

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут прикладної математики і фундаментальних наук

Кафедра прикладної математики



Лабораторна робота №9
з курсу “Програмування настільних
застосунків”
Тема: “Винятки в мові програмування Java.”

Виконав студент групи ПМ-33
Венгринюк Олег
Прийняла
Терендій О. В.

Завдання для лабораторної роботи:

1. Ознайомитись з теоретичним матеріалом.
2. Розробити алгоритм розв'язання індивідуального завдання.
3. Скласти програму на мові Java у відповідності з розробленим алгоритмом.
4. Виконати обчислення по програмі.

Програмний код

```
import java.util.Scanner;
public class MyMatrix {
    static private int defaultN = 3, defaultM = 3;
    private int n = defaultN, m = defaultM;
    private int min = 3, max = 5;
    private double[][] M;
    public MyMatrix(){
        this.M = this.initRandomMatrix(this.n, this.m);
    }
    public MyMatrix(double[][] M, int n, int m){
        this.n = n;
        this.m = m;
        this.M = M;
        try {
            if (n <= 0 || m <= 0) {
                throw new IllegalArgumentException("Matrix's dimensions cannot negative
or zero");
            }
        }catch(IllegalArgumentException exception){
            System.out.println("Incorrect n or m");
            System.out.println("Using default values");
            System.out.println("Creating new random values");
            this.n = defaultN;
            this.m = defaultM;
            this.M = this.initRandomMatrix(n, m);
        }
    }
    public MyMatrix(int n, int m){
        try {
            if (n <= 0 || m <= 0) {
                throw new IllegalArgumentException("Matrix's dimensions cannot negative
or zero");
            }
        }catch(IllegalArgumentException exception){
            System.out.println("Incorrect n or m");
            System.out.println("Using default values");
            System.out.println("Creating new random values");
            n = defaultN;
            m = defaultM;
        }finally {
            this.M = this.initRandomMatrix(n, m);
        }
    }
    public static MyMatrix initMatrix(int n, int m){
        try {
            if (n <= 0 || m <= 0) {
                throw new IllegalArgumentException("Matrix's dimensions cannot negative
or zero");
            }
        }catch(IllegalArgumentException exception){
            System.out.println("Incorrect n or m");
            System.out.println("Using default values");
        }
    }
}
```

```

        System.out.println("Creating new random values");
        n = defaultN;
        m = defaultM;
    }
    double[][] M = new double[n][m];
    Scanner input = new Scanner(System.in);
    String val;
    System.out.println("Input matrix");
    for(int i=0; i<n; ++i){
        for(int j=0; j<m; ++j) {
            System.out.print(String.format("M[%d][%d]: ", i + 1, j + 1));
            val = input.next();
            try{
                M[i][j] = Double.parseDouble(val);
            }catch (Exception e){
                System.out.println("Fail to save value");
                System.out.println("Set zezo value");
                M[i][j] = 0;
            }
        }
    }
    return new MyMatrix(M, n, m);
}
public double[][] getArray(){
    return this.M;
}
public double getEl(int i, int j){
    try{
        return this.M[i][j];
    } catch (ArrayIndexOutOfBoundsException e){
        System.out.println("Invalid index");
        System.out.println("Zero will be returned");
        return 0;
    }
}
public int getDim(int d){
    if (d==0){
        return n;
    }else if (d==1) {
        return m;
    }else throw new IllegalArgumentException("There is no more dimensions");
}
public MyMatrix add(MyMatrix a){
    try {
        if ((a.getDim(0) != n) || a.getDim(1) != m) {
            throw new IllegalArgumentException("Invalid matrix's dimensions");
        }
    }catch(ArrayIndexOutOfBoundsException excpetion){
        System.out.println("Invalid matrix's dimensions");
        System.out.println("Return first matrix");
        return this;
    }
    double[][] res = new double[n][m];
    double[][] aArray = a.getArray();
    for(int i=0; i<n; ++i){
        for(int j=0; j<m; ++j){
            res[i][j] = aArray[i][j] + this.M[i][j];
        }
    }
    return new MyMatrix(res, n, m);
}
public MyMatrix sub(MyMatrix a){
    try {
        if ((a.getDim(0) != n) || a.getDim(1) != m) {
            throw new IllegalArgumentException("Invalid matrix's dimensions");
        }
    }catch(ArrayIndexOutOfBoundsException excpetion){
        System.out.println("Invalid matrix's dimensions");
    }
}

