



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

## Лабораторна робота №5

### Масиви 2

Виконали  
студенти групи ІА-23:  
Степанов Нікіта,  
Семашко Олександр,  
Ширяєв Даніїл.

Перевірив:  
Колеснік В.М.

**Хід роботи:**

**Розподіл варіантів:**

3) Ширяєв Даніїл

9) Степанов Нікіта

## Завдання 3)

```
1 package lab5;
2
3 public class lab5task3 {
4     public static void main(String[] args) {
5         printResults(new int[][]{{0, 1, 10}, {1, 5, 0}, {-5, 2, 5}});
6         printResults(matrix: null);
7         printResults(new int[][]{{1}});
8         printResults(new int[][]{{0, 3, 7, 9, -2}, {-6, 1, 2, 0, 0}});
9         printResults(new int[][]{null, {1, 2}});
10        printResults(new int[][]{{1}, {2}, {3}, {4}});
11        printResults(new int[][]{{1, 6, 5, 6}, {-7, 0, -7, 3}, {4, 3, 1, 0}, {5, 4, 1, 2}});
12        printResults(new int[][] {{1}, {1, 2, 3}});
13    }
14
15    1 usage
16    @ static void swapColumns(int[][] matrix) {
17        int max = matrix[0][0];
18        int min = matrix[0][0];
19        int maxColumn = 0;
20        int minColumn = 0;
21        for (int i = 0; i < matrix.length; i++) {
22            for (int j = 0; j < matrix[i].length; j++) {
23                if (matrix[i][j] < min) {
24                    min = matrix[i][j];
25                    minColumn = j;
26                }
27                if (matrix[i][j] >= max) {
28                    max = matrix[i][j];
29                    maxColumn = j;
30                }
31            }
32            for (int i = 0; i < matrix.length; i++) {
33                int buffer = matrix[i][maxColumn];
34                matrix[i][maxColumn] = matrix[i][minColumn];
35                matrix[i][minColumn] = buffer;
36            }
37        }
38
39        1 usage
40        static void argumentCheck(int[][] matrix) {
41            if (matrix == null) {
42                throw new NullPointerException("The matrix is null!");
43            }
44            for (int[] row : matrix) {
45                if (row == null) {
46                    throw new NullPointerException("The matrix has null elements!");
47                }
48            }
49        }
```

```

48     }
49
50     8 usages
51     static void printResults(int[][] matrix) {
52         System.out.println("Matrix:");
53         try {
54             argumentCheck(matrix);
55             printMatrix(matrix);
56             swapColumns(matrix);
57             System.out.println("Result:");
58             printMatrix(matrix);
59         } catch (NullPointerException e) {
60             System.out.println("Exception! " + e.getMessage());
61         } catch (ArrayIndexOutOfBoundsException e) {
62             System.out.println("Exception! The matrix isn't rectangular");
63         }
64
65     2 usages
66     @ static void printMatrix(int[][] matrix) {
67         for (int[] row : matrix) {
68             for (int element : row) {
69                 System.out.print(element + " ");
70             }
71             System.out.println();
72         }
73     }
74

```

Matrix:

0 1 10

1 5 0

-5 2 5

Result:

10 1 0

0 5 1

5 2 -5

Matrix:

Exception! The matrix is null!

Matrix:

1

Result:

1

Matrix:

0 3 7 9 -2

-6 1 2 0 0

Result:

9 3 7 0 -2

0 1 2 -6 0

Matrix:

Exception! The matrix has null elements!

Matrix:

1

2

3

4

Result:

Result:

1

2

3

4

Matrix:

1 6 5 6

-7 0 -7 3

4 3 1 0

5 4 1 2

Result:

6 6 5 1

3 0 -7 -7

0 3 1 4

2 4 1 5

Matrix:

1

1 2 3

Exception! The matrix isn't rectangular

## Завдання 9)

```
1 package lab5task3;
2 public class task9 {
3     public static void main(String[] args) {
4
5         printArray(newMatrix( length: 3, height: 3, maxValuePlusOne: 10));
6         printArray(newMatrix( length: 1, height: 1, maxValuePlusOne: 2));
7         printArray(newMatrix( length: 4, height: 2, maxValuePlusOne: 15));
8         printArray(newMatrix( length: -1, height: -3, maxValuePlusOne: 10));
9         printArray(newMatrix( length: -1, height: 3, maxValuePlusOne: 10));
10        printArray(newMatrix( length: 0, height: 0, maxValuePlusOne: 1));
11        printArray(newMatrix( length: 3, height: 3, maxValuePlusOne: 0));
12        printArray(newMatrix( length: 3, height: 3, Integer.MAX_VALUE));
13        printArray(newMatrix(Integer.MAX_VALUE, Integer.MAX_VALUE, Integer.MAX_VALUE));
14    }
15
16    1 usage
17    @ public static int[][] toArray(int[][] matrix) {
18        if (matrix == null) {
19            throw new NullPointerException("Matrix doesn't exist or can't be created");
20        }
21        int arrayLength = numberOfNonZeroCells(matrix);
22        int [][] array = new int[arrayLength][3];
23        int lastNonOccupiedCell = 0;
24        for (int height = 0; height < matrix.length; height++) {
25            for (int length = 0; length < matrix[height].length; length++) {
26                if (matrix[height][length] != 0) {
27                    array[lastNonOccupiedCell][0] = height;
28                    array[lastNonOccupiedCell][1] = length;
29                    array[lastNonOccupiedCell][2] = matrix[height][length];
30                    lastNonOccupiedCell++;
31                }
32            }
33            lastNonOccupiedCell++;
34        }
35        return array;
36    }
37
38    9 usages
39    @ public static int[][] newMatrix(int length, int height, int maxValuePlusOne) {
40        if (length <= 0 | height <= 0 | maxValuePlusOne <= 1) {
41            return null;
42        }
43        try {
44            int[][] matrix = new int[height][length];
45            for (int i = 0; i < height; i++) {
46                for (int j = 0; j < length; j++) {
47                    int randomInt = (int) (Math.random() * maxValuePlusOne);
48                    matrix[i][j] = randomInt;
49                }
50            }
51            return matrix;
52        } catch (OutOfMemoryError e) {
53            return null;
54        }
55    }
```

```

55 public static void printArray(int[][] matrix) {
56     try {
57         int[][] array = toArray(matrix);
58         if (array.length != 0) {
59             System.out.print(Colors.ANSI_GREEN + "Array: " + Colors.ANSI_RESET);
60             for (int i = 0; i < array.length; i++) {
61                 System.out.print("[");
62                 for (int j = 0; j < array[i].length; j++) {
63                     // check if it's not the last element in a nested array
64                     if (j != array[i].length - 1) {
65                         System.out.print(array[i][j] + ", ");
66                     } else {
67                         System.out.print(array[i][j]);
68                     }
69                 }
70                 // check if it's not the last element in an array
71                 if (i != array.length - 1) {
72                     System.out.print(", ");
73                 } else {
74                     System.out.println("]");
75                 }
76             }
77         } else {
78             System.out.println(Colors.ANSI_GREEN + "Array: " + Colors.ANSI_RESET + "[]");
79         }
80     } catch (NullPointerException e) {
81         System.out.println(Colors.ANSI_RED + "EXCEPTION! " + Colors.ANSI_RESET + e.getMessage());
82     }
83 }
84 }

```

1 usage

```

86 @ public static int numberOfNonZeroCells(int[][] matrix) {
87     int nonZeroCounter = 0;
88     for (int i = 0; i < matrix.length; i++) {
89         for (int j = 0; j < matrix[i].length; j++) {
90             if (matrix[i][j] != 0) {
91                 nonZeroCounter++;
92             }
93         }
94     }
95     return nonZeroCounter;
96 }
97 }

```

```

1 package lab5task3;
2
3 public class Colors {
4     3 usages
5     public static final String ANSI_RESET = "\u001B[0m";
6     1 usage
7     public static final String ANSI_RED = "\u001B[31m";
8     2 usages
9     public static final String ANSI_GREEN = "\u001B[32m";
10 }

```



```
Array: [0, 0, 1], [0, 1, 3], [0, 2, 4], [1, 0, 2], [1, 1, 4],  
       [1, 2, 3], [2, 1, 9], [2, 2, 2]  
Array: []  
Array: [0, 0, 7], [0, 1, 1], [0, 2, 14], [0, 3, 3], [1, 0, 3],  
       [1, 1, 7], [1, 3, 11]  
EXCEPTION! Matrix doesn't exist or can't be created  
EXCEPTION! Matrix doesn't exist or can't be created  
EXCEPTION! Matrix doesn't exist or can't be created  
EXCEPTION! Matrix doesn't exist or can't be created  
Array: [0, 0, 2111616637], [0, 1, 1726662154], [0, 2,  
       1022780829], [1, 0, 1018328441], [1, 1, 1551555311], [1, 2,  
       791609588], [2, 0, 172960917], [2, 1, 1063807013], [2, 2,  
       1524291817]  
EXCEPTION! Matrix doesn't exist or can't be created
```

**Висновок:** на цій лабораторній роботі ми вчилися імплементувати двомірні масиви в java, користуватися та працювати з ними, дізналися як зберігаються двомірні масиви в пам'яті та покращили навички кодування.