

1. Să se definească o clasă MatriceExtinsa pentru o matrice pătrată de întregi, cu constructor fără parametri (crează 3×3 cu zerouri) și constructor parametrizat (dintr-o matrice dată, folosind partea ei pătrată), care oferă metode pentru citire de la tastatură, afișare, suma diagonalelor, verificarea simetriei față de diagonala principală și suma conturului; în main se creează un obiect, se citește, se afișează și se apelează toate metodele.

```
1 package lab5;
2 import java.util.*;
3 class MatriceExtinsa {
4     private int[][] a;
5     // 3x3 cu zero
6     public MatriceExtinsa() {
7         a = new int[3][3];
8     }
9     // dintr-o matrice sursă: ia partea pătrată de dimensiune min(n, m)
10    public MatriceExtinsa(int[][] src) {
11        int n = src.length;
12        int m = (n > 0 ? src[0].length : 0);
13        int d = Math.min(n, m);
14        if (d <= 0) { a = new int[0][0]; return; }
15        a = new int[d][d];
16        for (int i = 0; i < d; i++)
17            for (int j = 0; j < d; j++)
18                a[i][j] = src[i][j];
19    }
20    public void citire(Scanner sc) {
21        System.out.print("Dimensiune n (pătratică): ");
22        int n = sc.nextInt();
23        a = new int[n][n];
24        System.out.println("Introduceți " + (n*n) + " numere întregi:");
25        for (int i = 0; i < n; i++)
26            for (int j = 0; j < n; j++)
27                a[i][j] = sc.nextInt();
28    }
29    public void afisare() {
30        System.out.println("Matricea:");
31        for (int[] r : a) {
32            for (int x : r) System.out.printf("%5d", x);
33            System.out.println();
34        }
35    }
36 }
```

```

34     }
35 }
36
37 public int sumaDiagPrincipala() {
38     int n = a.length, s = 0;
39     for (int i = 0; i < n; i++) s += a[i][i];
40     return s;
41 }
42
43 public int sumaDiagSecundara() {
44     int n = a.length, s = 0;
45     for (int i = 0; i < n; i++) s += a[i][n-1-i];
46     return s;
47 }
48
49 public boolean esteSimetrica() {
50     int n = a.length;
51     for (int i = 0; i < n; i++)
52         for (int j = i+1; j < n; j++)
53             if (a[i][j] != a[j][i]) return false;
54     return true;
55 }
56
57 public int sumaContur() {
58     int n = a.length;
59     if (n == 0) return 0;
60     if (n == 1) return a[0][0];
61     int s = 0;
62     // prima si ultima linie
63     for (int j = 0; j < n; j++) { s += a[0][j]; s += a[n-1][j]; }
64     // marginile laterale (fără colțuri)
65     for (int i = 1; i < n-1; i++) { s += a[i][0]; s += a[i][n-1]; }
66     return s;
67 }

```

2. Să se implementeze clasele Student (cu nume, vârstă și vector de note, constructor parametrizat, metodă pentru medie și metodă de afișare) și Grupa (cu vector de studenți și metode pentru citirea unei grupe, afișare și determinarea studentului cu media maximă); în main se citește grupa, se afișează toți studenții și se afișează studentul cu media cea mai mare.

```

1 package lab5;
2 import java.util.*;
3 class Student {
4     private String nume;
5     private int varsta;
6     private double[] note;
7     public Student(String nume, int varsta, double[] note) {
8         this.nume = nume;
9         this.varsta = varsta;
10        this.note = new double[note.length];
11        for (int i = 0; i < note.length; i++) this.note[i] = note[i];
12    }
13    public double medie() {
14        if (note.length == 0) return 0;
15        double s = 0;
16        for (double x : note) s += x;
17        return s / note.length;
18    }
19    public void afisare() {
20        System.out.print("Student: " + nume + ", " + varsta + " ani, note=");
21        for (int i = 0; i < note.length; i++) {
22            System.out.print(note[i]);
23            if (i + 1 < note.length) System.out.print(", ");
24        }
25        System.out.println(", medie=" + String.format("%.2f", medie()));
26    }
27    public String getNume() { return nume; }
28 }
29 class Grupa {
30     private Student[] studenti;
31     public void citireGrupa(Scanner sc) {
32         System.out.print("Număr studenți: ");
33         int n = sc.nextInt();
34         studenti = new Student[n];
35
36         for (int i = 0; i < n; i++) {
37             System.out.println("=== Student " + (i+1) + " ===");
38             System.out.print("Nume: ");
39             String nume = sc.next();
40             System.out.print("Vârsta: ");
41             int varsta = sc.nextInt();
42             System.out.print("Număr note: ");
43             int k = sc.nextInt();
44             double[] note = new double[k];
45             System.out.println("Introduceți " + k + " note:");
46             for (int j = 0; j < k; j++) note[j] = sc.nextDouble();
47             studenti[i] = new Student(nume, varsta, note);
48         }
49     public void afisareGrupa() {
50         System.out.println("\n--- Grupa ---");
51         for (Student s : studenti) s.afisare();
52     }
53
54     public Student studentMaxMedie() {
55         if (studenti == null || studenti.length == 0) return null;
56         Student best = studenti[0];
57         for (int i = 1; i < studenti.length; i++)
58             if (studenti[i].medie() > best.medie()) best = studenti[i];
59         return best;
60     }

