

# Esercizi sulla definizione di classi

## (Fondamenti di Informatica 1 – Walter Didimo)

### Soluzioni

**Esercizio 1** Nella classe definita di seguito indicare quali sono le variabili di istanza, quali sono le variabili statiche e quali sono le variabili locali. Dire inoltre quali sono i costruttori, i metodi di istanza ed i metodi statici e per ciascun metodo dire se è di accesso o di modifica.

```
class Studente {
    private String nome, cognome;
    private int matricola;
    private static int ultimaMatricola = 0;

    public Studente (String nome, String cognome){
        this.nome = nome;
        this.cognome = cognome;
        Studente.ultimaMatricola++;
        this.matricola = Studente.ultimaMatricola;
    }

    public String getNome (){
        return this.nome;
    }

    public String getCognome (){
        return this.cognome;
    }

    public int getMatricola (){
        return this.matricola;
    }

    public String toString (){
        String s = "Nome = " + this.getNome() + ", cognome = " + this.getCognome() +
            ", matricola = " + this.getMatricola();
        return s;
    }

    public static int getProssimaMatricola (){
        return Studente.ultimaMatricola + 1;
    }

    public void setNome (String n){
        this.nome = n;
    }

    public void setCognome (String n){
        this.cognome = n;
    }
}
```

}

## Soluzione

- *Variabili di istanza*: sono le variabili nome, cognome, matricola, dichiarate fuori dai metodi, all'inizio del corpo della classe.
- *Variabili statiche*: soltanto la variabile ultimaMatricola, dichiarata fuori dai metodi, all'inizio del corpo della classe.
- *Variabili locali*: i parametri formali nome e cognome del costruttore, la variabile s dichiarata nel metodo toString, le variabili n parametri formali dei metodi setNome e setCognome.
- *Costruttore*:
  - Studente (String nome, String cognome)
- *Metodi di istanza*:
  - getNome (), setCognome(), getMatricola(), toString(): sono metodi di accesso
  - setNome(String n), setCognome(String n): sono metodi di modifica
- *Metodi statici (o di classe)*:
  - getNextMatricola(): è un metodo di accesso

**Esercizio 2** Dire che tipo di oggetti modella la classe Studente definita nell'Esercizio 1, e cosa fanno i metodi definiti nella classe.

## Soluzione

La classe Studente modella studenti universitari di un qualche ateneo. Ogni istanza della classe rappresenta uno studente universitario con un nome, un cognome, ed un numero di matricola. Ad una istanza della classe Studente si può impostare il proprio nome e cognome all'atto della sua costruzione (tramite il costruttore); il numero di matricola viene assegnato automaticamente come numero progressivo. In qualunque istante del suo ciclo di vita, ad un oggetto Studente si può chiedere di fornirci il suo nome (metodo getNome()), il suo cognome (metodo setCognome()), la sua matricola (metodo getMatricola()) o una sua descrizione completa (metodo toString()). Ad un oggetto Studente posso reimpostare il proprio nome o cognome, attraverso i metodi (setNome(String n) e setCognome(setCognome(String n))). Il metodo getNextMatricola() è invece un metodo statico che mi permette di chiedere alla classe Studente di indicarmi quale sarà il prossimo numero di matricola che verrà utilizzato.

**Esercizio 3** Scrivere una classe ProvaStudente che verifica il corretto funzionamento della classe Studente definita nell'Esercizio 1. La classe ProvaStudente avrà il solo metodo speciale main, il quale svolgerà le seguenti azioni:

- Fa inserire all'utente nome e cognome di due studenti e crea i rispettivi oggetti Studente.
- Visualizza all'utente una descrizione completa dei due studenti inseriti, utilizzando il metodo toString della classe Studente.
- Visualizza all'utente il numero di matricola dell'eventuale prossimo studente.
- Chiede all'utente di ridefinire i dati (nome e cognome) del primo studente inserito, ed aggiorna i dati del corrispondente oggetto Studente.
- Visualizza all'utente una descrizione completa del primo studente, al fine di mostrare come i dati dell'oggetto sono effettivamente cambiati.

## Soluzione

```
class ProvaStudente {

    public static void main (String[] args){

        /* Fa inserire all'utente due studenti */

        InputWindow in = new InputWindow ();
        String nome;
        String cognome;
        nome = in.readString ("Nome del primo studente");
        cognome = in.readString ("Cognome del primo studente");
        Studente stud1 = new Studente (nome, cognome);
        nome = in.readString ("Nome del secondo studente");
        cognome = in.readString ("Cognome del secondo studente");
        Studente stud2 = new Studente (nome, cognome);

        /* Visualizza i dati completi dei due studenti inseriti */

        OutputWindow out = new OutputWindow ("Output di ProvaStudenti");
        out.writeln ("Dati del primo studente:");
        out.writeln (stud1.toString ());
        out.writeln ();
        out.writeln ("Dati del secondo studente:");
        out.writeln (stud2.toString ());
        out.writeln ();

        /* Visualizza il numero della prossima matricola */
        int mat = Studente.getProssimaMatricola();
        out.writeln ("Prossima Matricola = " + mat);

        /* Chiede all'utente di cambiare i dati del primo studente e
           reimposta i dati */
        nome = in.readString ("Nuovo nome per il primo studente ");
        cognome = in.readString ("Nuovo cognome per il primo studente ");
        stud1.setNome (nome);
        stud1.setCognome (cognome);

        /* Visualizza di nuovo la descrizione completa del primo studente */
        out.writeln ("Nuovi dati per il primo studente:");
        out.writeln (stud1.toString ());

    }

}
```

**Esercizio 4** Definire una classe di nome ContoCorrente, le cui istanze modellano conti correnti bancari. Precisamente, un oggetto ContoCorrente deve avere i seguenti campi:

- nome dell'intestatario del conto (di tipo String),
- cognome dell'intestatario del conto (di tipo String),
- numero di conto (di tipo int),
- ammontare del conto in euro (di tipo double).

