

Lucrare de laborator nr. 5

Matrici

1. Scopul lucrării

În această lucrare se studiază tablourile bidimensionale (matricile) : modul de instanțiere, utilizarea matricilor ca parametrii în metode. Se dau exemple atât cu matrici de numere întregi (tipuri de date primitive) cât și cu matrici ce au ca elemente obiecte (instanțiate dintr-o altă clasă).

2. Breviar teroretic

Matricile reprezintă un caz particular de vectori. O matrice este un vector ale cărui componente sunt tot vectori (liniile matricii).

O matrice se instanțiază astfel:

```
tipDate nume[ ][ ]=new tip[nrLinii][nrColoane];
```

Exemplu:

```
int a[ ][ ]=new int[10][20];
```

O matrice poate fi inițializată și în mod direct, ca în exemplul următor:

Daca $a = \begin{pmatrix} -3 & 0 \\ 1 & 2 \\ -5 & 7 \end{pmatrix}$, o putem initializa direct, astfel:

```
int a[ ][ ]={{-3,0},
             {1,2},
             {-5,7}};
```

O matrice fiind un vector de vectori (vector de linii), plecând de la numele matricii, obținem numărul de linii și numărul de coloane, astfel:

```
nrLinii=nume.length
nrColoane=nume[0].length
```

De aceea, atunci când într-o metodă avem ca parametru o matrice, nu este necesar să mai transmitem ca parametrii și dimensiunile matricii, ci doar numele matricii (dimensiunile rezultă din nume).

3. Probleme rezolvate

Problema 1

Citim de la tastatură numărul de linii și numărul de coloane ale unei matrici. Inițializăm matricea cu numere aleatoare în gama 0...99 . Afișăm această matrice și apoi calculăm și afișăm maximul din matrice.

```
import java.util.*;
class M1
{
    public static void main(String args[ ])
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("numar linii matrice =");
        int nL=sc.nextInt();
        System.out.print("numar coloane matrice =");
        int nC=sc.nextInt();
        int a[ ][ ]=new int[nL][nC];
        final int GAMA=100;
        Random r=new Random( );
        int i,j;
        for(i=0;i<nL;i++)
            for(j=0;j<nC;j++)
                a[i][j]=r.nextInt(GAMA);
        int max=a[0][0];
        for(i=0;i<a.length;i++)
            for(j=0;j<a[0].length;j++)
                if(a[i][j]>max) max=a[i][j];
        //afisare matrice:
        for(i=0;i<nL;i++){
            for(j=0;j<nC;j++)
                System.out.print(a[i][j]+" ");
            System.out.println();
        }
        System.out.println("maxim="+max);
    }
}
```

```
}  
}
```

Problema 2

Se citește o matrice de numere întregi. Să se calculeze și afișeze câte linii din matrice au toate elementele egale între ele.

```
import java.util.*;  
class M2  
{  
    public static void main(String args[ ])   
    {  
        Scanner sc=new Scanner(System.in);  
        System.out.print("numar linii matrice =");  
        int nL=sc.nextInt();  
        System.out.print("numar coloane matrice =");  
        int nC=sc.nextInt();  
        int a[ ][ ]=new int[nL][nC];  
        //citire matrice de la tastatura:  
        int i,j;  
        for(i=0;i<nL;i++)  
            for(j=0;j<nC;j++){  
                System.out.print("nr=");  
                a[i][j]=sc.nextInt();  
            }  
        int contor=0;  
        //Parcurem toate liniile matricii:  
        for(i=0;i<nL;i++){  
            //are linia curenta i, toate elementele egale?  
            boolean are=true;//presupunem ca linia i are  
            // toate elementele egale intre ele  
            //parcurem linia i:  
            for(j=1;j<nC;j++){  
                if(a[i][j]!=a[i][0]){  
                    are=false;  
                    break;  
                }  
            }  
        }  
    }  
}
```

```

    if(are==true)contor++;
}
System.out.println("Sunt "+contor+" linii.");
}
}

```

Problema 3

Se citește de la tastatură un număr natural N . Să se copieze într-o matrice pătrată de dimensiune N , primii N^2 termeni din șirul lui Fibonacci, apoi să se afișeze această matrice.

```

import java.util.*;
class M3
{
    public static void main(String args[ ])
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("N = ");
        int N=sc.nextInt();
        int a[ ][ ]=new int[N][N];
        int i,j;
        //Primii doi termeni in sirul lui Fibonacci sunt 1 si 1.
        //In cazul general: termen curent= termen ultim + termen penultim,
        // deci: crt=ultim+pen.
        //Pentru a obtine si pe primii doi termeni ai sirului Fibonacci cu
        //aceeasi formula, initializam astfel:
        int pen=1;
        int ultim=0;
        for(i=0;i<N;i++)
            for(j=0;j<N;j++){
                a[i][j]=pen+ultim;
                //trecem la urmatorul termen:
                pen=ultim;
                ultim=a[i][j];
            }
        //afisarea matricii:
        for(i=0;i<N;i++){
            for(j=0;j<N;j++)
                System.out.print(a[i][j]+" ");
        }
    }
}

```

```
    System.out.println();  
  }  
}  
}
```

Problema 4

Se citește de la tastatură o matrice de numere întregi, de dimensiuni cunoscute (numărul de linii și numărul de coloane citite anterior). Să se calculeze și afișeze dacă matricea are toate numerele egale între ele. Se va defini o metodă separată ce are ca parametru matricea și care returnează *true* dacă matricea are toate elementele egale.

```
import java.util.*;  
class M4  
{  
    public static void main(String args[ ])   
    {  
        Scanner sc=new Scanner(System.in);  
        System.out.print("numar linii matrice =");  
        int nL=sc.nextInt();  
        System.out.print("numar coloane matrice =");  
        int nC=sc.nextInt();  
        int a[ ][ ]=new int[nL][nC];  
        //citire matrice de la tastatura:  
        int i,j;  
        for(i=0;i<nL;i++)  
            for(j=0;j<nC;j++){  
                System.out.print("nr=");  
                a[i][j]=sc.nextInt();  
            }  
        boolean egale=suntToateEgale(a);  
        if(egale==true)System.out.println("Sunt toate numerele egale.");  
        else System.out.println("Nu sunt toate egale");  
    }  
    private static boolean suntToateEgale(int a[ ][ ])
```

```

{
    for(int i=0;i<a.length;i++)
        for(int j=0;j<a[0].length;j++)
            if(a[i][j]!=a[0][0])return false;
    return true;
}
}

```

Problema 5

Se citește de la tastatură un număr natural N . Se citește o matrice pătrată de numere naturale, de dimensiune N . Să se calculeze și afișeze câte numere pare sunt în matrice.

```

import java.util.*;
class M5
{
    public static void main(String args[ ])
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("numar linii matrice =");
        int N=sc.nextInt();
        int a[ ][ ]=new int[N][N];
        //citire matrice de la tastatura:
        int i,j;
        for(i=0;i<N;i++)
            for(j=0;j<N;j++){
                System.out.print("nr=");
                a[i][j]=sc.nextInt();
            }
        int rezultat=cateNumerePare(a);
        System.out.println("Sunt "+rezultat+" numere pare.");
    }

    private static int cateNumerePare(int a[ ][ ])
    {
        int i,j;
        int contor=0;
        //Parcurgem matricea:
        for(i=0;i<a.length;i++)

```

```
        for(j=0;j<a[0].length;j++)
            if(a[i][j]%2==0)contor++;
    return contor;
}
}
```

Problema 6

Se citește de la tastatură un număr natural N. Se citește o matrice pătrată de numere întregi, de dimensiune N. Să se calculeze și afișeze câte numere pare sunt sub diagonala principală a matricii. Se va defini o metodă separată ce are ca parametru matricea și care returnează numărul de numere pare de sub diagonala principală.

```
import java.util.*;
class M6
{
    public static void main(String args[ ])
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("numar linii matrice =");
        int N=sc.nextInt();
        int a[ ][ ]=new int[N][N];
        //citire matrice de la tastatura:
        int i,j;
        for(i=0;i<N;i++)
            for(j=0;j<N;j++){
                System.out.print("nr=");
                a[i][j]=sc.nextInt();
            }
        System.out.println("Sunt "+calculPare(a)+" numere pare.");
    }

    private static int calculPare(int a[ ][ ])
    {
        int contor=0;
        int i,j;
        //Parcurgem toate liniile matricii, incepand cu a doua linie:
        for(i=1;i<a.length;i++)
            for(j=0;j<i;j++)//sub diagonale principala
                if(a[i][j]%2==0)contor++;
    }
}
```

```

    return contor;
}
}

```

Problema 7

Se citește de la tastatură un număr natural N . Se citește o matrice pătrată de numere întregi, de dimensiune N . Să se calculeze și afișeze dacă matricea are toate elementele în ordine crescătoare. Se va defini o metodă separată ce are ca parametru matricea și care returnează *true* dacă matricea are toate elementele în ordine crescătoare.

```

import java.util.*;
class M7
{
    public static void main(String args[ ])
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("numar linii matrice =");
        int N=sc.nextInt();
        int a[ ][ ]=new int[N][N];
        //citire matrice de la tastatura:
        int i,j;
        for(i=0;i<N;i++)
            for(j=0;j<N;j++){
                System.out.print("nr=");
                a[i][j]=sc.nextInt();
            }
        boolean rezultat=suntCrescatoare(a);
        if(rezultat)System.out.println("Sunt in ordine crescatoare.");
        else System.out.println("Nu sunt in ordine crescatoare.");
    }

    private static boolean suntCrescatoare(int a[ ][ ])
    {
        int i,j;
        //Parcurgem toate liniile matricii:
        for(i=0;i<a.length;i++){
            //sunt crescatoare in linia i?
            for(j=0;j<a.length-1;j++)

