



Laborator 7

Tablouri in limbajul Java - Tablouri bidimensionale (matrici)

Probleme rezolvate:

1. Sa se citeasca elemente de tip intreg si sa se stocheze intr-un tablou bidimensional, iar apoi sa se afiseze aceste elemente.

Codul sursa al programului cu numele **matrice.java** este urmatorul:

```
import java.util.Scanner;
public class lab_7_1 {
    public static void main(String[] args){
        /* declararea si citirea variabilelor de memorie in care se vor retine numarul de linii,
        respectiv numarul de coloane ale tabloului bidimensional */
        int n, m;
        Scanner input = new Scanner(System.in);
        n = input.nextInt();
        System.out.println("Numarul de linii = "+n);
        m = input.nextInt();
        System.out.println("Numarul de coloane = "+m);

        /* declararea variabilei de tip tablou bidimensional A[][] */
        int A[ ][ ]= new int[n][m];
        // citirea elementelor in tabloul dat
        int i,j;
        for(i=0;i<n;i++)
            for(j=0;j<m;j++)
                A[i][j] = input.nextInt();

        // afisarea elementelor tabloului
        System.out.println ("Matricea citita este: ");
        for(i=0;i<n;i++){
            for(j=0;j<m;j++)
                System.out.print(" "+A[i][j]);
            System.out.println();
        }
    }
}
```

```
1 import java.util.Scanner;
2 public class lab_7_1 {
3     public static void main(String[] args){
4         /* declararea si citirea variabilelor de memorie in care se vor retine numarul de linii,
5            */
6             int n, m;
7             Scanner input = new Scanner(System.in);
8             n = input.nextInt();
9             System.out.println("Numarul de linii = "+n);
10            m = input.nextInt();
11            System.out.println("Numarul de coloane = "+m);
12
13            /* declararea variabilei de tip tablou bidimensional A[][] */
14            int A[ ][ ]= new int[n][m];
15            // citirea elementelor in tabloul dat
16            int i,j;
17            for(i=0;i<n;i++)
18                for(j=0;j<m;j++)
19                    A[i][j] = input.nextInt();
20
21            // afisarea elementelor tabloului
22            System.out.println ("Matricea citita este: ");
23            for(i=0;i<n;i++){
24                for(j=0;j<m;j++)
25                    System.out.print(" "+A[i][j]);
26                System.out.println();
27            }
28        }
29    }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1 Interactive

Stdin Inputs

```
2 3
1 2 3
4 5 6
```

CommandLine Arguments

Execute   

Result

CPU Time: 0.21 sec(s), Memory: 37604 kilobyte(s) comp

```
Numarul de linii = 2
Numarul de coloane = 3
Matricea citita este:
1 2 3
4 5 6
```

2. Programul urmator (**loto.java**) ilustreaza modul de folosire al sirurilor in Java. In jocul de loterie, se selecteaza saptamanal sase numere de la 1 la 49. Programul alege aleator numere pentru 1000 de jocuri si afiseaza apoi de cate ori a aparut fiecare numar in cele 1000 de jocuri.

```
import java.util.Scanner;
import java.util.Random;
public class lab_7_2 {
    public static void main(String[] args){

        int numere[] = new int[50];
        for(int i = 0;i < numere.length; ++i)
            numere[i] = 0;
        for(int i = 0; i < 1000; ++i)
            for(int j = 0 ; j < 6; ++j)
                // genereaza numere de loterie intre 1 si 49
                numere[(int) (Math.random() * 49) + 1]++;
        // numara aparitiile de numere de joc
        for(int k = 1; k <= 49; ++k)
            System.out.println(k + ": "+numere[k]);
    }
}
```

Result

CPU Time: 0.16 sec(s), Memory: 33864 kilobyte(s)

