Corso di laurea in Comunicazione digitale — Programmazione I

AA 2009/10 — Prof. M.A. Alberti

II Appello - 22 febbraio 2010

Cognome **Nome**

Matricola

1 (9 punti)

È data la classe `Villaggio` che modella un insediamento composto da case per uso abitativo e edifici industriali, di cui si rappresenta la tecnica di costruzione e la tipologia del prodotto fabbricato. Il metodo `main` della classe potrebbe fornire, ad esempio, il seguente output, dopo aver caricato i dati relativi a 4 edifici:

```
Villaggio di 4 costruzioni.
Edificio di 4 piani: civile, cemento armato
Edificio di 3 piani: civile, cemento armato
Fabbrica di birra: industriale, mattoni
Fabbrica di mobili: industriale, prefabbricato
```

Il metodo `main` della classe che genera l'output di cui sopra:

```
public class Villaggio {
   static final int MAX = 20;
   public static void main (String[] args) {
      Costruzione[] costruzioni = new Costruzione[MAX];
      costruzioni[0]= new  Edificio (4)    ;
      costruzioni[1]= new  Edificio (3)    ;
      costruzioni[2]= new  Capannone("birra", "mattoni") ;
      costruzioni[3]= new  Capannone("mobili", "prefabbricat0");
      System.out.println("Villaggio di "
             + Costruzione.numeroCostruzioni() + " costruzioni.");
      for (int i = 0; i<Costruzione.numeroCostruzioni(); i++)
                  System.out.println(costruzioni[i]);
   }
}
```

a. Considerate la classe `Costruzione`:

```
abstract class Costruzione {
   private String tipo;
   private static int numero;

   Costruzione(String t)  {
     tipo = t
     numero++;
    }
  abstract String tecnicaCostruzione();
  public String toString(){
     return tipo + ", " + this.tecnica_costruzione();
```

```
  }
  public static int numeroCostruzioni()  {
     return numero;
  }
}
```

b. E' corretta l'istruzione `costruzione[n] = new Costruzione("civile");` ?
SI **NO** **Perché la classe Costruzione è astratta e quindi non può essere istanziata. Notate che può avere, come in questo caso, un costruttore.**

c. La classe `Edificio` estende la classe `Costruzione` e ha intestazione
`class Edificio extends Costruzione { … }`
Di conseguenza cosa è necessario che contenga il codice della classe?
Poiché dalla intestazione si deduce che la classe `Edificio` non è astratta ne consegue che questa deve fornire l'implementazione del metodo astratto `tecnicaCostruzione`.

d. Gli oggetti della classe `Edificio` sono caratterizzati dal numero di `piani`, sono sempre di tipo `civile` e realizzati con tecnica di costruzione `cemento armato`. Si completi il codice di conseguenza:

```
class Edificio extends Costruzione{
   int piani;
   Edificio(int p){
      super("civile");
      piani = p;
   }
    public String tecnicaCostruzione(){
       return "cemento armato";
   }
}
```

e. Definite ora una classe `Capannone` che modella una costruzione di tipo `industriale`, di cui si rappresenta la tipologia della produzione (membro `prodotto`) e la tecnica di costruzione del capannone (membro `tecnica`)

```
class Capannone extends Costruzione {
   private String prodotto;
   private String tecnica;
   Capannone (String tipoProduzione, String tecnicaCostruzione){
      super("industriale");
      prodotto = tipoProduzione;
      tecnica = tecnicaCostruzione;
   }
   public String tecnicaCostruzione() {
      return tecnica;
   }
```

f. Ora completate opportunamente (tenendo conto dell'output) il codice del metodo `main`

---

2 (3 punti)

Assumete che le variabili **a** e **b** siano definite **boolean**. Scegliete tra le situazioni seguenti quella che descrive il fatto che l'espressione **!(a && b) && (a || b)** venga valutata vera.

a. Sempre
b. Mai
c. Quando sia **a** che **b** sono **true**
d. Quando nè **a** nè **b** sono **true**
e. **Solo quando una sola delle variabili, a o b, è true**

a	b	a && b	!(a && b)	a || b	!(a&&b)&&(a||b)
F	F	F	T	F	F
F	T	F	T	T	T
T	F	F	T	T	T
T	T	T	F	T	F

---

3 (4 punti)

Nella classe `Intervallo`, definire il metodo di classe `appartiene(int n)` che riporta un valore booleano per indicare se il numero intero passato come parametro appartiene agli insiemi disgiunti `[a, b]` e `[c, d]`, dove i valori `a, b, c` e `d` sono definiti costanti.

```
public static boolean appartiene(int n){
    return (n >= a && n <= b || n >=c && n <= d);
}
```

---

4 (3 punti)

Considerate i due frammenti di codice seguenti (assumete **x** una variabile di tipo **int**):

```
while ( x>0 ) { x-- }                    // frammento 1
System.out.println("x = " + x);

do ( x-- ) while ( x>0 );                // frammento 2
System.out.println("x = " + x);
```

In quale tra le seguenti circostanze l'output dei due frammenti è diverso?

I. **x** è **0** prima che il segmento sia eseguito
II. **x** è maggiore di **0** prima che il segmento sia eseguito
III. **x** è minore di **0** prima che il segmento sia eseguito

Risposta:

a. solo I b. solo II c. solo III d. I e II e. **I e III**

---

5 (4 punti)

Generare `DIM` numeri pseudo casuali interi positivi pari e minori di `MAX`. Archiviarli in un array che verrà riportato all'ambiente chiamante.

```
final static int DIM = 5;
```

```
final static int MAX = 100;

public static int[] popola (){
   int[] archivio = new int[DIM];
   int numero;
   for (int i=0; i<DIM; i++){
      do
         numero = (int)(Math.random()* MAX);
      while (numero%2 != 0);
      archivio[i] = numero;
    }
    return archivio;
}
```

---

6 (3 punti)

Considerate le seguenti classi:

```
class Veicolo { … }
class Auto extends Veicolo { … }
class SUV extends Veicolo { … }
```

1.	`Veicolo v = new Auto();`	V	F
2.	`Veicolo v = new SUV();`	V	F
3.	`Auto a = new SUV();`	V	F
4.	`SUV s = new Auto();`	V	F

---

7 (4 punti)

Considerate il seguente metodo:

```
public static void mistero (int [] a, int [] b) {
    for (int i = 0; i <a.length; i++)
        a[i] += b [b.length – 1 – i];
}
```

Calcolate i valori dell'array **a1** dopo che è stato eseguito il codice **mistero(a1, a2)**

```
int [] a1 = { 1, 3, 5, 7, 9 }
int [] a2 = { 1, 4, 9, 16, 25 }
```

a1 = { 26, 19, 14, 11, 10 }