```

```

61 public static void main(String[] args) {
62     Scanner sc = new Scanner(System.in);
63     Grupa g = new Grupa();
64     g.citireGrupa(sc);
65     g.afisareGrupa();
66     Student top = g.studentMaxMedie();
67     if (top != null) {
68         System.out.println("Studentul cu media maximă: " + top.getNume());
69     } else {
70         System.out.println("Nu există studenți.");
71     }
72 }
73 }
74

```

3. Să se creeze clasa ProdusMatrice care primește prin constructor o matrice de prețuri preturi (n×m) și un vector cantitati (m), calculează matricea rezultată rez[i][j] = preturi[i][j] * cantitati[j], o afișează și determină suma totală a tuturor elementelor; în main se citesc datele, se construiește obiectul și se afișează rezultatele.

```

1 package lab5;
2 import java.util.*;
3 class ProdusMatrice {
4     private int[][] preturi; // n x m
5     private int[] cantitati; // m
6     public ProdusMatrice(int[][] preturi, int[] cantitati) {
7         // presupunem date corecte (m egal cu lungimea vectorului)
8         int n = preturi.length;
9         int m = (n > 0 ? preturi[0].length : 0);
10        this.preturi = new int[n][m];
11        for (int i = 0; i < n; i++)
12            for (int j = 0; j < m; j++)
13                this.preturi[i][j] = preturi[i][j];
14        this.cantitati = new int[m];
15        for (int j = 0; j < m; j++) this.cantitati[j] = cantitati[j];
16    }
17
18    public int[][] valoareTotala() {
19        int n = preturi.length, m = (n > 0 ? preturi[0].length : 0);
20        int[][] rez = new int[n][m];
21        for (int i = 0; i < n; i++)
22            for (int j = 0; j < m; j++)
23                rez[i][j] = preturi[i][j] * cantitati[j];
24        return rez;
25    }
26
27    public int sumaTotala() {
28        int s = 0;
29        int[][] r = valoareTotala();
30        for (int[] lin : r) for (int x : lin) s += x;
31        return s;
32    }
33

```

```

34 public static void afisareMatrice(int[][] a, String titlu) {
35     System.out.println(titlu);
36     for (int[] lin : a) {
37         for (int x : lin) System.out.printf("%8d", x);
38         System.out.println();
39     }
40 }
41
42 public static void main(String[] args) {
43     Scanner sc = new Scanner(System.in);
44
45     System.out.print("n (rânduri): ");
46     int n = sc.nextInt();
47     System.out.print("m (coloane): ");
48     int m = sc.nextInt();
49
50     int[][] preturi = new int[n][m];
51     System.out.println("Introduceți " + (n*m) + " valori pentru preturi:");
52     for (int i = 0; i < n; i++)
53         for (int j = 0; j < m; j++)
54             preturi[i][j] = sc.nextInt();
55
56     int[] cantitati = new int[m];
57     System.out.println("Introduceți " + m + " cantități:");
58     for (int j = 0; j < m; j++) cantitati[j] = sc.nextInt();
59
60     ProdusMatrice pm = new ProdusMatrice(preturi, cantitati);
61     int[][] rez = pm.valoareTotala();
62     afisareMatrice(rez, "Matricea valorilor (preturi * cantitati[j]):");
63     System.out.println("Suma totală = " + pm.sumaTotala());
64 }
65 }
66

```