```
1: 136
2: 127
3: 118
4: 132
5: 112
6: 122
7: 112
8: 133
9: 120
10: 133
11: 146
12: 125
13: 122
14: 130
15: 135
16: 110
17: 125
18: 121
19: 125
20: 117
21: 121
22: 112
23: 114
24: 116
25: 124
26: 100
27: 128
28: 116
29: 109
30: 122
31: 116
32: 115
33: 120
34: 110
35: 114
36: 129
37: 123
38: 135
39: 124
40: 130
41: 110
42: 127
43: 132
44: 127
45: 138
46: 111
47: 128
48: 117
49: 131
```

```
1 import java.util.Scanner;
2 import java.util.Random;
3 public class lab_7_2 {
4     public static void main(String[] args){
5
6         int numere[] = new int[50];
7         for(int i = 0;i < numere.length; ++i)
8             numere[i] = 0;
9         for(int i = 0; i < 1000; ++i)
10            for(int j = 0 ; j < 6; ++j)
11                // genereaza numere de loterie intre 1 si 49
12                numere[(int) (Math.random() * 49) + 1]++;
13        // numara aparitiile de numere de joc
14        for(int k = 1; k <= 49; ++k)
15            System.out.println(k + ": "+numere[k]);
16    }
17 }
18
19
```

3. Se consideră o matrice A cu $n \times m$ numere întregi. Se cere să se obțină **transpusa** sa.

```
import java.util.Scanner;
public class lab_7_3 {
    public static void main(String[] args){
        /* declararea si citirea variabilelor de memorie in care se vor retine numarul de
linii,
        respectiv numarul de coloane ale tabloului bidimensional */
        int n, m;
        Scanner input = new Scanner(System.in);
        n = input.nextInt();
        System.out.println("Numarul de linii = "+n);
        m = input.nextInt();
        System.out.println("Numarul de coloane = "+m);

        /* declararea variabilei de tip tablou bidimensional A[][] */
        int A[ ][ ] = new int[n][m];
        // citirea elementelor in tabloul dat
        int i,j;
        for(i=0;i<n;i++)
            for(j=0;j<m;j++)
                A[i][j] = input.nextInt();

        // afisarea elementelor tabloului bidimensional (matrice)
        System.out.println ("Matricea citita este: ");
        for(i=0;i<n;i++){
            for(j=0;j<m;j++)
                System.out.print(" "+A[i][j]);
            System.out.println();
        }

        int B[ ][ ] = new int[m][n];
        for(i=0;i<n;i++)
            for(j=0;j<m;j++)
                B[j][i] = A[i][j];

        // afisarea elementelor matricei transpuse
        System.out.println ("Matricea transpusa este: ");
        for(i=0;i<m;i++){
            for(j=0;j<n;j++)
                System.out.print(" "+B[i][j]);
            System.out.println();
        }
    }
}
```

```

1 import java.util.Scanner;
2 public class lab_7_3 {
3     public static void main(String[] args){
4         /* declararea si citirea variabilelor de memorie in care se vor retine numarul de linii,
5         respectiv numarul de coloane ale tabloului bidimensional */
6         int n, m;
7         Scanner input = new Scanner(System.in);
8         n = input.nextInt();
9         System.out.println("Numarul de linii = "+n);
10        m = input.nextInt();
11        System.out.println("Numarul de coloane = "+m);
12
13        /* declararea variabilei de tip tablou bidimensional A[][] */
14        int A[ ][ ] = new int[n][m];
15        // citirea elementelor in tabloul dat
16        int i,j;
17        for(i=0;i<n;i++)
18            for(j=0;j<m;j++)
19                A[i][j] = input.nextInt();
20
21        // afisarea elementelor tabloului bidimensional (matrice)
22        System.out.println ("Matricea citita este: ");
23        for(i=0;i<n;i++){
24            for(j=0;j<m;j++)
25                System.out.print(" "+A[i][j]);
26            System.out.println();
27        }
28
29        int B[ ][ ] = new int[m][n];
30        for(i=0;i<n;i++)
31            for(j=0;j<m;j++)
32                B[j][i] = A[i][j];
33
34        // afisarea elementelor matricei transpuse
35        System.out.println ("Matricea transpusa este: ");
36        for(i=0;i<m;i++){
37            for(j=0;j<n;j++)
38                System.out.print(" "+B[i][j]);
39            System.out.println();
40        }
41    }
42 }
43
44

```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1



Interactive

Stdin Inputs

