

Esercizio 1.

Un multinsieme è una generalizzazione del concetto di insieme che permette elementi ripetuti. Ad esempio `[[1, 2, 2, 2, 5, 6, 6]]` è un multinsieme dove il numero 2 è ripetuto tre volte.

Si progetti una classe `Multinsieme` con una variabile d'istanza di tipo array di interi che conterrà gli elementi del multinsieme.

Si scrivano un costruttore ed i metodi della classe `Multinsieme` per:

- inserire un nuovo elemento (non è necessario mantenere l'ordinamento tra gli elementi);
- calcolare il numero di elementi contenuti nel multinsieme, tenendo conto delle ripetizioni (esempio: il multinsieme `[[1, 2, 2, 2, 5, 6, 6]]` contiene 7 elementi);
- eliminare una occorrenza di un elemento: il metodo deve ritornare `true` se è stato possibile eliminare l'elemento, `false` altrimenti (esempio: eliminando 2 da `[[1, 2, 2, 2, 5, 6, 6]]` otteniamo `[[1, 2, 2, 5, 6, 6]]` ed il metodo ritorna `true`);
- dato un numero "n", calcolare il numero di occorrenze di "n" (esempio: le occorrenze di 2 in `[[1, 2, 2, 2, 5, 6, 6]]` sono tre).

Infine si progetti una classe di test che costruisce un multinsieme, invoca tutti i metodi, e ne stampa i risultati.

Esercizio 2.

Data l'interfaccia:

```
public interface FormaGeometrica {  
  
    int numeroLati();  
    double calcolaPerimetro();  
}
```

si progettino due classi `Rettangolo` (con due variabili di istanza) e `Quadrato` che implementino l'interfaccia `FormaGeometrica`, in cui `Quadrato` è una sottoclasse di `Rettangolo`.

Esercizio 3.

Si dica cosa stampa il seguente programma, motivando la risposta.

```
public class A {  
    private int n;  
  
    public A() {  
        n=0;  
    }  
  
    public int calcola(int a) {  
        return n+a;  
    }  
}
```

```
public class B extends A {  
    public B() {  
        super();  
    }  
  
    public int calcola(int a) {  
        return super.calcola(a)+1;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        B b = new B();  
        A a = b;  
  
        System.out.println(b.calcola(10));  
        System.out.println(a.calcola(10));  
    }  
}
```