

Gruppo 1

1) Si considerino le interfacce I1, I2 e I3 e le classi A e B. Valga inoltre:

class A implements I1, I2; class B extends A implements I3;

Quale delle seguenti affermazioni è corretta?

1. La classe B può implementare solamente tutti i metodi della interfaccia I3.
2. **La classe B deve implementare tutti i metodi delle tre interfacce.**
3. Le classi A e B devono implementare tutti i metodi delle interfacce I1 e I3.
4. La classe B può implementare solo le interfacce delle classi da cui eredita.

2) Un frammento di codice usa le classi A, B (che eredita da A), e C (che eredita da B).

A a; B b; C c;

1. Il polimorfismo consente di assegnare a = a; b = b; c = c;
2. Il polimorfismo non consente di assegnare a = a; b = b; c = c;
3. **Il polimorfismo verticale consente di assegnare a = b; a = c; b = c;**
4. Il polimorfismo consente di assegnare b = a; c = a; c = b;

3) Sono state definite in questo ordine:

una interfaccia IA;

una classe A astratta che implementa la interfaccia IA e che ha alcuni metodi implementati;

una classe SA astratta che eredita da A e che non implementa la interfaccia IA.

Quale affermazione è vera?

1. **Non è possibile che la classe SA che eredita da A non implementi tutti i metodi della interfaccia IA.**
2. Non sono rispettati i vincoli temporali di sviluppo.
3. Non è possibile che una classe astratta SA abbia alcuni metodi implementati.
4. Si possono creare istanze dalla classe SA, ma non dalla classe A.

4) Quale delle seguenti definizioni di filtro è la più corretta:

1. Il filtro è un programma che fa staticamente i controlli sulla esistenza dei file da leggere e sulla possibilità di creare i file di uscita e lavora in modo veloce durante la esecuzione.
2. Il filtro è un programma che al momento delle esecuzione deve controllare gli argomenti passati sulla linea di comando e lavora trattando tutto il file di ingresso; terminata questa fase porta le informazioni necessarie sul file di uscita.
3. **Il filtro è un programma che al momento delle esecuzione deve controllare gli argomenti passati sulla linea di comando e lavora trattando tutto il file di ingresso fino a produrre un file di uscita; un filtro ben fatto tende a impegnare poche risorse di memoria e a portare le informazioni in uscita prima possibile.**
4. Il filtro è un programma che al momento delle esecuzione deve controllare gli argomenti passati sulla linea di comando, lavorando sui file di ingresso e di uscita; un filtro ben fatto tende a impegnare poche risorse di memoria e a portare le informazioni dal file di uscita sul file di ingresso prima possibile.

5) Si considerino due classi A e B non in relazione tra loro, e una classe C che eredita da A e ha una variabile di classe B.

Quale affermazione è corretta?

1. Avere la classe C compilata (C.class) implica che almeno la classe A sia presente in forma compilata (A.class).
2. Avere la classe C compilata (C.class) implica che almeno la classe B sia presente in forma compilata (B.class).
3. **Avere la classe C compilata (C.class) implica che entrambe le classi A e B siano presenti in forma compilata (A.class, B.class).**
4. Avere la classe C compilata (C.class) non implica affatto che le classi A e B siano presenti in forma compilata (A.class, B.class).

6) Un frammento di codice usa le classi ClasseA, ClasseB (che eredita da ClasseA), e ClasseC (che eredita da ClasseB). Tutte le classi implementano la interfaccia Intl.

ClasseA a; ClasseB b1, b2; ClasseC c; Intl i;

Quale affermazione è falsa?

1. È scorretto assegnare: c = b2;
2. È scorretto assegnare: c = i;
3. **È scorretto assegnare: b2 = (ClasseB) i;**
4. È scorretto assegnare: b1 = (ClasseC) i;

7) Si consideri la clausola seguente:

package GraphicalUtilities;

all'interno di un file sorgente. Quale delle seguenti affermazioni è più corretta?

1. La clausola permette di raggruppare alcune classi (e/o oggetti) logicamente allacciate in modo da consentire una visibilità esterna ampliata.
2. **La clausola permette di raggruppare classi (e/o oggetti) logicamente interallacciate e con riferimenti reciproci in modo da consentire una visibilità interna specifica.**
3. La clausola permette di raggruppare alcune classi non in ereditarietà ma logicamente interallacciate e con riferimenti reciproci.
4. La clausola permette di raggruppare solo oggetti specifici le cui classi siano logicamente interallacciate e con riferimenti reciproci in modo da consentire una visibilità interna specifica.

8) Sia p un oggetto di classe JPanel e b1, b2 due oggetti di classe JButton. Dato il codice:

b1.addActionListener(p);

b2.addActionListener(p);

quale affermazione risulta vera?

