

Programmazione orientata agli oggetti

Luca Iocchi, Massimo Mecella
Daniele Sora

Esercitazione 2



Esercizio E2_1

Scrivere una classe Java contenente i seguenti metodi statici **ricorsivi**:

- *inverti*, che, data una stringa, restituisca una nuova stringa contenente tutti i caratteri della stringa in input in ordine inverso,
- *concatena*, che, date due stringhe A e B, restituisca una nuova stringa contenente la concatenazione di A e B.

Scrivere un programma Java che legga in input due stringhe e stampi il risultato della concatenazione della prima stringa con l'inversa della seconda, invocando i metodi suddetti.

Nota: non usare funzioni Java che rendono l'esercizio inutile.



Esercizio E2_2

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, stampa la matrice per righe
- *stampaArray*: che, dato un array, lo stampi
- *diagonale*: che, data una matrice, restituisce un array contenente gli elementi della diagonale principale della 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 E2_3

Scrivere una classe contenente i metodi statici della classe dell'esercizio precedente e un metodo 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
```



Esercizio E2_4

Si consideri la classe contenente il metodo statico *differenziaVicini* sviluppato nell'esercizio 2_3. Scrivere un programma Java che legga dal file input4.txt una matrice di double, applichi su tale matrice il metodo *differenziaVicini* e scriva quindi su file output4.txt la matrice ottenuta.

Il file input4.txt è strutturato come segue: la prima riga contiene due interi 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.

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

