

## Il 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                                ;  
        costruzioni[1]= new                                ;  
        costruzioni[2]= new                                ;  
        costruzioni[3]= new                                ;  
  
        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:

- 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?

- 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){  
  
    }  
    public String tecnicaCostruzione(){  
  
    }  
}
```

- 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){  
  
    }  
    public String tecnicaCostruzione() {  
  
    }  
}
```

- 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**

<b>a</b>	<b>b</b>	<b>a &amp;&amp; b</b>	<b>!(a &amp;&amp; b)</b>	<b>a    b</b>	<b>!(a&amp;&amp;b)&amp;&amp;(a  b)</b>

---

## 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.

---

## 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:

