

Appello di Programmazione 2

23 gennaio 2015

Esercizio 1.

Si progettino le classi `Ristorante`, `Tavolo`, `Cameriere` e `Sommelier` per la gestione di un ristorante.

Le classi `Cameriere` e `Sommelier` implementano entrambe l'interfaccia `Personale` così definita:

```
public interface Personale {
    String getNome();
}
```

Ad ogni tavolo sono associati un cameriere ed un sommelier. La classe `Tavolo` ha 3 variabili di istanza: `cameriere`, `sommelier` e `listaOrdini` che tiene traccia degli ordini per il tavolo ed è implementata con un array di `double` di dimensione 100. La classe `Tavolo` ha inoltre i seguenti metodi:

- `void addOrdine(double costo)`: aggiunge un nuovo ordine alla `listaOrdini` del tavolo;
- `double calcolaTotaleTavolo()`: calcola il totale degli ordini presenti (o fatturato) nella `listaOrdini` del tavolo.

La classe `Ristorante` ha 3 variabili di istanza: `nome` (di tipo `stringa`), `listaTavoli` e `listaPersonale`, entrambe implementate con `array list`, ed i seguenti metodi:

- `calcolaTotale(Personale p)`: restituisce il totale fatturato dalla persona `p` calcolato sommando il totale del fatturato per tutti i tavoli a cui è associato;
- `double calcolaTotale()`: calcola il fatturato totale del ristorante, per tutti i tavoli;
- `Tavolo migliorTavolo()`: restituisce il tavolo con il più alto fatturato;
- `Personale migliore()`: restituisce il cameriere o sommelier con il più alto fatturato; il fatturato di un cameriere o di un sommelier è dato dalla somma dei fatturati dei tavoli a cui è associato;
- `int eccedenzaPersonale()`: calcola il numero di camerieri e sommelier che non hanno nessun tavolo associato;
- `int tavoliVuoti()`: calcola il numero di tavoli che non sono stati occupati (il cui fatturato è zero).

Si scriva una classe di test che crea un ristorante con due tavoli, un cameriere ed un sommelier. Si facciano due ordini per tavolo e si richiamino tutti i metodi del ristorante.

Esercizio 2.

Si progetti una nuova classe `SmartTavolo`, simile alle classe `Tavolo`, nella quale si tiene traccia non solo del costo delle consumazioni effettuate, ma anche del loro nome. Ad esempio, ad un tavolo posso essere associate le seguenti consumazioni:

“pasta al pomodoro” 5.00 €

“sformato di verdure” 7.50 €

“tortino al cioccolato” 3.50 €

Si implementi un nuovo metodo che individui la consumazione più costosa ordinata nel tavolo.

Esercizio 3.

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

```
public class A {  
  
    private int a=10;  
  
    int calcola(int n) throws Exception {  
        if (n==0) throw new Exception("ERRORE A");  
        return n+a;  
    }  
}
```

```
public class B extends A {  
  
    int calcola(int n) throws Exception {  
        if (n==1)  
            throw new Exception("ERRORE B");  
        return 2*n+super.calcola(n+1);  
    }  
}
```

```
public class Test {  
  
    public static void main(String[] args) {  
  
        B b = new B();  
        A a = b;  
        try {  
            System.out.println(a.calcola(0));  
            System.out.println(a.calcola(1));  
            System.out.println(b.calcola(0));  
            System.out.println(b.calcola(1));  
        }  
        catch (Exception e) {  
            System.out.println(e.getMessage());  
        }  
        System.out.print("END");  
    }  
}
```