La classe ContoCorrente ha inoltre una variabile statica, di nome massimoScoperto, che indica (in valore assoluto) il massimo valore di scoperto consentito per ogni conto corrente (cioè di quanto al più l'ammontare di un qualunque conto corrente può essere negativo).

La classe deve avere i due costruttori seguenti:

- /\* crea un oggetto ContoCorrente con intestatario e numero di conto specificati; l'ammontare iniziale è fissato a 0 dal costruttore \*/  
public ContoCorrente (String nome, String cognome, int num)
- /\* crea un oggetto ContoCorrente con intestatario, numero di conto e ammontare iniziale specificati \*/  
public ContoCorrente (String nome, String cognome, int num, double ammontare)

La classe deve inoltre avere i seguenti metodi di istanza:

- /\* incrementa (o decrementa) il valore del conto corrente con l'ammontare specificato \*/  
public void aggiungi (double somma)
- /\* restituisce true se l'ammontare corrente è al di sotto dello scoperto massimo consentito e false altrimenti \*/  
public boolean isScoperto ( )
- /\* restituisce una descrizione completa del conto corrente \*/  
public String toString ( )

La classe ha infine il seguente metodo statico:

- /\* modifica il valore del massimo scoperto con la nuova somma indicata \*/  
public static void setMassimoScoperto (double somma)

Una volta definita la classe ContoCorrente, scrivere una classe ProvaContoCorrente che ne verifica il corretto funzionamento. La classe ProvaContoCorrente avrà il solo metodo main, che svolge le seguenti azioni:

- Fissa lo scoperto massimo per i conti correnti alla somma di 1.000,00 Euro.
- Crea un primo oggetto ContoCorrente facendo impostare all'utente tutti i dati ad esso relativo, compreso l'ammontare iniziale.
- Visualizza all'utente una descrizione completa del conto creato.
- Chiede all'utente di specificare una somma che si vuol aggiungere al conto corrente creato e modifica l'oggetto corrispondente.
- Visualizza all'utente la nuova descrizione del conto corrente.
- Visualizza poi all'utente il messaggio true se il conto è scoperto e false altrimenti.

Dire infine se la classe ProvaContoCorrente ha effettivamente testato tutti i metodi della classe ContoCorrente.

## Soluzione

```
class ContoCorrente{
    private String nome, cognome;
    private int numero;
    private double ammontare;
    private static double massimoScoperto;

    public ContoCorrente (String nome, String cognome, int num){
        this.nome = nome;
        this.cognome = cognome;
        this.numero = num;
        this.ammontare = 0;
    }

    public ContoCorrente (String nome, String cognome, int num, double
ammontare){
        this.nome = nome;
        this.cognome = cognome;
        this.numero = num;
        this.ammontare = ammontare;
    }

    public void aggiungi (double somma){
        this.ammontare += somma;
    }

    public boolean isScoperto (){
        return (this.ammontare + ContoCorrente.massimoScoperto < 0);
    }

    public String toString (){
        String s = "Conto corrente numero: " + this.numero + "\n";
        s += "Cognome: " + this.cognome + "\n";
        s += "Nome: " + this.nome + "\n";
        s += "Ammontare: " + this.ammontare;
        return s;
    }

    public static void setMassimoScoperto (double somma){
        ContoCorrente.massimoScoperto = somma;
    }
}
```

---

```

class ProvaContoCorrente{

    public static void main (String[] args){

        /* Fissa lo scoperto massimo per i conti correnti alla somma 1000
           Euro */
        ContoCorrente.setMassimoScoperto(1000);

        /* Crea un primo oggetto ContoCorrente facendo impostare all'utente
           tutti i dati ad esso relativo, compreso l'ammontare iniziale */
        InputWindow in = new InputWindow ();
        String nome = in.readString ("Nome dell'intestatario del conto");
        String cognome = in.readString ("Cognome intestatario del conto");
        int numero = in.readInt ("Numero del conto");
        double ammontare = in.readDouble ("Ammontare iniziale");
        ContoCorrente conto;
        conto = new ContoCorrente (nome, cognome, numero, ammontare);

        /* Visualizza una descrizione completa del conto creato */
        OutputWindow out = new OutputWindow ();
        out.writeln ("Descrizione del conto corrente");
        out.writeln (conto.toString ());

        /* Chiede di specificare una somma che si vuol aggiungere al conto
           corrente creato e modifica l'oggetto corrispondente*/
        double somma = in.readDouble ("Somma da aggiungere/sottrarre");
        conto.aggiungi (somma);

        /* Visualizza all'utente la nuova descrizione del conto corrente */
        out.writeln ("Descrizione aggiornata del conto corrente");
        out.writeln (conto.toString ());

        /* Visualizza all'utente il messaggio true se il conto è scoperto e
           false altrimenti*/
        out.writeln ("Conto scoperto? " + conto.isScoperto());
    }
}

```