```

3 4
1 2 3 4
5 6 7 8
9 10 11 12

```

CommandLine Arguments

Result

CPU Time: 0.27 sec(s), Memory: 37280 kilobyte(s)

```

Numarul de linii = 3
Numarul de coloane = 4
Matricea citita este:
1 2 3 4
5 6 7 8
9 10 11 12
Matricea transpusa este:
1 5 9
2 6 10
3 7 11
4 8 12

```

4. Se considera o matrice $A_{n \times m}$ ($1 \leq n, m \leq 30$) avand componente numere intregi. Sa se memoreze intr-un vector V sumele elementelor de pe fiecare linie a matricii ($V[i]$ va reprezenta suma elementelor de pe linia i in matrice.)

```
import java.util.Scanner;
public class lab_7_4 {
    public static void main(String[] args){
        /* declararea si citirea variabilelor de memorie in care se vor retine numarul de
linii,
        respectiv numarul de coloane ale tabloului bidimensional */
        int n, m;
        Scanner input = new Scanner(System.in);
        n = input.nextInt();
        System.out.println("Numarul de linii = "+n);
        m = input.nextInt();
        System.out.println("Numarul de coloane = "+m);

        /* declararea variabilei de tip tablou bidimensional A[][] */
        int A[ ][ ] = new int[n][m];
        // citirea elementelor in tabloul dat
        int i,j;
        for(i=0;i<n;i++)
            for(j=0;j<m;j++)
                A[i][j] = input.nextInt();

        // afisarea elementelor tabloului bidimensional (matrice)
        System.out.println ("Matricea citita este: ");
        for(i=0;i<n;i++){
            for(j=0;j<m;j++)
                System.out.print(" "+A[i][j]);
            System.out.println();
        }

        int V[ ] = new int[m], suma;
        for(i = 0; i < n; i++){
            suma = 0;
            for(j = 0; j < m; j++)
                suma += A[i][j];
            V[i] = suma;
        }

        System.out.println ("Vectorul suma este: ");
        for(i=0;i<n;i++)
            System.out.print(" "+V[i]);
        }
}
```




```
1 import java.util.Scanner;
2 public class lab_7_4 {
3     public static void main(String[] args){
4         /* declararea si citirea variabilelor de memorie in care se vor retine numarul de linii,
5         respectiv numarul de coloane ale tabloului bidimensional */
6         int n, m;
7         Scanner input = new Scanner(System.in);
8         n = input.nextInt();
9         System.out.println("Numarul de linii = "+n);
10        m = input.nextInt();
11        System.out.println("Numarul de coloane = "+m);
12
13        /* declararea variabilei de tip tablou bidimensional A[][] */
14        int A[ ][ ]= new int[n][m];
15        // citirea elementelor in tabloul dat
16        int i,j;
17        for(i=0;i<n;i++)
18            for(j=0;j<m;j++)
19                A[i][j] = input.nextInt();
20
21        // afisarea elementelor tabloului bidimensional (matrice)
22        System.out.println ("Matricea citita este: ");
23        for(i=0;i<n;i++){
24            for(j=0;j<m;j++)
25                System.out.print(" "+A[i][j]);
26            System.out.println();
27        }
28
29        int V[ ]= new int[m], suma;
30        for(i = 0; i < n; i++){
31            suma = 0;
32            for(j = 0; j < m; j++)
33                suma += A[i][j];
34            V[i] = suma;
35        }
36
37        System.out.println ("Vectorul suma este: ");
38        for(i=0;i<n;i++)
39            System.out.print(" "+V[i]);
40        }
41    }
42 }
43 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1 Interactive Stdin Inputs