```

```

        System.out.println("Return first matrix");
        return this;
    }
    double[][] res = new double[n][m];
    double[][] aArray = a.getArray();
    for(int i=0; i<n; ++i){
        for(int j=0; j<m; ++j){
            res[i][j] = this.M[i][j] - aArray[i][j];
        }
    }
    return new MyMatrix(res, n, m);
}
public MyMatrix mul(MyMatrix a) {
    try {
        if (m != a.getDim(0)) {
            throw new IllegalArgumentException("Invalid matrix's dimensions");
        }
    } catch (ArrayIndexOutOfBoundsException excpetion){
        System.out.println("Invalid matrix's dimensions");
        System.out.println("Return empty matrix");
        return this;
    }
    int n = this.n;
    int m = a.getDim(1);
    double[][] res = new double[n][m];
    double[][] aMatrix = a.getArray();
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            res[i][j] = 0.;
        }
    }
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            for (int k = 0; k < this.m; k++) {
                res[i][j] += this.M[i][k] * aMatrix[k][j];
            }
        }
    }
    return new MyMatrix(res, n, m);
}
@Override
public String toString() {
    String resultString = "";
    for (int i = 0; i < M.length; i++) {
        for (int j = 0; j < M[i].length; j++) {
            resultString += String.format("M[%d][%d]: %.1f ", i+1, j+1, M[i][j]);
        }
        resultString += "\n";
    }
    return resultString;
}
private double[][] initRandomMatrix(int n, int m){
    double[][] M = new double[n][m];
    for(int i=0; i<n; ++i){
        for(int j=0; j<m; ++j){
            M[i][j] = Math.round((Math.random() * ((max - min) + 1)) + min);
        }
    }
    return M;
}
}
}

```

Результат виконання:

```
Input matrix
M[1][1]: 1
M[1][2]: 2
M[1][3]: 1
Fail to save value
Set zero value
M[2][1]: k
Fail to save value
Set zero value
M[2][2]: j
Fail to save value
Set zero value
M[2][3]: -0
M[3][1]: -1
Fail to save value
Set zero value
M[3][2]: 9
M[3][3]: 8
A
M[1][1]: 3.0 M[1][2]: 5.0 M[1][3]: 3.0
M[2][1]: 4.0 M[2][2]: 5.0 M[2][3]: 3.0
M[3][1]: 4.0 M[3][2]: 5.0 M[3][3]: 3.0

B:
M[1][1]: 1.0 M[1][2]: 2.0 M[1][3]: 0.0
M[2][1]: 0.0 M[2][2]: 0.0 M[2][3]: -0.0
M[3][1]: 0.0 M[3][2]: 9.0 M[3][3]: 8.0

A + B
M[1][1]: 4.0 M[1][2]: 7.0 M[1][3]: 3.0
M[2][1]: 4.0 M[2][2]: 5.0 M[2][3]: 3.0
M[3][1]: 4.0 M[3][2]: 14.0 M[3][3]: 11.0

A - B
M[1][1]: 2.0 M[1][2]: 3.0 M[1][3]: 3.0
M[2][1]: 4.0 M[2][2]: 5.0 M[2][3]: 3.0
M[3][1]: 4.0 M[3][2]: -4.0 M[3][3]: -5.0

A * B
M[1][1]: 3.0 M[1][2]: 33.0 M[1][3]: 24.0
M[2][1]: 4.0 M[2][2]: 35.0 M[2][3]: 24.0
M[3][1]: 4.0 M[3][2]: 35.0 M[3][3]: 24.0

Catch exception IllegalArgumentException
Invalid index
Zero will be returned
0.0

Catch exception IllegalArgumentException
Invalid index
Zero will be returned
0.0
```

Висновок: в ході виконання було освоєно вийняти в мові програмування Java. Було реалізовано калькулятор для роботи з матрицями. Було передбачено вийнятки, що виникають під час виконання.