



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційні систем та технологій

**Лабораторна робота №5**  
із дисципліни «Основи програмування»

**Тема:** Масиви 2

Виконали:  
Студенти групи ІА-24  
Призвіще:  
Шкарніков Антон,  
Кармазіна Анастасія,  
Сотніков Олексій.

Перевірив:  
Колеснік Валерій Миколайович

### Хід роботи:

1. Повторити теоретичні відомості
2. Виконати завдання з таблиці 2 відповідно до свого варіанту у таблиці 1.
  - Кожне завдання має бути реалізовано як окремий клас.
  - Кожен клас має складатись щонайменше з двох методів:
    - `public static void main(String[] args)` - точка входу. Містить код, що кілька разів знаходить результат завдання при різних значеннях аргументів та параметрів. Для перевірки мають бути присутні як дозволені так і заборонені комбінації аргументів та параметрів.
    - Метод, що реалізує задане завдання. Метод має перевіряти аргументи та у разі їх помилковості аварійно закінчувати свою роботу шляхом викидання стандартного виключення `IllegalArgumentException`, `NullPointerException` або `IndexOutOfBoundsException` (дивись л/р №3). В жодному разі цей метод не повинен напряду взаємодіяти з користувачем через консоль або інший UI (ніколи не змішуйте бізнес-логіку та користувацький інтерфейс).
  - Клас може містити інші допоміжні методи.

### Результат виконання роботи:

З Таблиці 1 ( Таблиця варіантів). Варіант 4 (група 4)

Завдання № 1 - 4

Завдання № 2 - 10

4	<code>void transpose(int[][] matrix) {}</code> Транспонувати квадратну матрицю.
10	<code>boolean isUnbalanced(int[][] results) {}</code> Таблиця футбольного чемпіонату задана квадратною матрицею, у якій усі елементи, що належать головній діагоналі, дорівнюють нулю, а кожний елемент, що не належить головній діагоналі, дорівнює 2, 1 або 0 (числу очок, набраних у грі: 2 - вигрaш, 1 - нічия, 0 - прогpaш). З'ясувати, чи є хоча б одна команда, що виграла більше половини ігор.

### Код завдання № 1 - 4

```
import java.lang.Math;  
  
public class Main {
```

```

public static void main(String[] args) {
    int[][] original = {{1, 6, 4}, {2, 4, 3}, {3, 4, 2}};

    System.out.println("Матриця без транспортування:");
    printResult(original);

    System.out.println("Транспортована матриця:");
    printResult(Transpose(original));
}

public static int[][] Transpose(int[][] original){
    int[][] transposed = new int[original[1].length]
[original[0].length];

    for (int i = 0; i < original[0].length; i++) {
        for (int j = 0; j < original[1].length; j++) {
            transposed[i][j] = original[j][i];
        }
    }
    return transposed;
}

public static void printResult(int[][] matrix){
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            System.out.print(matrix[i][j] + " ");
        }
        System.out.println();
    }
}

```

C:\Users\Nastya\.jdk\openjdk-19\bin\java.exe "-javaagent:C:\Program Files\JetBrai  
Матриця без транспортування:  
1 6 4  
2 4 3  
3 4 2  
Транспортована матриця:  
1 2 3  
6 4 4  
4 3 2

Process finished with exit code 0

## Код завдання № 2 — 10

```

public class task2 {

    public static void main(String[] args) {
        int[][] results = new int[4][];
        results[0] = new int[] {0, 2, 1, 0};
    }
}

```

```

results[1] = new int[] {0, 0, 1, 1, 2};
results[2] = new int[] {1, 1, 0, 1};
results[3] = new int[] {2, 1, 1, 0};

printResults(results);

results = null;

printResults(results);

results = new int[][] {};

printResults(results);

results = new int[][] { {1, 1, 2},
                        {1, 0, 2},
                        {0, 0, 0} };

printResults(results);

results = new int[][] { {0, 2, 1, 0, 0},
                        {0, 0, 2, 1, 2},
                        {1, 0, 0, 1, 0},
                        {2, 1, 1, 0, 2},
                        {2, 0, 2, 0, 0} };

printResults(results);

results = new int[][] { {0, 2, 2},
                        {0, 0, 2},
                        {0, 0, 0} };

printResults(results);
}

public static boolean isUnbalanced(int[][] results) {
    if (results == null) {
        throw new NullPointerException("Matrix shouldn't be null!");
    }
    if (results.length == 0) {
        throw new IndexOutOfBoundsException("Matrix shouldn't be empty!");
    }
    for (int i = 0; i < results.length; i++) {
        int count = 0;
        if (results[i].length != results.length) {
            throw new IndexOutOfBoundsException("Matrix should be square!");
        }
        for (int j = 0; j < results.length; j++) {

```

```

        if (i != j) {
            count += results[i][j] == 2 ? 1 : 0;
        } else if (results[i][j] != 0){
            throw new IllegalArgumentException("Main diagonal should
consist of zeros only!");
        }
    }
    if (count > (results.length - 1) / 2) {
        return true;
    }
}
return false;
}

public static void printResults(int[][] results) {
    if (results == null){
        System.out.print("matrix: " + results + " result: ");
    } else {
        System.out.println("matrix: ");
        for (int i = 0; i < results.length; i++) {
            for (int j = 0; j < results[i].length; j++) {
                System.out.print(results[i][j] + " ");
            }
            System.out.println();
        }
        System.out.print("result: ");
    }
    try {
        System.out.println(isUnbalanced(results));
    } catch (IllegalArgumentException | IndexOutOfBoundsException |
NullPointerException e) {
        System.out.println("Error! " + e.getMessage());
    }

    System.out.println("=====");
}
}

```

```

matrix:
0 2 1 0
0 0 1 1 2
1 1 0 1
2 1 1 0
result: Error! Matrix should be square!
=====
matrix: null result: Error! Matrix shouldn't be null!
=====
matrix:
result: Error! Matrix shouldn't be empty!
=====
matrix:
1 1 2
1 0 2
0 0 0
result: Error! Main diagonal should consist of zeros only!
=====
matrix:
0 2 1 0 0
0 0 2 1 2
1 0 0 1 0
2 1 1 0 2
2 0 2 0 0
result: false
=====
matrix:
0 2 2
0 0 2
0 0 0
result: true
=====

```

**Висновок:** у даній лабораторній роботі ми створювали код, що транспортує квадратну матрицю, розв'язали задачу про футбольне поле, з'ясували, чи є хоча б одна команда, що виграла більше половини ігор.

Масив – впорядкований скінчений набір даних одного типу, які зберігаються в послідовно розташованих комірках оперативної пам'яті і мають спільну назву, яку надає користувач.

Операція знаходження матриці, транспонованої до даної, називається *транспонуванням матриці*. Транспонуванням матриці називають впорядковану заміну рядків матриці стовпцями.