CommandLine Arguments

3 4
1 2 3 4
5 6 7 8
9 10 11 12

Execute   

Result

CPU Time: 0.23 sec(s), Memory: 37468 kilobyte(s)

```
Numarul de linii = 3
Numarul de coloane = 4
Matricea citita este:
1 2 3 4
5 6 7 8
9 10 11 12
Vectorul suma este:
10 26 42
```

5. Se considera o matrice $A_{n \times m}$ ($1 \leq n, m \leq 30$) avand componente numere intregi. Sa se determine **maximul fiecărei coloane si minimul fiecărei linii**.

```
import java.util.Scanner;
public class lab_7_5 {
    public static void main(String[] args){
        /* declararea si citirea variabilelor de memorie in care se vor retine numarul de
linii,
        respectiv numarul de coloane ale tabloului bidimensional */
        int n, m;
        Scanner input = new Scanner(System.in);
        n = input.nextInt();
        System.out.println("Numarul de linii = "+n);
        m = input.nextInt();
        System.out.println("Numarul de coloane = "+m);

        /* declararea variabilei de tip tablou bidimensional A[][] */
        int A[ ][ ]= new int[n][m];
        // citirea elementelor in tabloul dat
        int i,j;
        for(i=0;i<n;i++)
            for(j=0;j<m;j++)
                A[i][j] = input.nextInt();

        // afisarea elementelor tabloului bidimensional (matrice)
        System.out.println ("Matricea citita este: ");
        for(i=0;i<n;i++){
            for(j=0;j<m;j++)
                System.out.print(" "+A[i][j]);
            System.out.println();
        }

        int minim, maxim;

        for(j = 0; j < m; j++){
            maxim = -10000;
            for(i = 0; i < n; i++){
                if(maxim < A[i][j]) maxim = A[i][j];
                System.out.println("Maximul coloanei "+(j+1)+" este egal cu = "+maxim);
            }
        }
        for(i = 0; i < n; i++){
            minim = 10000;
            for(j = 0; j < m; j++){
                if(minim > A[i][j]) minim = A[i][j];
                System.out.println("Minimul liniei "+(i+1)+" este egal cu = "+minim);
            }
        }
    }
}
```

```

1 import java.util.Scanner;
2 public class lab_7_5 {
3     public static void main(String[] args){
4         /* declararea si citirea variabilelor de memorie in care se vor retine numarul de linii,
5         respectiv numarul de coloane ale tabloului bidimensional */
6         int n, m;
7         Scanner input = new Scanner(System.in);
8         n = input.nextInt();
9         System.out.println("Numarul de linii = "+n);
10        m = input.nextInt();
11        System.out.println("Numarul de coloane = "+m);
12
13        /* declararea variabilei de tip tablou bidimensional A[][] */
14        int A[ ][ ] = new int[n][m];
15        // citirea elementelor in tabloul dat
16        int i, j;
17        for(i=0; i<n; i++){
18            for(j=0; j<m; j++){
19                A[i][j] = input.nextInt();
20            }
21        }
22        // afisarea elementelor tabloului bidimensional (matrice)
23        System.out.println ("Matricea citita este: ");
24        for(i=0; i<n; i++){
25            for(j=0; j<m; j++){
26                System.out.print(" "+A[i][j]);
27            }
28            System.out.println();
29        }
30
31        int minim, maxim;
32
33        for(j = 0; j < m; j++){
34            maxim = -10000;
35            for(i = 0; i < n; i++){
36                if(maxim < A[i][j]) maxim = A[i][j];
37            }
38            System.out.println("Maximul coloanei "+(j+1)+" este egal cu = "+maxim);
39        }
40
41        for(i = 0; i < n; i++){
42            minim = 10000;
43            for(j = 0; j < m; j++){
44                if(minim > A[i][j]) minim = A[i][j];
45            }
46            System.out.println("Minimul liniei "+(i+1)+" este egal cu = "+minim);
47        }
48    }
49 }

```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1 Interactive

Stdin Inputs

```

3 4
12 24 -2 123
-90 2 56 11
5 3 -1 10

```

CommandLine Arguments

Result

CPU Time: 0.26 sec(s), Memory: 38896 kilobyte(s)

```

Numarul de linii = 3
Numarul de coloane = 4
Matricea citita este:
12 24 -2 123
-90 2 56 11
5 3 -1 10
Maximul coloanei 1 este egal cu = 12
Maximul coloanei 2 este egal cu = 24
Maximul coloanei 3 este egal cu = 56
Maximul coloanei 4 este egal cu = 123
Minimul liniei 1 este egal cu = -2
Minimul liniei 2 este egal cu = -90
Minimul liniei 3 este egal cu = -1

```

6. Se considera o matrice $A_{n \times n}$ ($1 \leq n \leq 30$) avand componente numere intregi. Sa se memoreze intr-un vector V elementele pozitive aflate deasupra diagonalei principale in matrice, apoi sa se tipareasca vectorul.

```
import java.util.Scanner;
public class lab_7_6 {
    public static void main(String[] args){
        /* declararea si citirea variabilelor de memorie in care se vor retine numarul de
linii
si de coloane ale tabloului bidimensional */
        int n;
        Scanner input = new Scanner(System.in);
        n = input.nextInt();
        System.out.println("Numarul de linii = "+n);

        /* declararea variabilei de tip tablou bidimensional A[][] */
        int A[ ][ ]= new int[n][n];
        // citirea elementelor in tabloul dat
        int i,j;
        for(i=0;i<n;i++)
            for(j=0;j<n;j++)
                A[i][j] = input.nextInt();

        // afisarea elementelor tabloului bidimensional (matrice)
        System.out.println ("Matricea citita este: ");
        for(i=0;i<n;i++){
            for(j=0;j<n;j++)
                System.out.print(" "+A[i][j]);
            System.out.println();
        }

        int V[] = new int[n], k = 0;
        for(i = 0; i < n; i++)
            for(j = 0; j < n; j++)
                if(i < j && A[i][j] > 0) V[k++] = A[i][j];

        System.out.println("Vectorul este = ");
        for(i = 0; i < n; i++)
            System.out.print(" "+V[i]);
    }
}
```

```
1 import java.util.Scanner;
2 public class lab_7_6 {
3     public static void main(String[] args){
4         /* declararea si citirea variabilelor de memorie in care se vor retine numarul de linii
5         si de coloane ale tabloului bidimensional */
6         int n;
7         Scanner input = new Scanner(System.in);
8         n = input.nextInt();
9         System.out.println("Numarul de linii = "+n);
10
11        /* declararea variabilei de tip tablou bidimensional A[][] */
12        int A[ ][ ]= new int[n][n];
13        // citirea elementelor in tabloul dat
14        int i,j;
15        for(i=0;i<n;i++){
16            for(j=0;j<n;j++){
17                A[i][j] = input.nextInt();
18            }
19        }
20        // afisarea elementelor tabloului bidimensional (matrice)
21        System.out.println ("Matricea citita este: ");
22        for(i=0;i<n;i++){
23            for(j=0;j<n;j++){
24                System.out.print(" "+A[i][j]);
25            }
26            System.out.println();
27        }
28
29        int V[] = new int[n], k = 0;
30        for(i = 0; i < n; i++)
31            for(j = 0; j < n; j++)
32                if(i < j && A[i][j] > 0) V[k++] = A[i][j];
33
34        System.out.println("Vectorul este = ");
35        for(i = 0; i < n; i++)
36            System.out.print(" "+V[i]);
37    }
38 }
```

Execute Mode, Version, Inputs & Arguments

JDK 17.0.1 Interactive

Stdin Inputs

```
3
1 2 3
4 5 6
7 8 9
```

CommandLine Arguments

Execute

Result

CPU Time: 0.35 sec(s), Memory: 37116 kilobyte(s)

```
Numarul de linii = 3
Matricea citita este:
1 2 3
4 5 6
7 8 9
Vectorul este =
2 3 6
```

Probleme propuse spre rezolvare

Lab7_1). Să se construiască o matrice pătratică ($n=m$) de dimensiune n^2 ($1 \leq n \leq 30$) cu primele numere pare începând cu 2.

