

Sia data la classe `StudUniv` per gli studenti universitari (definita in una lezione precedente). Scrivere una nuova classe `Corso`, dove ciascun corso è caratterizzato da nome del corso, nome del docente titolare del corso, numero di crediti associati al corso, settore scientifico disciplinare del corso e da un elenco di studenti universitari, il cui numero massimo deve essere impostato dal costruttore della classe. Definire i seguenti metodi per la classe `Corso`:

1. i metodi che restituiscono i valori delle variabili istanza;
2. un metodo che aggiunge uno studente al corso;
3. un metodo che restituisce l'elenco delle matricole di tutti gli studenti del corso;
4. un metodo che, data una matricola  $k$ , restituisce l'elenco degli studenti del corso la cui matricola è maggiore di  $k$ ;
5. un metodo che, date due stringhe che denotano un nome ed un cognome, restituisce *true* se nell'elenco degli studenti del corso compare uno studente con nome e cognome dati, altrimenti il metodo restituisce *false*.

La prima soluzione presentata utilizza una variabile istanza `cont`, inizializzata a 0, che denota la prima posizione “libera” nell'array che rappresenta l'elenco di studenti in un oggetto della classe `Corso`.

```
public class Corso {

    private String nomeCorso, nomeDocente, ssid;
    private int cfu;
    private StudUniv[] studenti;
    private int cont = 0;

    public Corso (String nc, String nd, int c, String ssid, int m) {
        this.nomeCorso = nc;
        this.nomeDocente = nd;
        this.cfu = c;
        this.ssid = ssid;
        this.studenti = new StudUniv[m];
    }

    public String leggiNomeCorso() {
        return this.nomeCorso;
    }

    public String leggiNomeDocente() {
        return this.nomeDocente;
    }

    public int leggiNumCfu() {
        return this.cfu;
    }

    public String leggiSSID() {
        return this.ssid;
    }
}
```

```
public StudUniv[] leggiElencoStudenti() {  
    return this.studenti;  
}
```

// DISCUTERE...

```
public void aggStudente (StudUniv s) {  
    if (this.cont < this.studenti.length) {  
        this.studenti[cont] = s;  
        this.cont++;  
    }  
}
```

```
public int[] elencoMatricole() {  
    int[] a = new int[cont];  
    for (int i = 0; i < cont; i++) {  
        a[i] = this.studenti[i].leggiMatricola();  
    }  
    return a;  
}
```

```
public StudUniv[] elencoStudenti (int k) {  
    int c = 0;  
    for (int i = 0; i < cont; i++) {  
        if (this.studenti[i].leggiMatricola() > k)  
            c++;  
    }  
    StudUniv[] a = new StudUniv[c];  
    int j = 0;  
    for (int i = 0; i < cont; i++) {  
        if (this.studenti[i].leggiMatricola() > k) {  
            a[j] = this.studenti[i];  
            j++;  
        }  
    }  
    return a;  
}
```

```
public boolean esisteStudente (String nome, String cognome) {  
    for (int i = 0; i < cont; i++) {  
        if (this.studenti[i].leggiNome().equals(nome) &&  
            this.studenti[i].leggiCognome().equals(cognome))  
            return true;  
    }  
    return false;  
}
```

```
}
```

**Osservazione** Sia la specifica dell'ultimo metodo definito nella classe **Corso** modificata come segue: “definire un metodo che, dato uno studente universitario *s*, restituisce *true* se *s* compare nell'elenco degli studenti del corso, altrimenti il metodo restituisce *false*”.

In tal caso, ~~può risultare~~ utile definire un metodo **equals** nella classe **StudUniv** che confronta due studenti e restituisce *true* se tutti i campi dei due oggetti sono uguali, altrimenti

restituisce *false*. Quindi, se aggiungiamo nella classe StudUniv il seguente metodo

```
public boolean equals (StudUniv s) { //N.B. OVERLOADING
if return (this.nome.equals(s.nome) && this.cognome.equals(s.cognome) &&
    this.matricola == s.matricola) ;
return true;
else
return false;
}
```

il metodo nella classe Corso che controlla se un dato studente compare nell'elenco degli studenti del corso può essere definito come segue:

```
public boolean esisteStudente (StudUniv s) {
    for (int i = 0; i < cont; i++) {
        if (this.studenti[i].equals(s))
            return true;
    }
    return false;
}
```

Scrivere un breve programma di prova che illustri l'uso della classe Corso e dei suoi metodi.

```
public class CorsoTest {
    public static void main(String[] args) {

        Corso c = new Corso("Lab.Progr.", "Nesi", 6, "INF/01", 100);
        StudUniv s1 = new StudUniv("Andrea", "Rossi", 134678);
        StudUniv s2 = new StudUniv("Elena", "Bianchi", 149987);
        StudUniv s3 = new StudUniv("Paolo", "Neri", 150456);

        c.aggStudente(s1);
        c.aggStudente(s2);
        c.aggStudente(s3);

        int[] m = c.elencoMatricole();
        for (int i=0; i<m.length; i++)
            System.out.println(m[i]);

        int matr = 149500;
        StudUniv[] a = c.elencoStudenti(matr);
        System.out.println("Gli studenti del corso " + c.leggiNomeCorso() +
            " con matricola > " + matr + " sono:");
        for (int i=0; i<a.length; i++)
            System.out.println(a[i].leggiNome() + " " + a[i].leggiCognome());

        if (c.esisteStudente(s2))
            System.out.println(s2.leggiNome() + " " + s2.leggiCognome() +
                " segue il corso " + c.leggiNomeCorso());
    }
}
```

Il risultato di tale programma di prova è la seguente stampa:

134678

149987

150456

Gli studenti del corso Lab.Progr. con matricola > 149500 sono:

Elena Bianchi

Paolo Neri

Elena Bianchi segue il corso Lab.Progr.

Una soluzione alternativa per la definizione della classe Corso, in cui non viene utilizzata la variabile istanza cont, è la seguente:

```
public class CorsoN {

    private String nomeCorso, nomeDocente, ssd;
    private int cfu;
    private StudUniv[] studenti;

    public CorsoN (String nc, String nd, int c, String ssd, int m) {
        this.nomeCorso = nc;
        this.nomeDocente = nd;
        this.cfu = c;
        this.ssd = ssd;
        this.studenti = new StudUniv[m];
    }

    public String leggiNomeCorso() {
        return this.nomeCorso;
    }

    public String leggiNomeDocente() {
        return this.nomeDocente;
    }

    public int leggiNumCfu() {
        return this.cfu;
    }

    public String leggiSSD() {
        return this.ssd;
    }

    public StudUniv[] leggiElencoStudenti() {
        return this.studenti;
    }

    public void aggStudente (StudUniv s) {
        int i = 0;
        while (i < this.studenti.length) {
            if (this.studenti[i] != null)
                i++;
            else {
```

```

        this.studenti[i] = s;
        return;
    }
}

public int[] elencoMatricole() {
    int c = 0;
    for (int i = 0; i < this.studenti.length; i++) {
        if (this.studenti[i] != null)
            c++;
    }
    int[] a = new int[c];
    int j = 0;
    for (int i = 0; i < this.studenti.length; i++) {
        if (this.studenti[i] != null) {
            a[j] = this.studenti[i].leggiMatricola();
            j++;
        }
    }
    return a;
}

public StudUniv[] elencoStudenti (int k) {
    int c = 0;
    for (int i = 0; i < this.studenti.length; i++) {
        if (this.studenti[i] != null && this.studenti[i].leggiMatricola() > k)
            c++;
    }
    StudUniv[] a = new StudUniv[c];
    int j = 0;
    for (int i = 0; i < this.studenti.length; i++) {
        if (this.studenti[i] != null && this.studenti[i].leggiMatricola() > k) {
            a[j] = this.studenti[i];
            j++;
        }
    }
    return a;
}

public boolean esisteStudente (StudUniv s) {
    int i = 0;
    while (i < this.studenti.length) {
        if (this.studenti[i] != null && this.studenti[i].equals(s))
            return true;
        else {
            i++;
        }
    }
    return false;
}
}

```