

**Corso di Laurea in Ingegneria Informatica e Automatica**  
**A. A. 2018/2019**

# **Progettazione del Software**

**Giuseppe De Giacomo, Paolo Liberatore,  
Massimo Mecella**

## **Esercitazione 3**

# Esercizio E3\_1

Scrivere i seguenti metodi statici Java:

- *generaMatrice*: che, dati due valori interi  $n$  e  $m$ , genera una matrice di dimensione  $n \times m$  contenente valori double random compresi tra 0 e 100
- *stampaMatrice*: che, data una matrice di double, stampa la matrice per righe (senza usare `Arrays.toString()`)
- *stampaArray*: che, dato un array di double, lo stampa (senza usare `Arrays.toString()`)
- *diagonale*: che, data una matrice di double, restituisce un array di double contenente gli elementi della diagonale principale della matrice
  - La diagonale di una matrice è la diagonale della matrice quadrata più grande contenuta nella matrice
- *sommaArray*: che, dato un array di double, restituisce la somma delle sue componenti

Scrivere un programma Java che legga in input due valori interi  $n$  e  $m$ , generi una matrice di double con valori random, calcoli e stampi il vettore della diagonale principale e la somma degli elementi della diagonale principale.

# Esercizio E3\_2

Scrivere una classe contenente un metodo statico Java *differenzaVicini* che, data in ingresso una matrice di valori double, restituisca una matrice ottenuta nel seguente modo:

ogni cella  $[r, c]$  della matrice di output contiene la differenza tra il valore della cella  $[r, c]$  della matrice di input e la somma dei valori dei suoi vicini.

I vicini della cella  $[r, c]$  sono le celle  $[r-1, c-1]$ ,  $[r-1, c]$ ,  $[r-1, c+1]$ ,  $[r, c-1]$ ,  $[r, c+1]$ ,  $[r+1, c-1]$ ,  $[r+1, c]$  e  $[r+1, c+1]$ . Se una cella si trova sul bordo, i vicini di una cella non includono gli elementi fuori della matrice.

## Esempio

•Data in input la matrice

```
3 4 3 3 4 2
4 4 1 0 -3 3
3 -4 1 -2 1 -2
```

si ottiene in output la matrice

```
-9 -11 -9 -2 -1 -2
-6 -11 -8 -8 -12 1
-1 -17 2 -2 5 -3
```

Per la verifica, si utilizzi il metodo *stampaMatrice* dell'esercizio precedente, senza riscriverlo

# Esercizio E3\_3

Si consideri la classe contenente il metodo statico *differenziaVicini* sviluppato nell'esercizio E3\_2. Scrivere un programma Java che legga console una matrice di double (vedere sotto per i dettagli), applichi su tale matrice il metodo *differenziaVicini* e stampi quindi la matrice ottenuta.

L'input da console è strutturato come segue: la prima riga contiene due interi, separati da «,» che indicano le righe e le colonne della matrice, dalla seconda riga in poi, la matrice vera e propria con i valori indicati per righe e separati da «,».

Esempio:

```
2, 5  
1.2, 2.0, 3.9, 4.3, 5.5  
6.4, 7.0, 8.4, 9.7, 0.1
```

La prima riga indica che  
seguirà una matrice 2x5

Esercizio E3\_4

# Scuole elementari

Realizzare le classi necessarie ad una applicazione per elaborare le informazioni su scuole elementari, insegnanti e classi. Di ogni scuola elementare interessa il nome, l'indirizzo e il provveditorato di appartenenza. Degli insegnanti interessa il nome, il cognome, l'anno di vincita del concorso, la scuola elementare di cui sono dipendenti. Ogni classe è caratterizzata dalla scuola a cui appartiene, dal nome (ad es. "IV A"), dagli insegnanti e dal numero di alunni. Ogni insegnante afferisce ad una sola scuola e ogni classe ha esattamente tre insegnanti, ordinati. I nomi delle scuole sono univoci. Nome e cognome degli insegnanti sono univoci. Il nome di una classe è univoco all'interno di ciascuna scuola.

# Interfaccia delle classi

## Scuola

```
// proprietà  
nome (immutabile)  
indirizzo (immutabile)  
provveditorato (immutabile)  
// operazioni  
costruttore  
get/set  
// servizio  
stampa
```

# Interfaccia delle classi

## Insegnante

```
// proprietà  
nome (immutabile)  
cognome (immutabile)  
annoVincita (immutabile)  
scuola  
// operazioni  
costruttore  
get/set  
// servizio  
stampa
```

## Classe

```
// proprietà  
scuola (immutabile)  
nome (immutabile)  
insegnanti  
numAlunni  
// operazioni  
costruttore  
get/set  
inserisciInsegnante(int posizione,  
Insegnante i)  
// servizio  
stampa
```