Lab7_2). (Puncte șa) Fie un tablou bidimensional $A(n,m)$. Un punct șa al acestui tablou este un element $a[i_0][j_0]$ cu proprietatea:

$$a[i_0][j_0] = \min \{ a[i_0][j], \text{unde } 0 \leq j \leq m-1 \} = \max \{ a[i][j_0], \text{unde } 0 \leq i \leq n-1 \}$$

Scrieți un program Java care determină punctele șa (dacă există) ale unui tablou bidimensional.

Lab7_3). Fiind dat un tablou bidimensional $A(n,m)$ de numere întregi, să se calculeze și să afișeze suma elementelor aflate pe marginea (rama) tabloului.

Lab7_4). Fiind dat un tablou bidimensional $A(n,m)$ de numere întregi, sa se afișeze:

- a) produsul elementelor pozitive aflate pe linii pare și coloane impare.
- b) suma elementelor pozitive impare din tablou.
- c) numărul elementelor pozitive, negative și respectiv nule din tablou.
- d) elementele aflate pe linii impare și coloane divizibile cu 3.

Lab7_5). Fiind dat un tablou bidimensional $A(n,m)$ de numere întregi, să scrie un program Java care să stabilească de câte ori apare o valoare dată x ca element al tabloului.

Lab7_6). Fiind dat un tablou bidimensional $A(n,m)$ de elemente 0 și 1. Să se scrie un program Java care afișează numărul maxim de elemente egale cu 1 conținute pe o linie.

Lab7_7). Fiind dat un tablou bidimensional $A(n,m)$ de numere întregi. Să se scrie un program Java care să determine linia (liniile) din tablou care conține cele mai multe elemente nenule.

Lab7_8). Fiind dat un tablou bidimensional $A(n,m)$ de numere întregi. Să se scrie un program Java care să afișeze numărul/numerele de ordine a liniilor din tablou care au cele mai multe componente distincte.

Lab7_9). Fiind dat un tablou bidimensional $A(n,m)$ de numere întregi. Să se scrie un program Java care să verifice dacă matricea este diagonală (toate elementele în afara diagonalei principale sunt zero).

Lab7_10). Fiind dat un tablou bidimensional $A(n,m)$ de numere întregi. Să se scrie un program Java care să calculeze suma elementelor aflate sub diagonala principală a unei matrice pătratice.

Bibliografie:

[1] <http://www.pbinfo.ro> Descrierea site-ului: “*www.pbinfo.ro* îți propune să rezolvi probleme de informatică, cu evaluator automat. Știi pe loc dacă soluția ta este corectă sau dacă trebuie să mai lucrezi la ea. Problemele sunt grupate după programa de informatică pentru liceu. Dar nu trebuie să fii la liceu ca să rezolvi aceste probleme. Poți fi elev de gimnaziu, student, profesor sau pur și simplu pasionat de informatică. De fapt, trebuie doar să vrei!!”

[2] <https://www.runceanu.ro/adrian>

[3] Adrian Runceanu „*Programarea și utilizarea calculatoarelor*”, Editura Academica Brâncuși din Târgu-Jiu, 2003, ISBN 973-8436-44-3

[4] Adrian Runceanu, Mihaela Runceanu, „*Noțiuni de programare – limbajul C++*”, Editura Academica Brâncuși din Târgu-Jiu, 2012, ISBN 978-973-144-550-2

[5] Adrian Runceanu, Mihaela Runceanu, „*Algoritmi implementati in limbajul C++. Volumul I – Algoritmi elementari*”, Editura Academica Brâncuși din Târgu Jiu, 2021, ISBN 978-606-9614-06-8