1. L'oggetto di classe JPanel si registra come gestore degli eventi(ActionEvent) di entrambi i bottoni e deve mettere a disposizione due metodi actionPerformed(). Ad ogni pressione di uno dei due bottoni, il panel p esegue il metodo relativo per trattare il bottone specificato.
2. L'oggetto di classe JPanel si registra come gestore degli eventi(ActionEvent) di entrambi i bottoni. Alla pressione contemporanea di entrambi i bottoni, il panel p esegua il metodo actionPerformed().
3. L'oggetto di classe JPanel si registra come gestore degli eventi(ActionEvent) di entrambi i bottoni. Ad ogni pressione di uno dei due bottoni, il panel p viene ridisegnato tramite l'invocazione automatica del metodo show() del frame di appartenenza.
4. **L'oggetto di classe JPanel si registra come gestore degli eventi(ActionEvent) di entrambi i bottoni. Ad ogni pressione di uno dei due bottoni, il panel p mette in esecuzione il metodo actionPerformed() che può distinguere quale dei due bottoni è stato premuto e agire conseguentemente.**

9) Che cosa produce il frammento di codice:

```
public class ExPanel extends JPanel {
    public ExPanel(){
        super();
        JLabel l = new JLabel("Lab1");
        JButton b = new JButton("Lab1");
        add(l); add(b);
        b.addActionListener(this);
    }
    ...
}
```

1. **La classe ExPanel eredita dal pannello predefinito e aggiunge alla istanza del pannello sia una label, sia un bottone, che vengono visualizzati. Essa si incarica anche di fare in modo che la gestione del bottone sia delegata a un metodo actionPerformed() sulla istanza.**
2. La classe ExPanel eredita dal pannello predefinito e aggiunge alla classe pannello sia una label, sia un bottone, che vengono visualizzati direttamente dalla classe che si incarica anche di gestire il bottone con una actionPerformed().
3. La classe ExPanel eredita dal pannello predefinito e aggiunge al pannello sia una label, sia un bottone, che non vengono visualizzati fino alla esecuzione della addActionListener().
4. La classe ExPanel non compila perché non si possono avere due componenti grafici che fanno riferimento alla stessa stringa "Lab1".

10) Si consideri un main, invocato con la linea di comando

```
pack.MainClass abc 123 pippo
public class MainClass {
    public static void main(String[] args){
        char as [3] = {'v','a','i'};
        for (int i=0; i < args.length && i < 2; i++)
            {as[i] = args[i].charAt(i);}
        System.out.println(as); }
}
```

1. Il processo di esecuzione stampa ia2.
2. Il processo di esecuzione stampa a i.
3. Il processo di esecuzione stampa a l p.
4. **Il processo di esecuzione stampa a2i.**

11) Che effetto produce il seguente blocco di codice se la stringa s contiene "1234"?

```
try { double a = Double.parseDouble(s); }
catch (NumberFormatException e) {
    System.out.println("Stringa mal fatta"); }
```

1. Il programma produce una eccezione, stampa un messaggio di errore e termina.
2. Il programma produce una eccezione, stampa un messaggio di errore e continua.
3. **Il programma produce in a il valore 1234.0, che deriva dal parsing di s.**
4. Il programma produce una eccezione e termina.

12) Date le seguenti classi:

```
class A {
```

```
A va; B vb;
public A () { super(); }
public void setA (A a) { va = a; }
public void setB (B b) { vb = b; }
public A getA () { return va; }
public B getB () { return vb; }
}
class B {
    A va; B vb;
    public B () { super (); }
    public void setA (A a) { va = a; }
    public void setB (B b) { vb = b; }
    public A getA () { return va; }
    public B getB () { return vb; }
}
```

Quale affermazione è vera?

1. Le due classi sono del tutto equivalenti e non possono fare parte dello stesso package entrambe.
2. **Le due classi sono del tutto equivalenti: se si creano istanze da entrambe, non è possibile fare assegnamenti dall'una all'altra.**
3. Le due classi sono del tutto equivalenti: se si creano istanze da entrambe, si può assegnare da una all'altra.
4. Le due classi non sono del tutto equivalenti; lo sarebbero se derivassero dalla stessa superclasse.

13) Siano definite le classi seguenti in file sorgenti differenti:

```
public class Test1 {
    public float aMethod(float a, float b) throws IOException { }
}
```

```
public class Test2 extends Test1 { }
```

Quale dei seguenti metodi sarebbe corretto e in overriding se inserito nella classe Test2?

1. float aMethod(float a) { }
2. nessuno
3. public int aMethod(int a, int b) throws IOException { }
4. **public float aMethod(float a, float b) throws IOException { }**

14) Considerate le seguenti classi, definite in file sorgenti separati:

```
public class Base {
    public void method(int i) {
        System.out.println("Valore è " + i); }
}
public class Estende extends Base {
    public void method(int j) { System.out.println("Questo valore è " + j); }
    public void method(String s) { System.out.println("Mi è stato passato " + s); }
    public static void main(String args[]) {
        Base b1 = new Base();
        Base b2 = new Estende();
        b1.method(5); b2.method(6); }
}
```

Che cosa viene scritto alla console all'esecuzione del metodo main della classe Sub?

1) **Valore è 5**

Questo valore è 6

2) Valore è 5

Valore è 6

3) Questo valore è 5

Questo valore è 6

4) Mi è stato passato 5

Mi è stato passato 6

15) Data la seguente classe:

```
public class Test extends Base {  
    public Test(int j) { }  
    public Test(int j, int k) { super(j, k); }  
}
```

Quale dei seguenti costruttori deve esistere esplicitamente nella definizione della classe Base?

1. Base(int p, int q) { }

2. Base() { }

3. Base(int j) { }

4. Base(int j, int k, int l) { }