova n. 28:



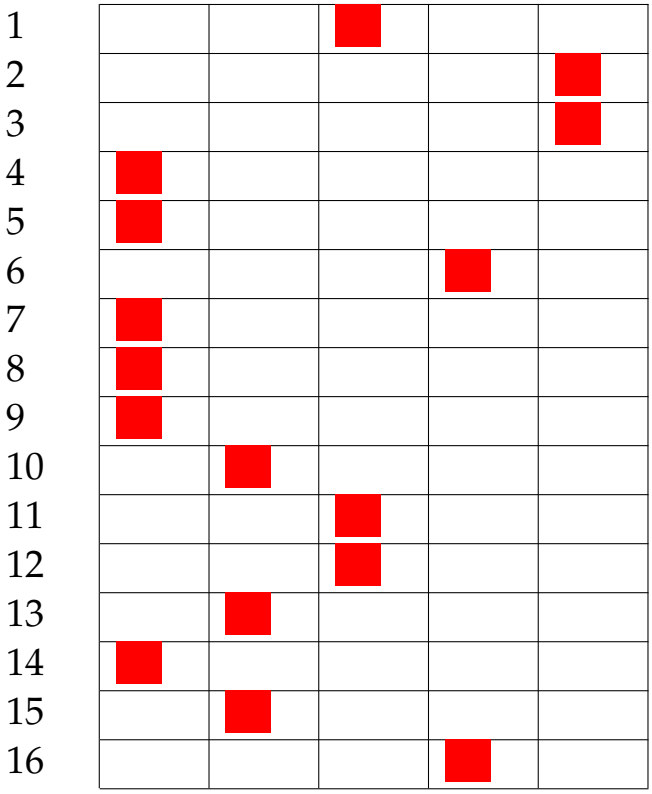
Risultato prova n. 29:



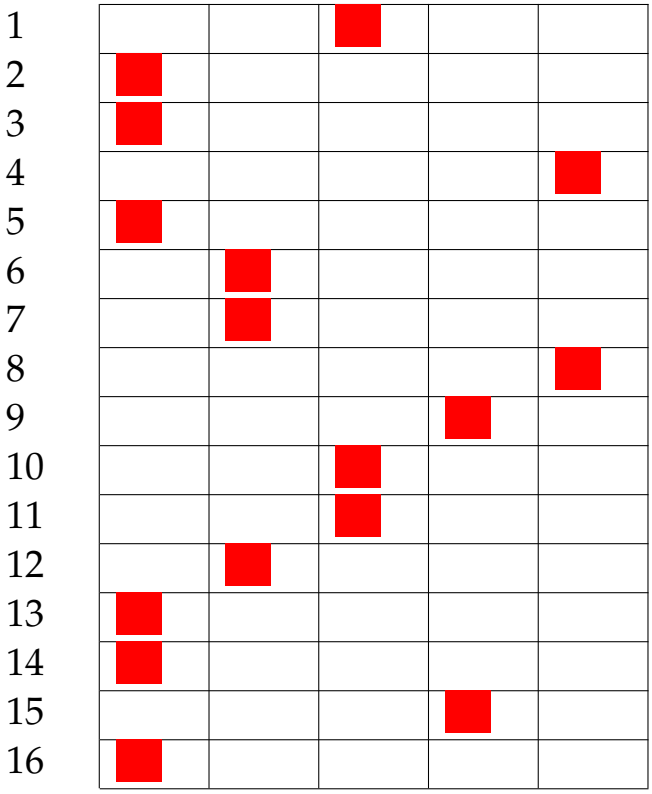
Risultato prova n. 30:



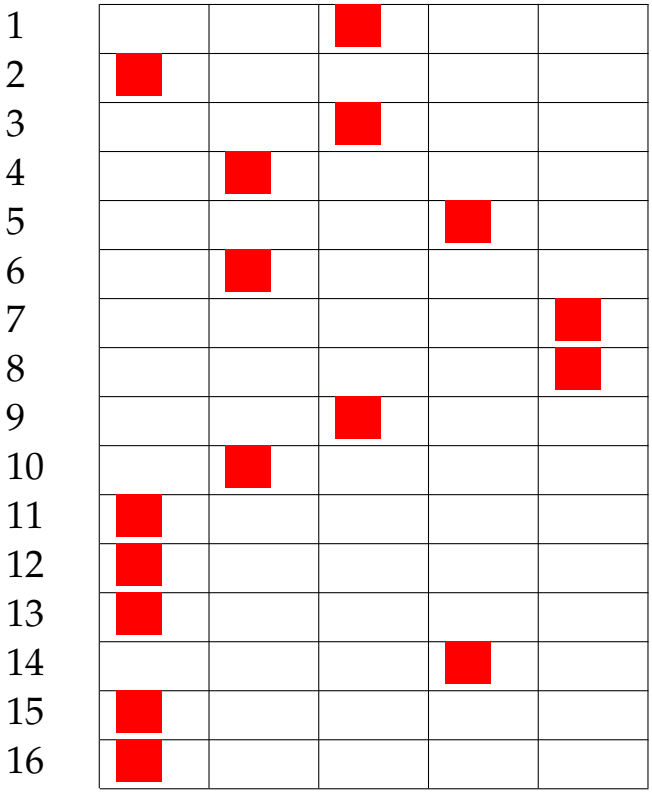
Risultato prova n. 31:



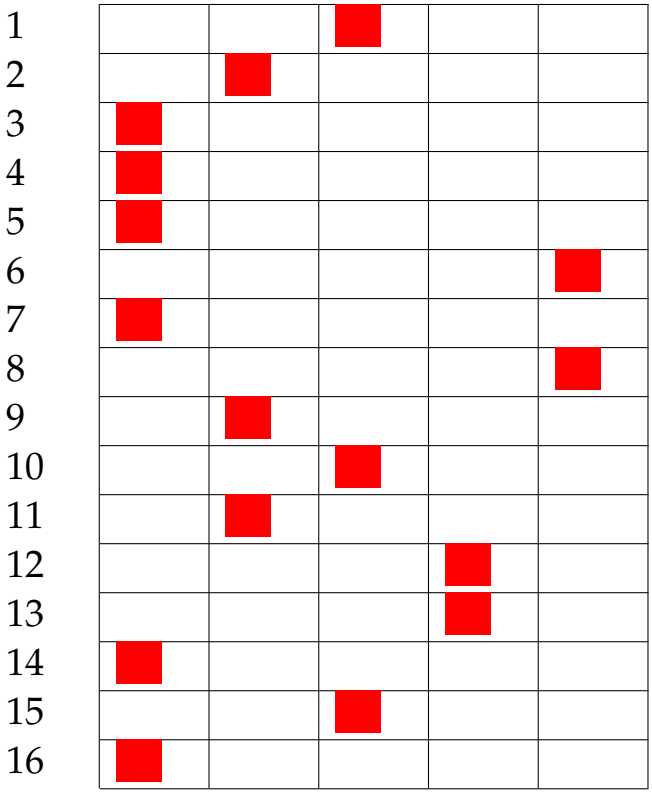
Risultato prova n. 32:



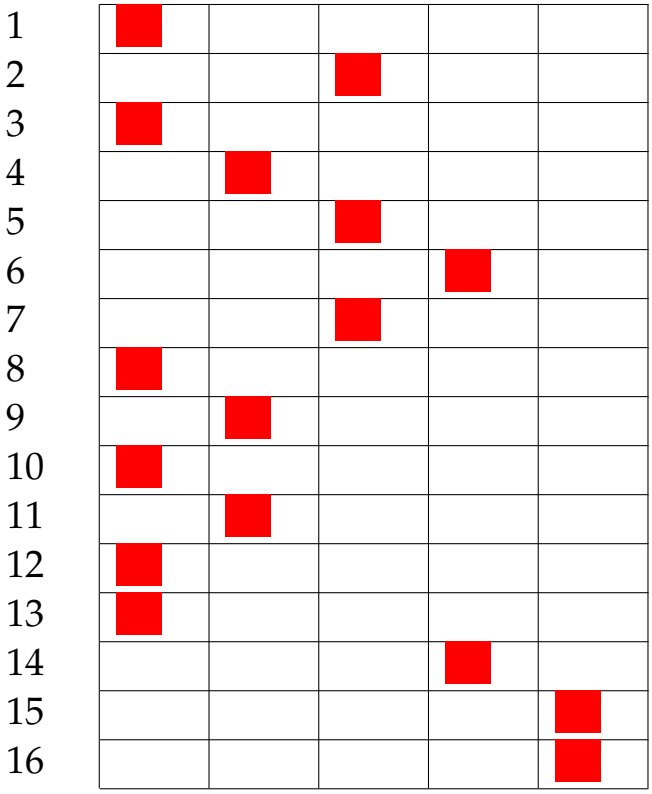
Risultato prova n. 33:



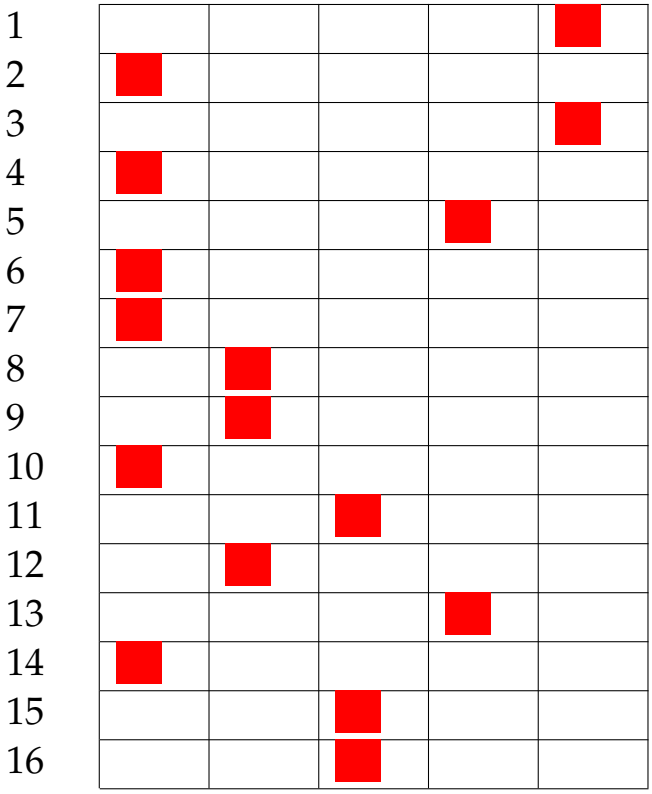
Risultato prova n. 34:



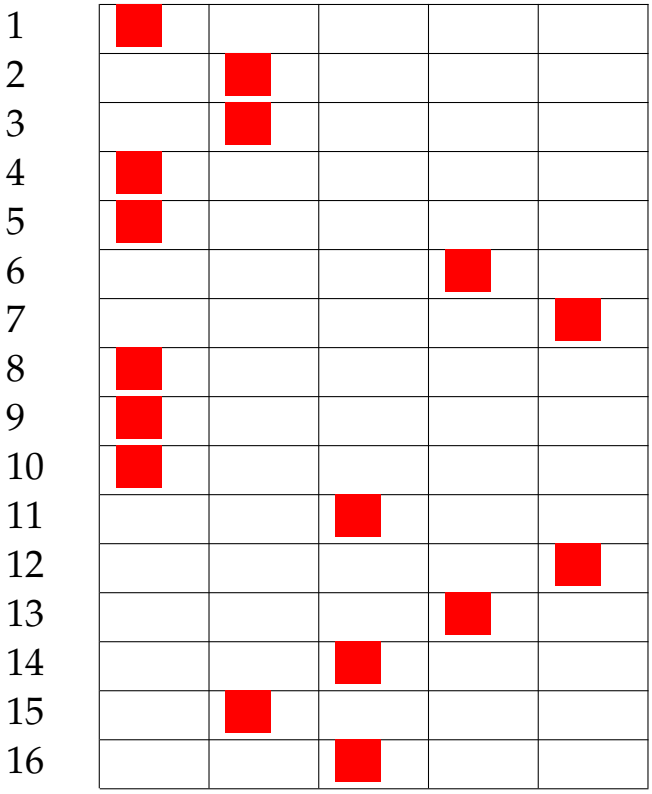
Risultato prova n. 35:



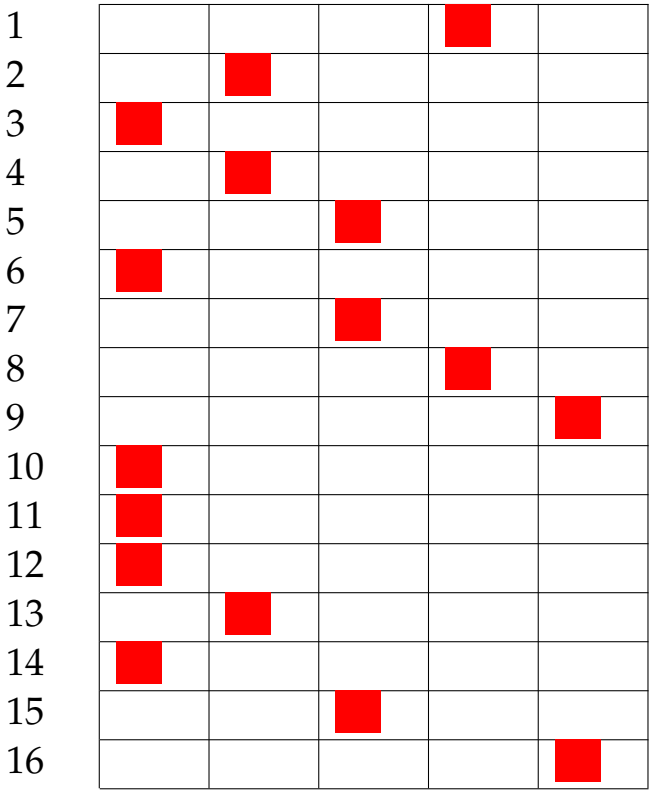
Risultato prova n. 36:



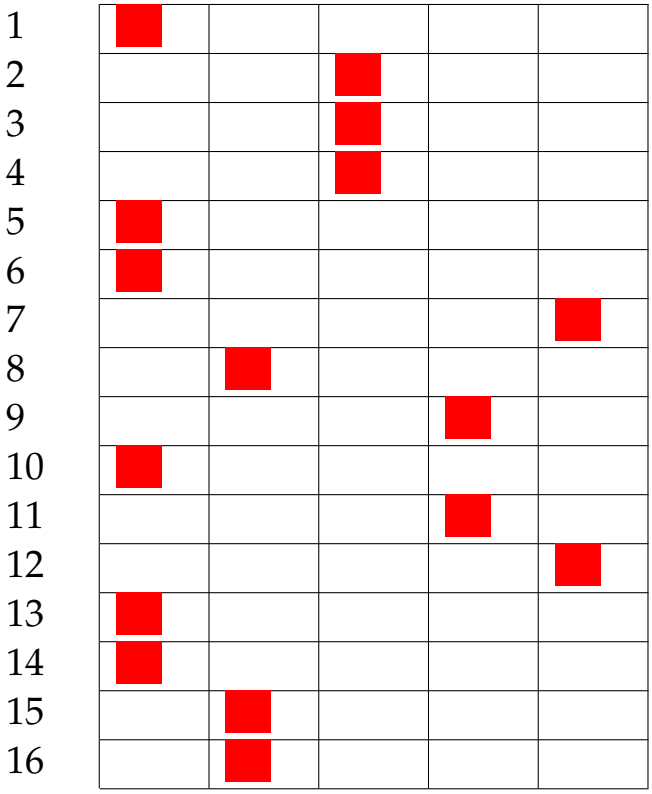
Risultato prova n. 37:



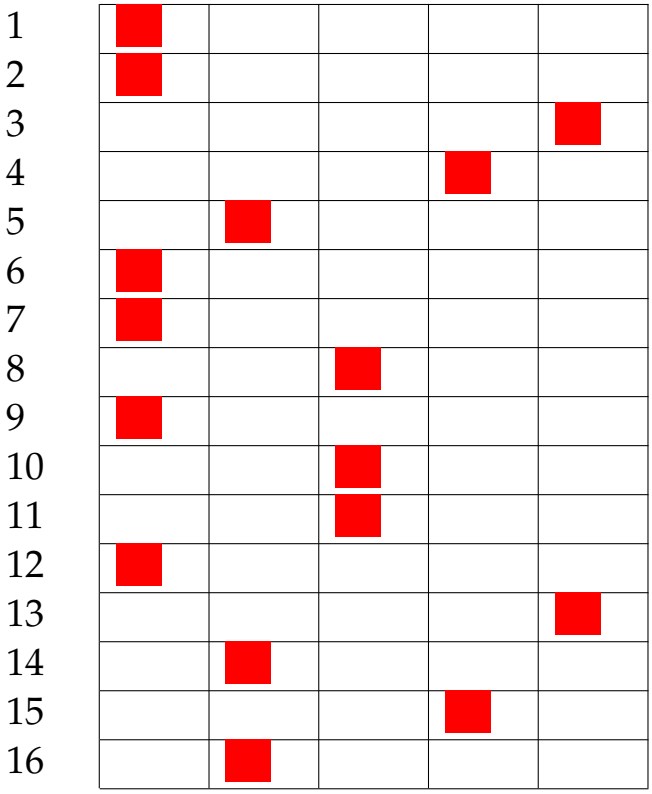
Risultato prova n. 38:



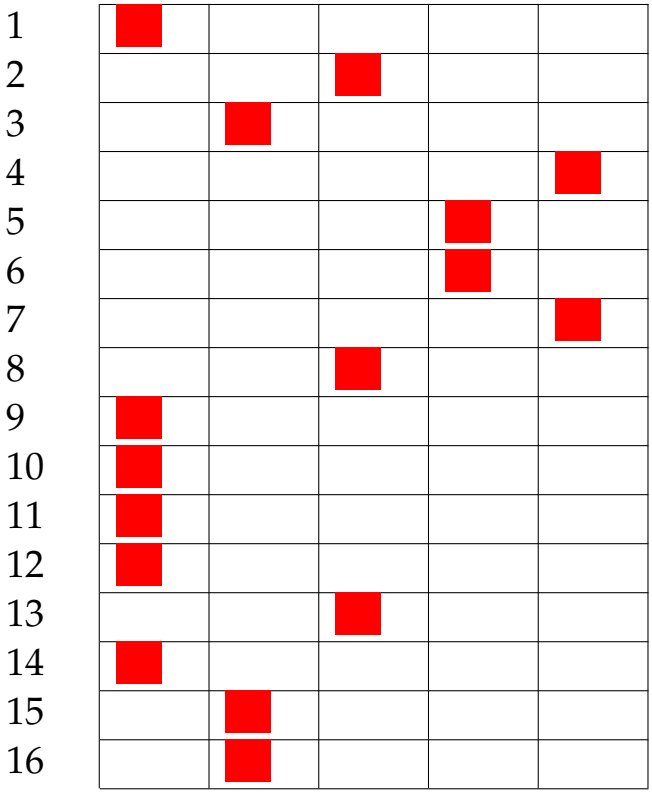
Risultato prova n. 39:



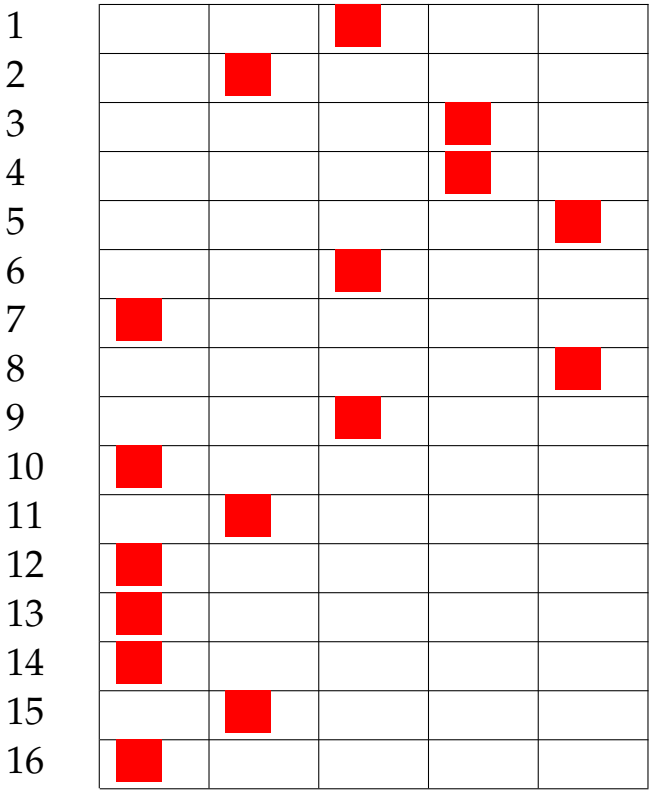
Risultato prova n. 40:



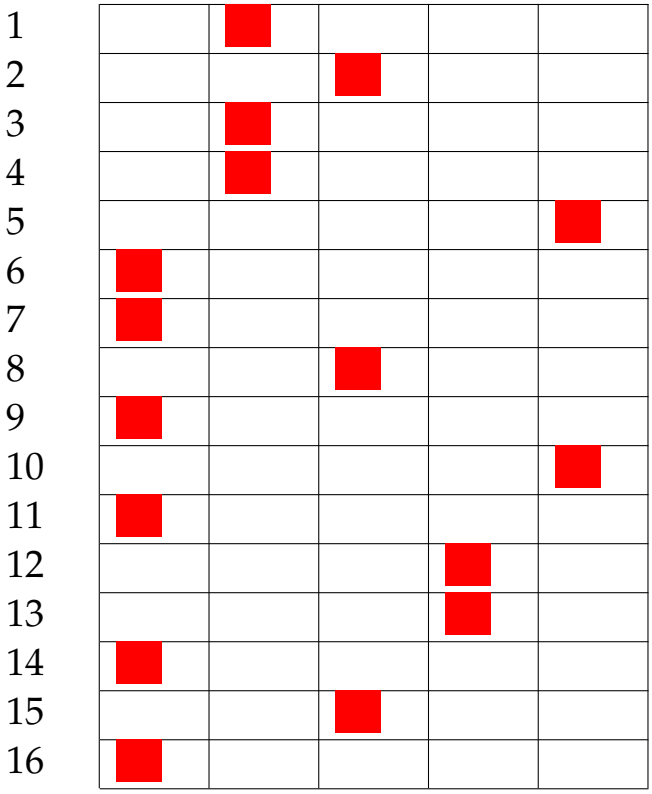
Risultato prova n. 41:



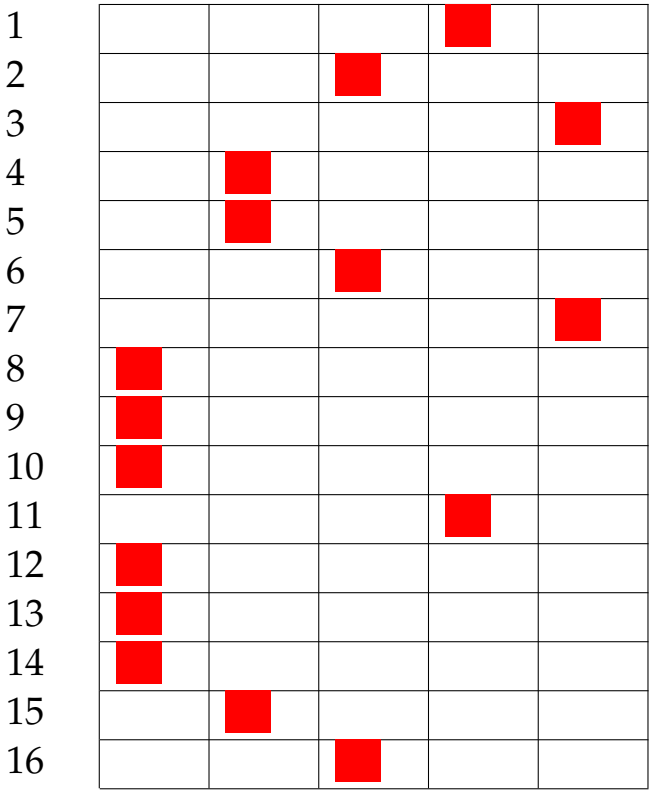
Risultato prova n. 42:



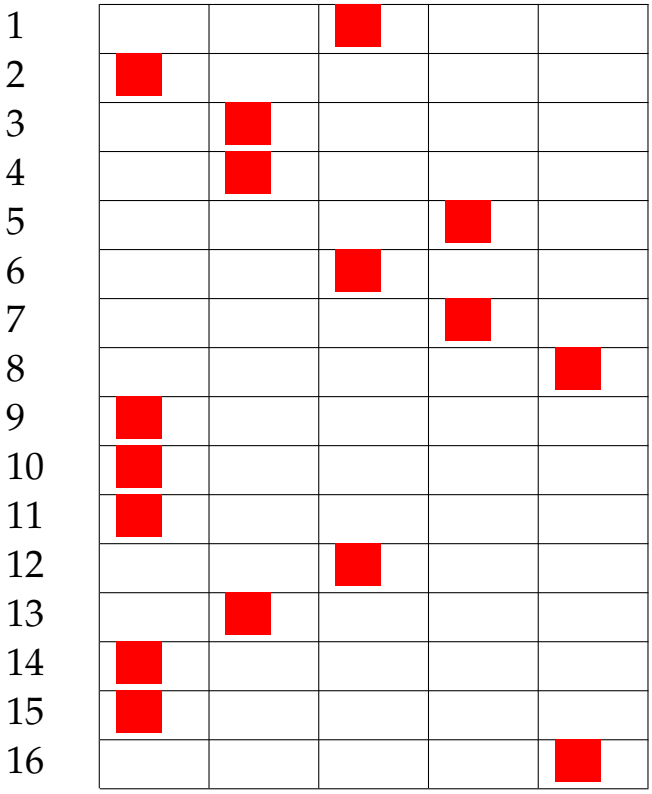
Risultato prova n. 43:



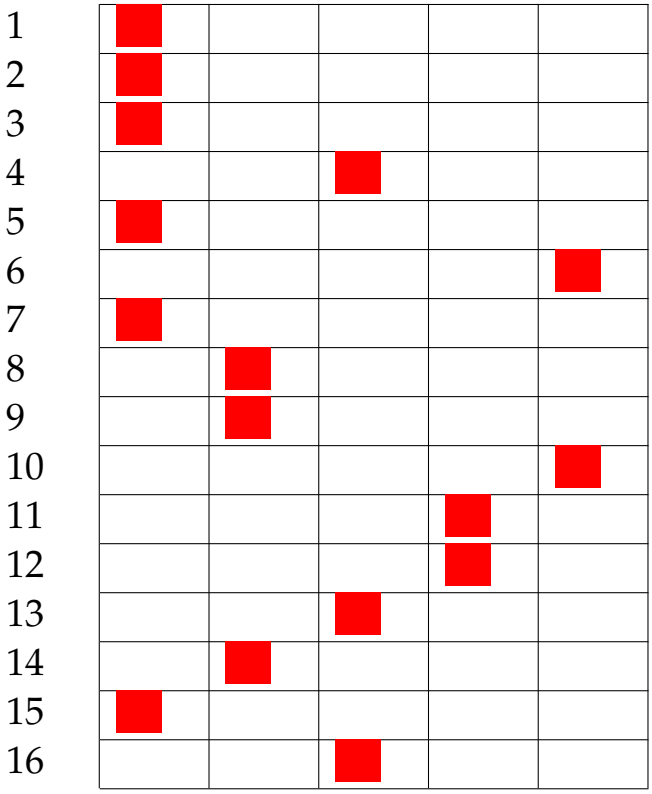
Risultato prova n. 44:



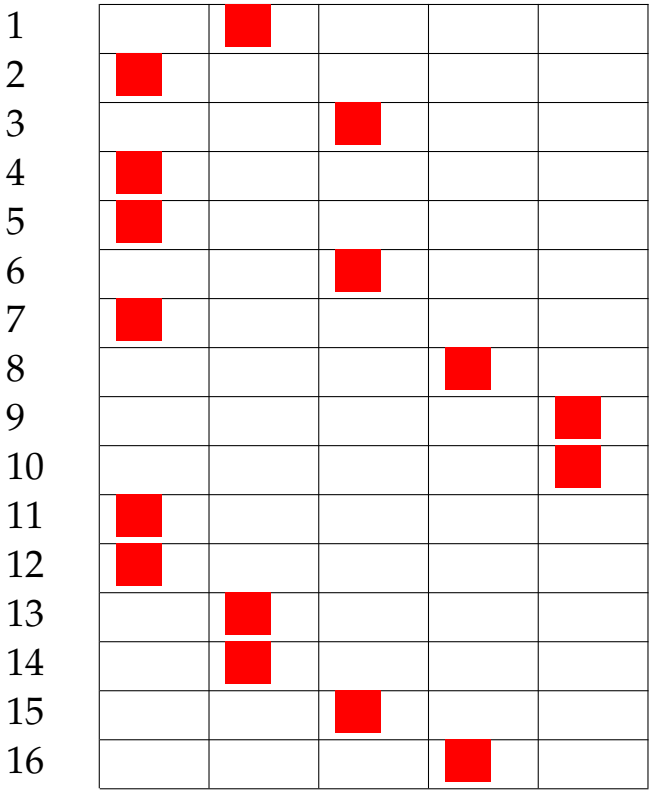
Risultato prova n. 45:



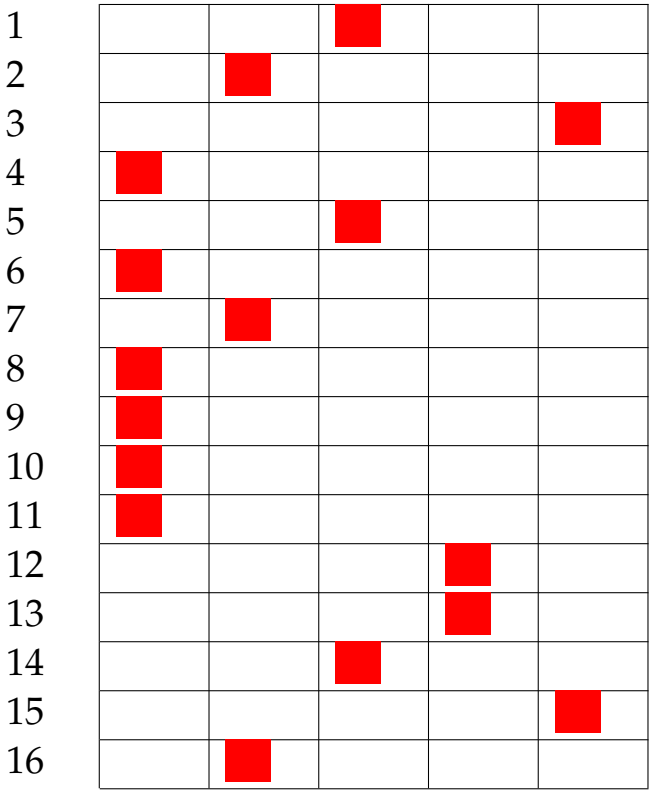
Risultato prova n. 46:



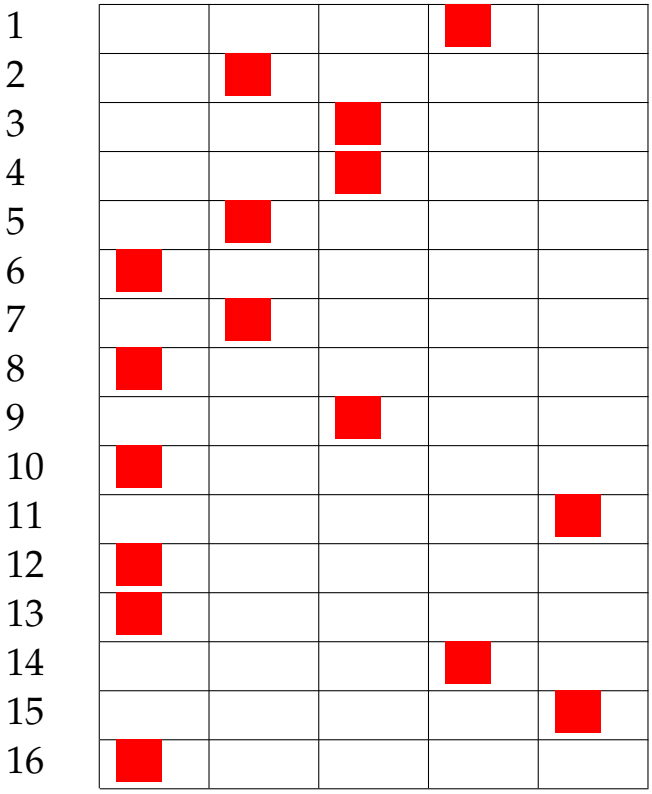
Risultato prova n. 47:



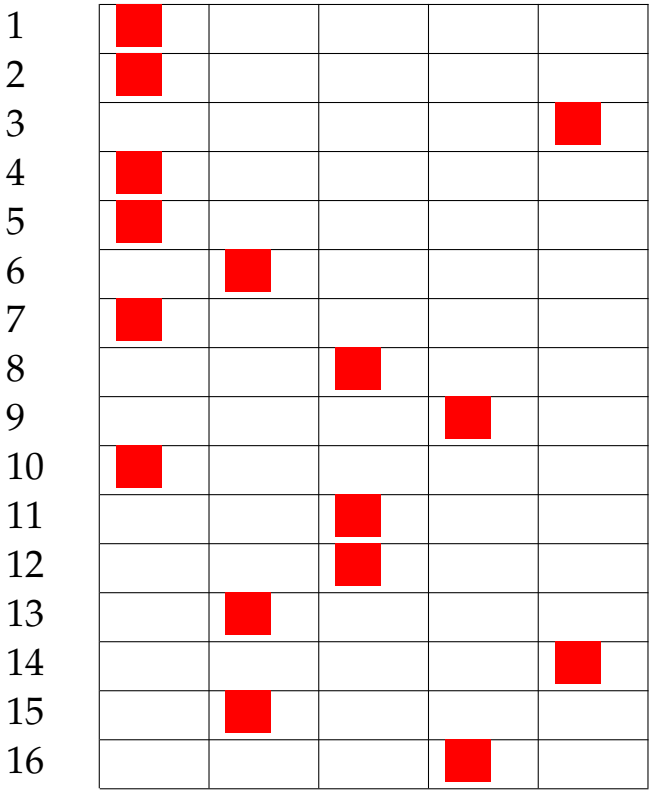
Risultato prova n. 48:



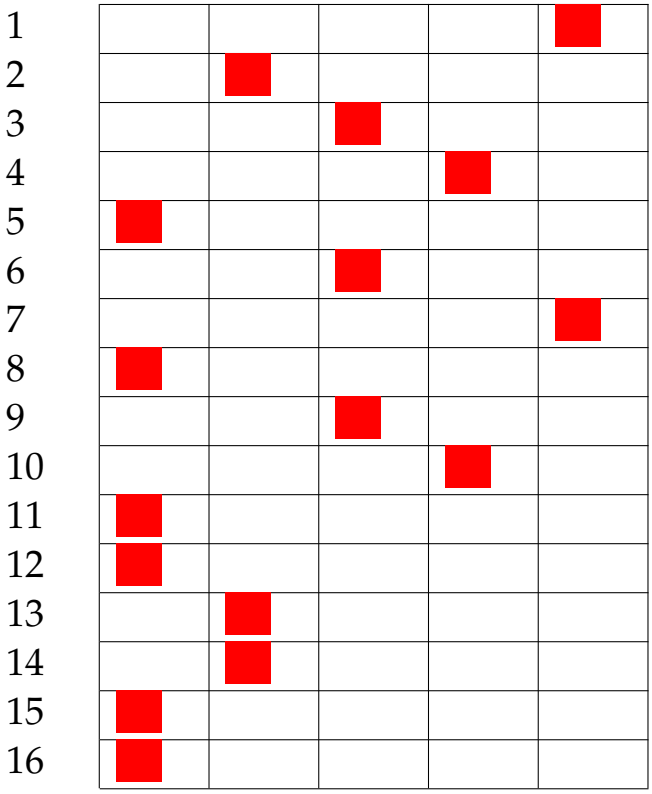
Risultato prova n. 49:



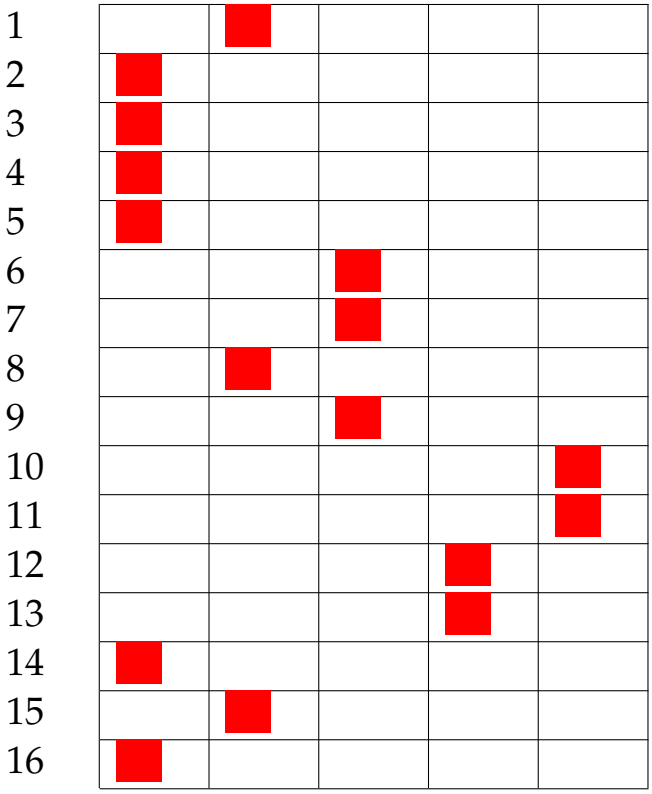
Risultato prova n. 50:



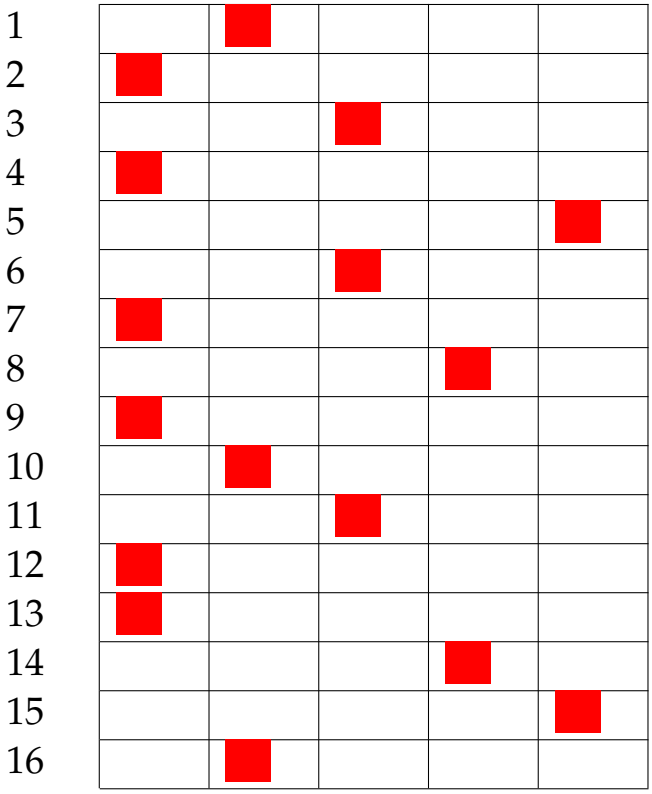
Risultato prova n. 51:



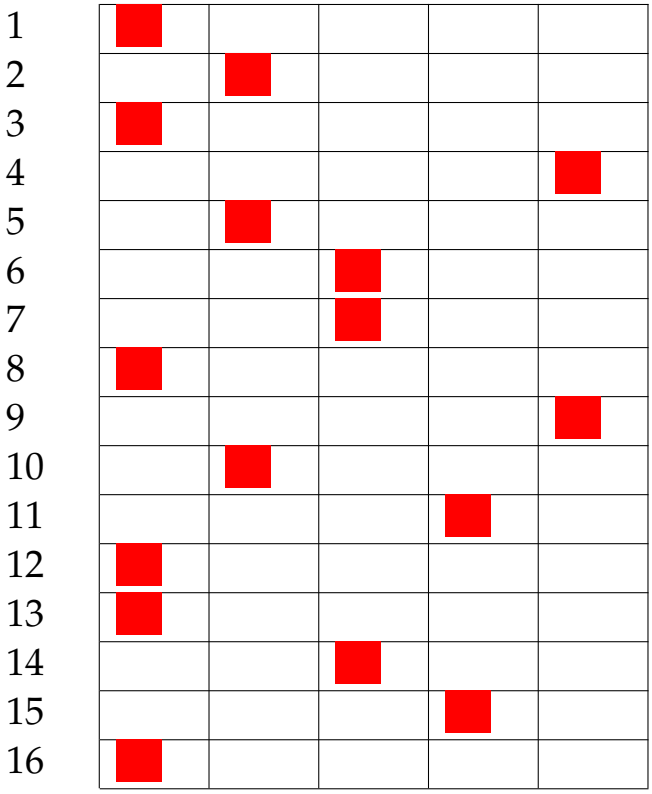
Risultato prova n. 52:



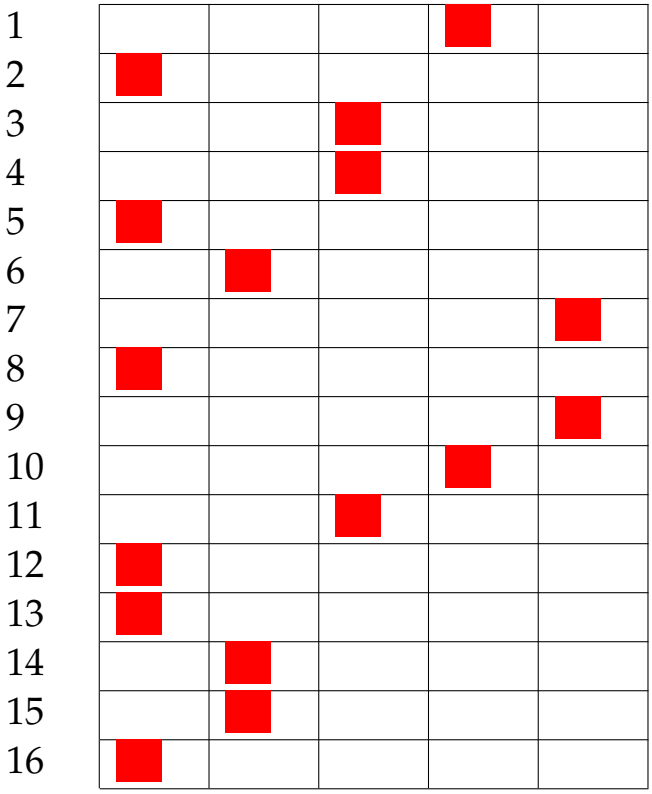
Risultato prova n. 53:



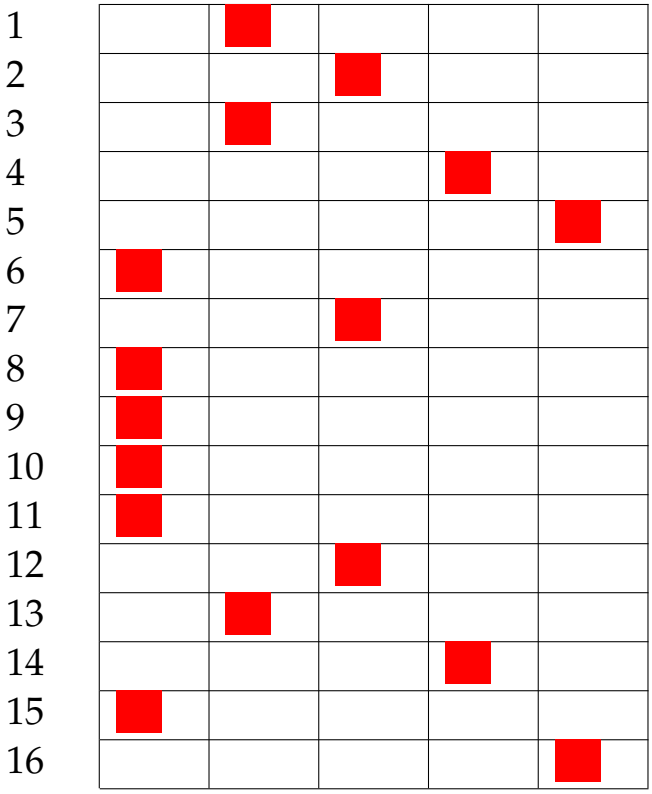
Risultato prova n. 54:



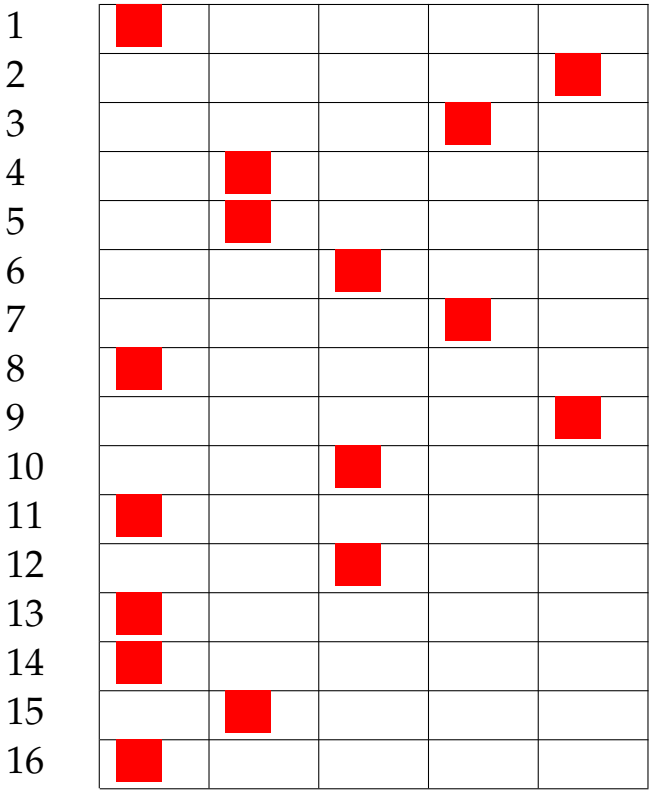
Risultato prova n. 55:



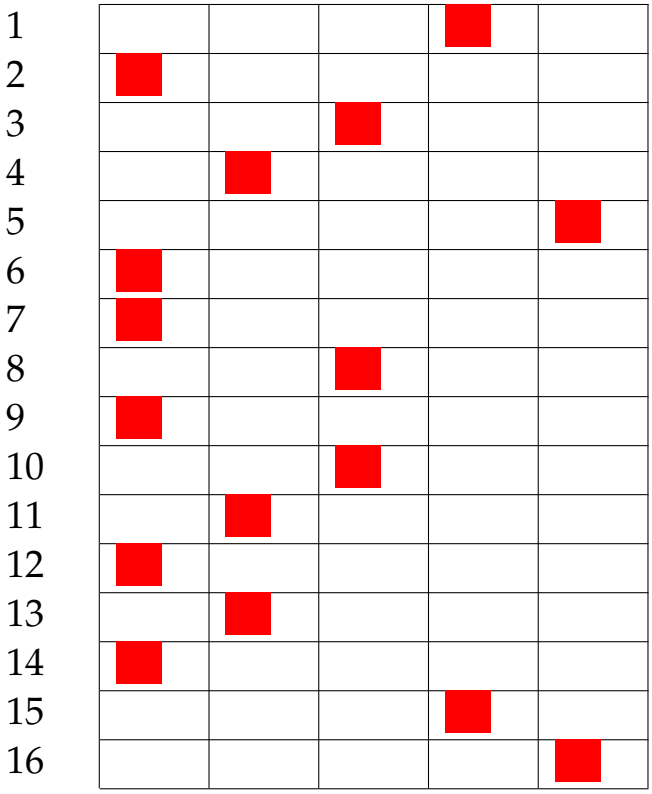
Risultato prova n. 56:



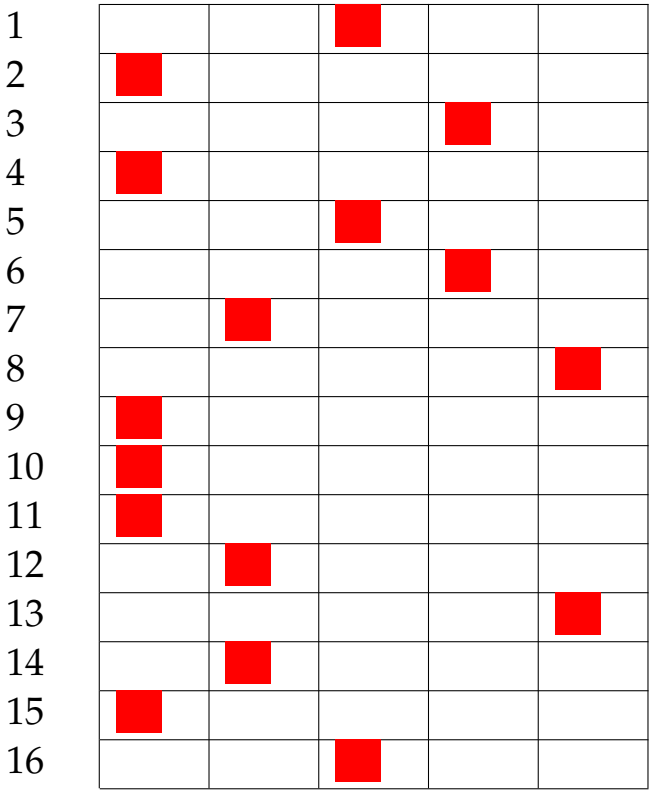
Risultato prova n. 57:



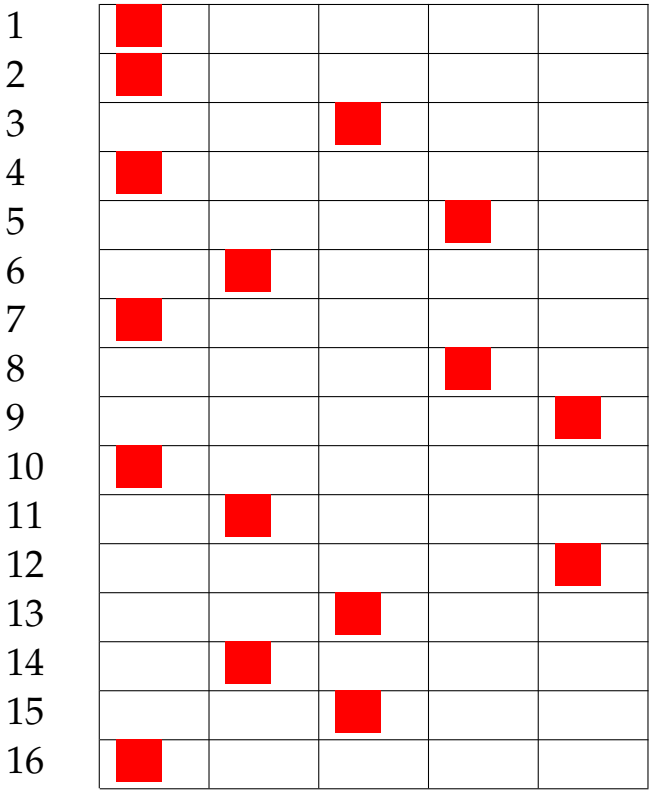
Risultato prova n. 58:



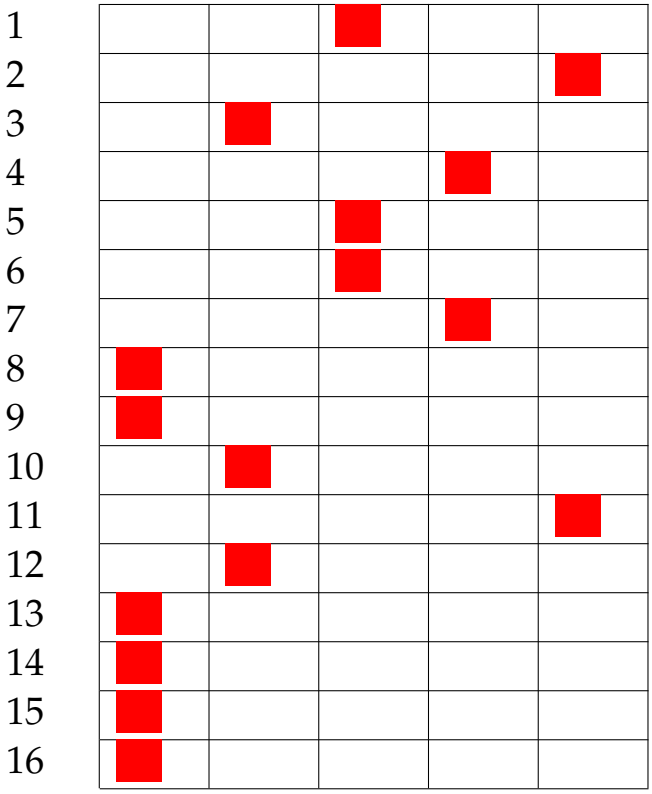
Risultato prova n. 59:



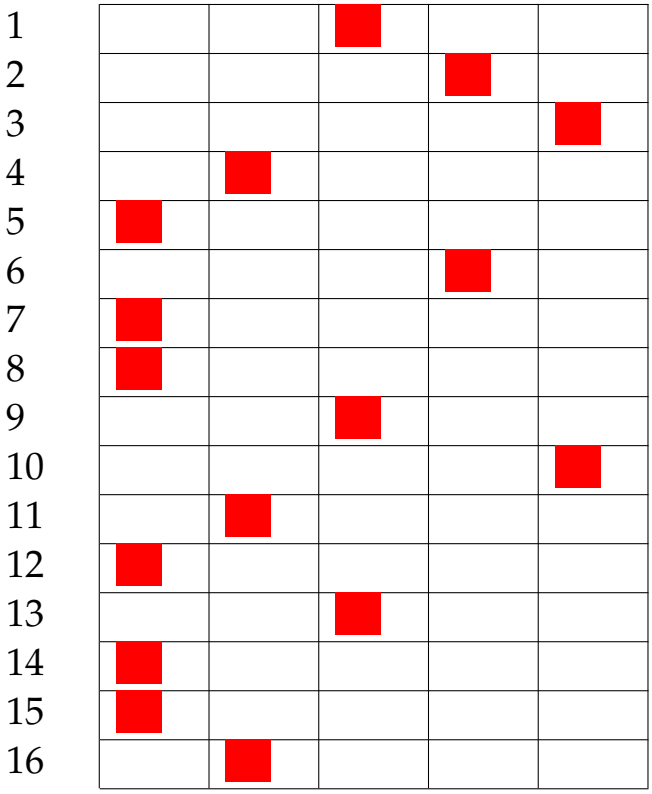
Risultato prova n. 60:



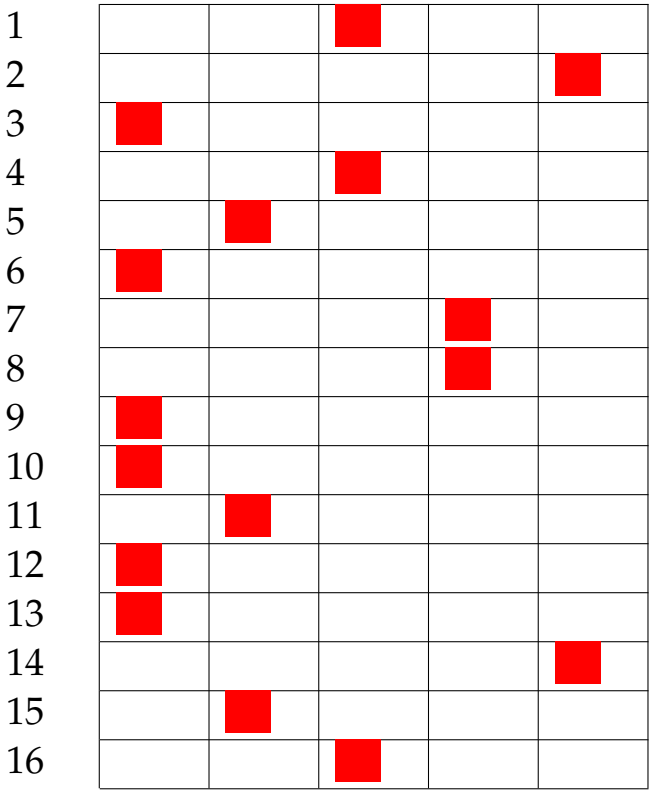
Risultato prova n. 61:



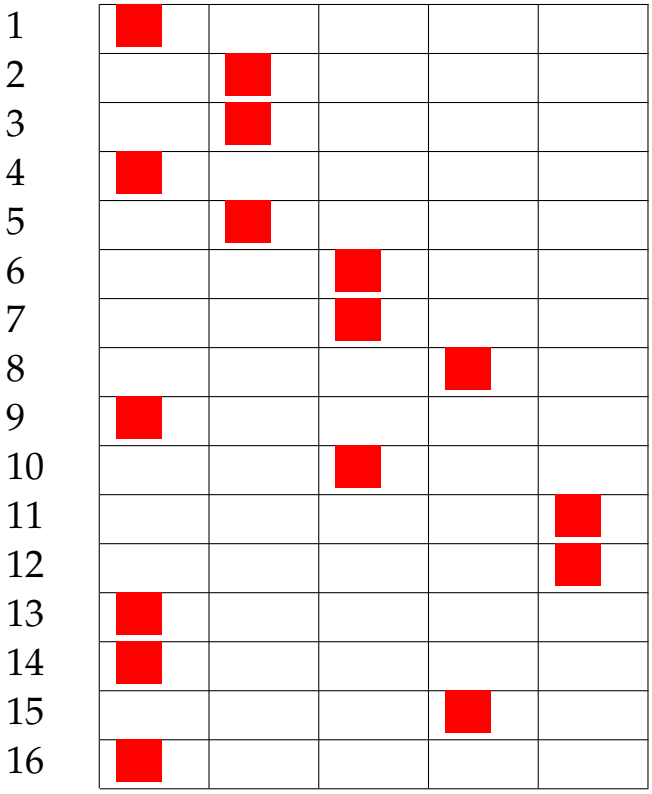
Risultato prova n. 62:



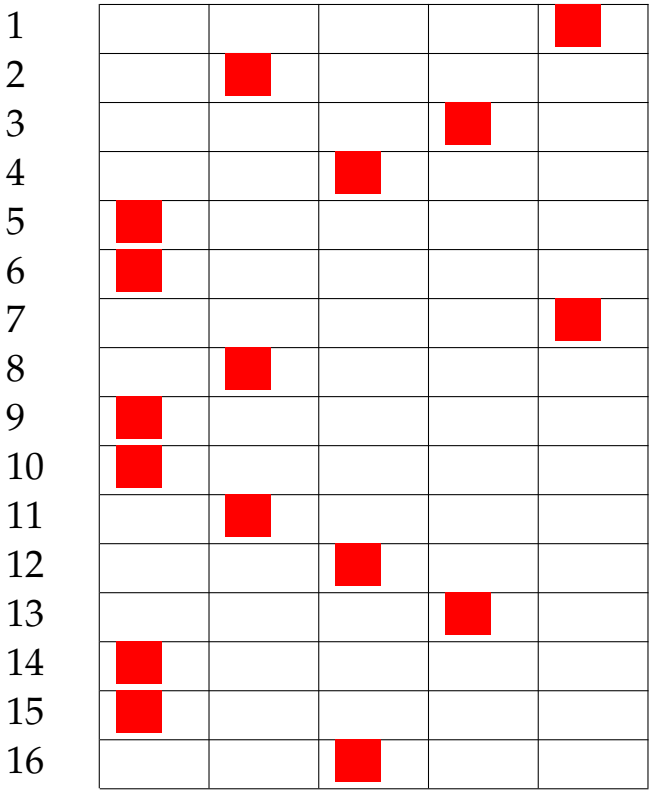
Risultato prova n. 63:



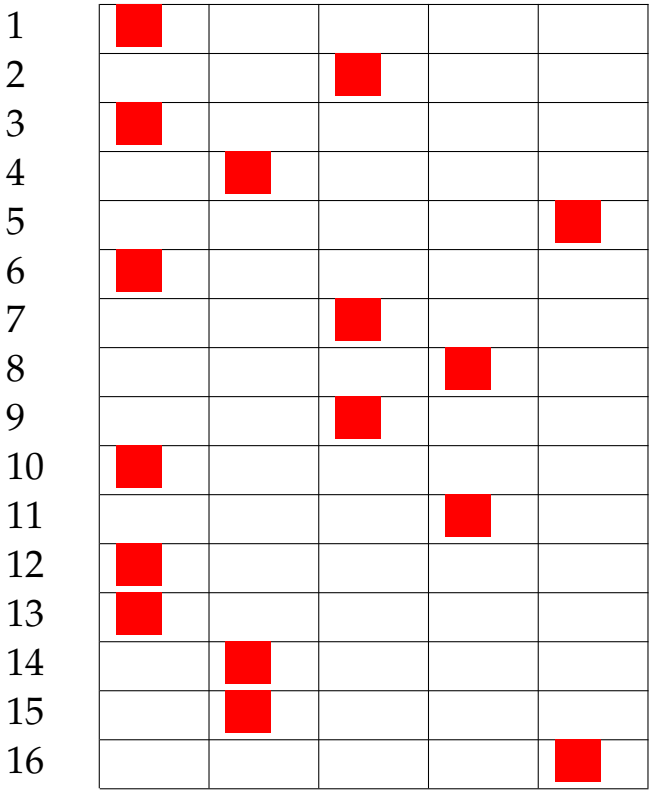
Risultato prova n. 64:



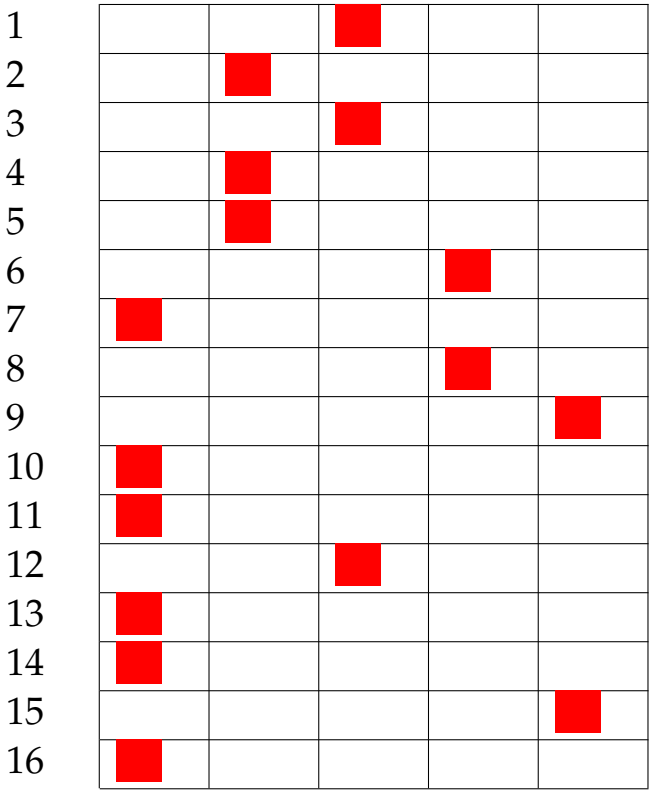
Risultato prova n. 65:



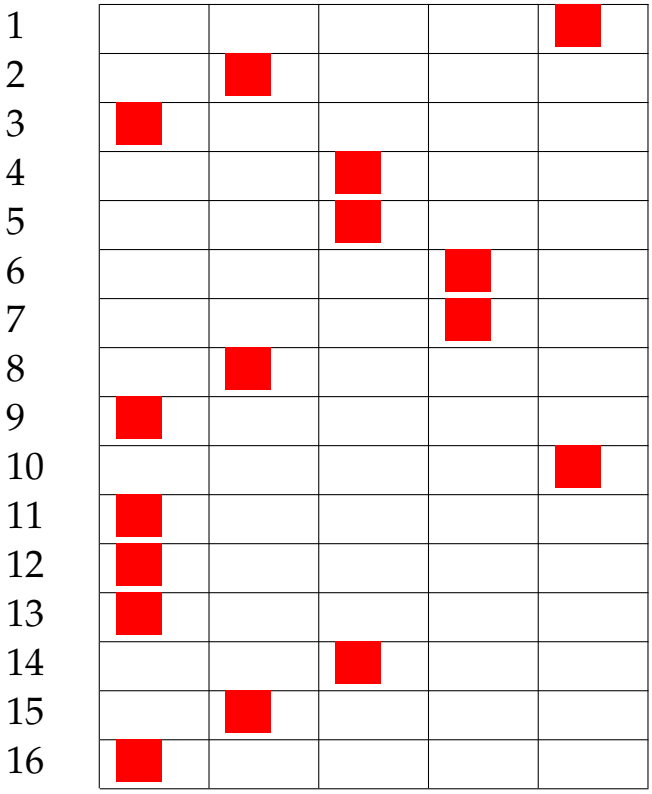
Risultato prova n. 66:



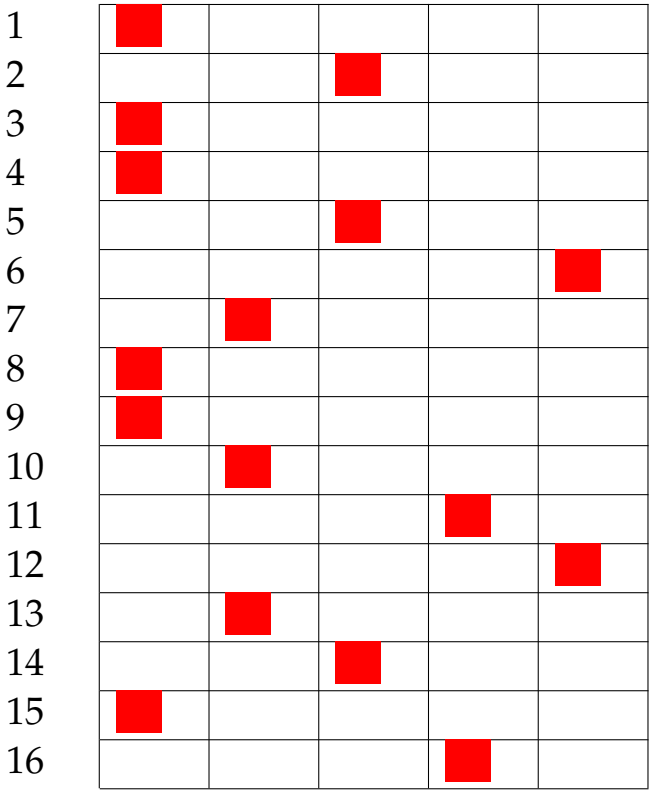
Risultato prova n. 67:



Risultato prova n. 68:



Risultato prova n. 69:



Risultato prova n. 70: