

Esercizi su Array

(Fondamenti di Informatica – Emilio Di Giacomo)

Esercizio 1 Le istanze della classe **SequenzaDiParole** rappresentano sequenze qualsiasi di parole. La classe ha il seguente scheletro.

```
public class SequenzaDiParole {  
  
    private String[] seq; // memorizza le parole della sequenza  
  
    /* Crea un oggetto SequenzaDiParole. La sequenza è composta dalle stringhe che  
    compaiono nell'array seq. */  
    public SequenzaDiParole (String[] seq){...}  
  
    /* Crea un oggetto SequenzaDiParole. La sequenza è composta dalle parole che  
    compaiono nella stringa testo (le parole sono separate da uno o più spazi). Per  
    ottenere le diverse parole che compongono il testo si consiglia di utilizzare il metodo  
    String[] split(String sep) della classe String che consente di spezzare una stringa  
    in sottostringhe separate dal separatore sep e le restituisce sotto forma di array di  
    stringhe. */  
    public SequenzaDiParole (String testo){...}  
  
    /* Restituisce un oggetto di tipo SequenzaDiParole la cui sequenza contiene le  
    parole della sequenza rappresentata dall'oggetto ricevente che contengono la  
    stringa s come sottostringa */  
    public SequenzaDiParole paroleCheContengono(String s){...}  
  
    /* Restituisce una stringa in cui vengono mostrate (una per riga) tutte le parole della  
    sequenza. */  
    public String toString(){...}  
}
```

Dopo aver scritto la classe **SequenzaDiParole** si utilizzi la classe **ProvaSequenzaDiParole** (che è fornita già implementata) per testarla.

Esercizio 2 Un *sistema lineare* è un insieme di m equazioni lineari in n incognite; un sistema lineare può essere genericamente scritto come segue:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \quad \quad \quad \dots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

I termini a_{ij} sono numeri reali chiamati *coefficienti*, i termini b_i sono numeri reali e sono detti *termini noti*, i termini x_j sono le incognite del sistema. Un sistema lineare può essere descritto mediante una matrice A ed un vettore b ; la matrice A , detta matrice dei coefficienti, è la matrice

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

Il vettore b , detto vettore dei termini noti, è il vettore:

$$b = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_m \end{pmatrix}$$

Una soluzione di un'equazione del sistema è un vettore di numeri reali $(\overline{x_1}, \overline{x_2}, \dots, \overline{x_n})$ che sostituiti alle incognite rendono vera l'equazione. Una soluzione del sistema è un vettore di numeri reali $(\overline{x_1}, \overline{x_2}, \dots, \overline{x_n})$ che sostituiti alle incognite rendono vere tutte le equazioni del sistema.

Ogni oggetto della classe **Sistema** rappresenta un sistema lineare di m equazioni lineari in n incognite. La classe ha il seguente scheletro:

```
public class Sistema {

    private double[][] A; // matrice dei coefficienti
    private double[] b;   // vettore dei termini noti

    /* crea un sistema lineare la cui matrice dei coefficienti è A e il cui vettore dei termini
    noti è b. Si può assumere che A e b abbiano dimensioni coerenti (cioè il numero di
    elementi di b sia pari al numero di righe di A) */
    public Sistema(double[][] A, double[] b){...}

    /* Restituisce true se il vettore x passato come parametro è una soluzione della
    equazione i-esima del sistema. Si può assumere che la dimensione di x sia corretta
    (cioè sia pari al numero di colonne della matrice dei coefficienti) */
    public boolean verificaEquazione(int i, double[] x){...}

    /* Restituisce true se il vettore x passato come parametro è una soluzione del
    sistema. Si può assumere che la dimensione di x sia corretta (cioè sia pari al
    numero di colonne della matrice dei coefficienti) */
    public boolean verificaSistema(double[] x){...}

    /* Restituisce sotto forma di stringa l'equazione i-esima del sistema */
    public String stampaEquazione(int i){...}

    /* Restituisce sotto forma di stringa il sistema */
    public String toString(){...}
}
```

Si scriva la classe **Sistema** ed una classe **ProvaSistema** che contiene il solo metodo **main** e che esegue le seguenti azioni:

- fa inserire all'utente un sistema lineare chiedendogli di fornire sia la matrice dei coefficienti A sia il vettore dei termini noti b (è necessario garantire che A e b abbiano dimensioni coerenti);
- crea un oggetto Sistema s che rappresenta il sistema lineare inserito dall'utente;
- visualizza il sistema creato;
- fa inserire all'utente un vettore x che rappresenta una possibile soluzione del sistema (è necessario garantire che la dimensione del vettore inserito sia corretta);
- Dice all'utente se x è effettivamente una soluzione del sistema, oppure no.

Esercizio 3 Ogni oggetto della classe **Tartaglia** rappresenta un triangolo di Tartaglia con un certo numero di righe. Si ricorda che il triangolo di Tartaglia è una disposizione geometrica a forma di triangolo dei coefficienti binomiali, ossia dei coefficienti dello sviluppo del binomio $(a+b)$ elevato ad una qualsiasi potenza n . Il triangolo di Tartaglia può essere così definito:

- l' i -esima riga ha $i+1$ elementi ($i=0,1,2,\dots$)
- l'elemento t_{ij} in posizione j -esima sulla riga i -esima ($0 \leq j \leq i$) è definito come segue:

$$t_{ij} = \begin{cases} 1 & \text{se } j=0 \text{ oppure } j=i \\ t_{i-1,j-1} + t_{i-1,j} & \text{altrimenti} \end{cases}$$

Le prime cinque righe del triangolo di Tartaglia sono le seguenti:

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1

```

Si scriva la classe **Tartaglia** che ha il seguente scheletro:

```

class Tartaglia{

    private int[][] triangolo;

    /* costruttore. Costruisce un triangolo di Tartaglia con n righe */
    public Tartaglia(int n){...}

    /* restituisce la riga i-esima del triangolo rappresentato */
    public int[] getRiga(int i){...}

    /* restituisce una descrizione del triangolo rappresentato
       sotto forma di stringa */
    public String toString(){..}

}

```

Si scriva poi una classe di test **ProvaTartaglia** che testa la precedente.