```

```
        if(a[i][j]>a[i][j+1])return false;//nu sunt
        //Daca linia curenta nu este ultima, ultimul element al liniei
        // curente este mai mic decat primul din linia urmatoare?
        if(i<a.length-1)//nu e ultima linie
            if(a[i][a.length-1]>a[i+1][0])return false;
    }
    return true;
}
}
```

Problema 8

Se definește clasa Punct ce are ca variabilele de instanță *private* pe x și y, coordonatele unui punct în plan. În această clasă se definește constructorul și se definesc și metodele publice de tip *set()* și *get()*. Se formează o matrice pătrată de obiecte Punct. Dimensiunea matricii este N și se citește de la tastatură. De asemenea, se citesc de la tastatură cele N^2 obiecte Punct, care se memorează pe pozițiile corespunzătoare în matrice. Care linie din matrice are cele mai multe obiecte Punct ce au coordonatele x și y egale ?

```
import java.util.*;
class Punct
{
    private int x;
    private int y;
    //constructorul clasei:
    public Punct(int x, int y)
    {
        this.x=x;
        this.y=y;
    }
    public int getX()
    {
        return x;
    }
    public int getY()
    {
        return y;
    }
    public void setX(int x)
```

```

        {
            this.x=x;
        }
        public void setY(int y)
        {
            this.y=y;
        }
    }

class M8
{
    public static void main(String args[ ])
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("N = ");
        int N=sc.nextInt();
        Punct a[ ][ ]=new Punct[N][N];
        int i,j;
        int x,y;
        //construirea matricii:
        for(i=0;i<N;i++)
            for(j=0;j<N;j++){
                System.out.print("punct curent: x= ")      ;
                x=sc.nextInt();
                System.out.print("punct curent: y= ")      ;
                y=sc.nextInt();
                //formarea si memorarea obiectului Punct curent:
                a[i][j]=new Punct(x,y);
            }
        //initializari:
        int nrLinie=-1;
        int maxEgale=-1;
        //Parcurem matricea:
        for(i=0;i<N;i++) {
            //calcul nr puncte ce au coordonatele x si y egale, in linia i:
            int nrEgale=0;
            //parcurem linia i:
            for(j=0;j<N;j++)
                if( a[i][j].getX()==a[i][j].getY() )nrEgale++;
            if(nrEgale>maxEgale){

```

```
        maxEgale=nrEgale;  
        nrLinie=i;  
    }  
}  
System.out.print("Linia "+nrLinie+" are "+maxEgale+  
    " puncte cu x=y");  
}  
}
```

3. Probleme propuse

Problema 1

Se citește de la tastatură o matrice de numere întregi, de dimensiuni cunoscute (numărul de linii și numărul de coloane citite anterior). Se citesc în variabilele lin1 și lin2, numerele a două linii din matrice. Să se comute în matrice aceste două linii între ele.

Problema 2

Se citește de la tastatură un număr natural N. Să se copieze într-o matrice pătrată de dimensiune N, primele N^2 numere prime. Apoi să se afișeze această matrice.

Problema 3

Se citește de la tastatură un număr natural N. Se citește o matrice pătrată de numere întregi, de dimensiune N. Să se calculeze și afișeze câte numere pare sunt deasupra diagonalei secundare a matricii. Se va defini o metodă separată ce are ca parametru matricea și care returnează numărul de numere pare de deasupra diagonalei secundare.

Problema 4

Se citește de la tastatură un număr natural N. Se citește o matrice pătrată de numere naturale, de dimensiune N. Să se calculeze și afișeze câte numere prime sunt în matrice.

Problema 5

Se citește de la tastatură un număr natural N. Se citește o matrice pătrată de numere naturale, de dimensiune N. Să se calculeze și afișeze care linie din matrice conține cele mai multe numere prime.

Problema 6

Să se copieze prin program, într-o matrice pătrată de dimensiune 6, toate literele mari ale alfabetului englez. Pozițiile neocupate din matrice se vor completa cu spații. Să se afișeze apoi această matrice.

Problema 7

Se citește de la tastatură un număr natural N . Se citesc N numere naturale într-un vector $a[]$. Să se copieze toate numerele din vectorul $a[]$ într-o matrice pătrată $b[][]$, de dimensiune minimă, corespunzătoare. Spațiile eventual neocupate din matrice se vor completa cu valoarea -1.

Problema 8

Se citește de la tastatură un număr natural N . Se citește de la tastatură o matrice pătrată de dimensiune N , de numere întregi. Să se calculeze și afișeze maximul din matrice și linia și coloana în care apare maximul. Se va defini pentru aceasta o metodă separată, ce va fi apelată din *main()*, care are ca parametru de intrare matricea și care returnează trei rezultate: maximul, linia în care apare maximul și coloana în care apare maximul.
