



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ                    **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА                    **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа,  
обработки и интерпретации больших данных**

## **О Т Ч Е Т**

**по лабораторной работе № 2**

**Вариант 13**

**Название:**                    Арифметические операции

**Дисциплина:**                Языки программирования для работы с большими данными

Студент                    ИУ6-13М  
(Группа)

\_\_\_\_\_  
(Подпись, дата)                    Е.К. Матвиенко  
(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)                    П.В. Степанов  
(И.О. Фамилия)

Москва, 2021

**Цель лабораторной работы:** получение навыков работы с арифметическими операциями языка программирования Java.

### Ход работы:

#### Задание №1:

1. Ввести n строк с консоли. Вывести на консоль те строки, длина которых меньше (больше) средней, а также длину.
2. Ввести n слов с консоли. Найти слово, в котором число различных символов минимально. Если таких слов несколько, найти первое из них.

#### Листинг выполнения подзадачи 1:

```
package laba_2.task_1;
import java.util.Scanner;

public class task_1_1 {
    //Ввести n строк с консоли. Вывести на консоль те строки,
    //длина которых меньше (больше) средней, а также длину.
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter number of strings (n): ");
        int n = in.nextInt();
        in.skip("\n");

        System.out.print("Enter strings: \n");
        String[] array_string = new String[n];
        for (int i = 0; i < n; i++) {
            array_string[i] = in.nextLine();
        }

        int avrg_length = 0;
        for (int i = 0; i < n; i++) {
            avrg_length += array_string[i].length();
        }
        avrg_length = avrg_length/n;
        System.out.println("Average string length: " + avrg_length);

        System.out.print("\nStrings with minimum length: \n");
        for (int i = 0; i < n; i++) {
            if(array_string[i].length() < avrg_length){
                System.out.println(array_string[i]);
            }
        }
        System.out.print("\nStrings with bigger length: \n");
        for (int i = 0; i < n; i++) {
            if(array_string[i].length() > avrg_length){
                System.out.println(array_string[i]);
            }
        }
    }
}
```

### Результат выполнения подзадачи 1:

```
task_1_1 x
C:\Users\metel\.jdk\openjdk-18.0.1.1\bin\
Educational Edition 2021.3.2\lib\idea_rt.
Edition 2021.3.2\bin" -Dfile.encoding=UTF
C:\Users\metel\IdeaProjects\my_lab_2\out\
Enter number of strings (n): 3
Enter strings:
fggggnnn ddd
ddd jjnjbjb hdhh
bbbbbb hhhhhhhcjq
Average string length: 15

Strings with minimum length:
fggggnnn ddd

Strings with bigger length:
ddd jjnjbjb hdhh
bbbbbb hhhhhhhcjq

Process finished with exit code 0
```

### Листинг выполнения подзадачи 2:

```
package laba_2.task_1;

import java.util.Scanner;
import java.util.HashSet;

public class task_1_2 {
    //Ввести n слов с консоли. Найти слово,
    // в котором число различных символов минимально.
    //Если таких слов несколько, найти первое из них.
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter number of words (n): ");
        int n = in.nextInt();
        in.skip("\n");

        System.out.print("Enter strings: \n");
        String[] array_string = new String[n];
        for (int i = 0; i < n; i++) {
            array_string[i] = in.nextLine();
        }

        int min_length = 1000;
        String return_word = "no words";
        for (int i = 0; i < n; i++) {
            HashSet<Character> symbols = new HashSet<>();
            for (char ch: array_string[i].toCharArray()) {
                symbols.add(ch);
            }
            if (symbols.size() < min_length) {
```

```

        min_length = symbols.size();
        return_word = array_string[i];
    }
    System.out.println("Word with minimal number of diff symbols:" +
return_word);
    System.out.println("Number of diff symbols: " + min_length);
}
}

```

## Результат выполнения подзадачи 2:

```

task_1_2 x
C:\Users\metel\.jdk\openjdk-18.0.1.1\bin\java.exe "
Educational Edition 2021.3.2\lib\idea_rt.jar=64310:
Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -classp
C:\Users\metel\IdeaProjects\my_lab_2\out\production
Enter number of words (n): 3
Enter strings:
sdsdsds
ggggg
djsagbchaelo
Word with minimal number of diff symbols:ggggg
Number of diff symbols: 1

Process finished with exit code 0

```

## Задание №2:

1. Найти и вывести наибольшее число возрастающих (убывающих) элементов матрицы, идущих подряд.
2. Найти сумму элементов матрицы, расположенных между первым и вторым положительными элементами каждой строки.

## Листинг выполнения подзадачи 1:

```

package laba_2.task_2;
import java.util.Scanner;
import java.util.concurrent.ThreadLocalRandom;

public class task_2_1 {
    //Найти и вывести наибольшее число возрастающих (убывающих)
    //элементов матрицы, идущих подряд.
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter number of matrix (n): ");
        int n = in.nextInt();
        int[][] matrix = new int[n][n];
    }
}

```

```

for(int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        matrix[i][j] = ThreadLocalRandom.current().nextInt(-n, n + 1);
        System.out.print(matrix[i][j] + " ");
    }
    System.out.println();
}

int number_of_vostrast_el_row = 0;
int index_of_vostrast_el_row = 0;
int number_of_ybuvaush_el_row = 0;
int index_of_ybuvaush_el_row = 0;
for(int i = 0; i < n; i++) {
    int vostrastCount = 1;
    int ybuvaushCount = 1;
    for (int j = 0; j < n-1; j++) {
        if (matrix[i][j] < matrix[i][j+1]){
            vostrastCount++;
        }
        else {
            vostrastCount = 1;
        }
        if (vostrastCount > number_of_vostrast_el_row){
            number_of_vostrast_el_row = vostrastCount;
            index_of_vostrast_el_row = i;
        }
        if (matrix[i][j] > matrix[i][j+1]){
            ybuvaushCount++;
        }
        else {
            ybuvaushCount = 1;
        }
        if (ybuvaushCount > number_of_ybuvaush_el_row){
            number_of_ybuvaush_el_row = ybuvaushCount;
            index_of_ybuvaush_el_row = i;
        }
    }
}
if (number_of_vostrast_el_row > number_of_ybuvaush_el_row){
    System.out.print("number of maximum vostrast(ybuvaushih) elements in row:
" + number_of_vostrast_el_row);
}
else System.out.print("number of maximum vostrast(ybuvaushih) elements in row:
" + number_of_ybuvaush_el_row);

int number_of_vostrast_el_col = 0;
int index_of_vostrast_el_col = 0;
int number_of_ybuvaush_el_col = 0;
int index_of_ybuvaush_el_col = 0;
for(int j = 0; j < n; j++) {
    int ascCount = 1;
    int descCount = 1;
    for (int i = 0; i < n-1; i++) {
        if (matrix[i][j] < matrix[i+1][j]){
            ascCount++;
        }
        else {
            ascCount = 1;
        }
        if (ascCount > number_of_vostrast_el_col){
            number_of_vostrast_el_col = ascCount;
            index_of_vostrast_el_col = i;
        }
    }
}

```

```

        if (matrix[i][j] > matrix[i+1][j]){
            descCount++;
        }
        else {
            descCount = 1;
        }
        if (descCount > number_of_ybuvaush_el_col){
            number_of_ybuvaush_el_col = descCount;
            index_of_ybuvaush_el_col = i;
        }
    }
    if (number_of_vostrast_el_col > number_of_ybuvaush_el_col){
        System.out.print("\n number of maximum vostrast(ybuvaushih) elements in
col: " + number_of_vostrast_el_col);
    }
    else System.out.print("\n number of maximum vostrast(ybuvaushih) elements in
col: " + number_of_ybuvaush_el_col);
}
}

```

### Результат выполнения подзадачи 1:

```

task_2_1 x
C:\Users\metel\.jdk\openjdk-18.0.1.1\bin\java.exe "-javaage
Educational Edition 2021.3.2\lib\idea_rt.jar=64339:C:\Progr
Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -classpath
C:\Users\metel\IdeaProjects\my_lab_2\out\production\my_lab_2
Enter number of matrix (n): 4
0 4 2 -3
2 2 0 -1
0 0 3 2
-1 -3 -4 -3
number of maximum vostrast(ybuvaushih) elements in row: 3
number of maximum vostrast(ybuvaushih) elements in col: 4
Process finished with exit code 0

```

### Листинг выполнения подзадачи 2:

```

package laba_2.task_2;

import java.util.Scanner;
import java.util.concurrent.ThreadLocalRandom;

public class task_2_2 {
    //Найти сумму элементов матрицы,
    //расположенных между первым и вторым положительными элементами каждой строки.

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter number of matrix (n): ");
    }
}

```

```

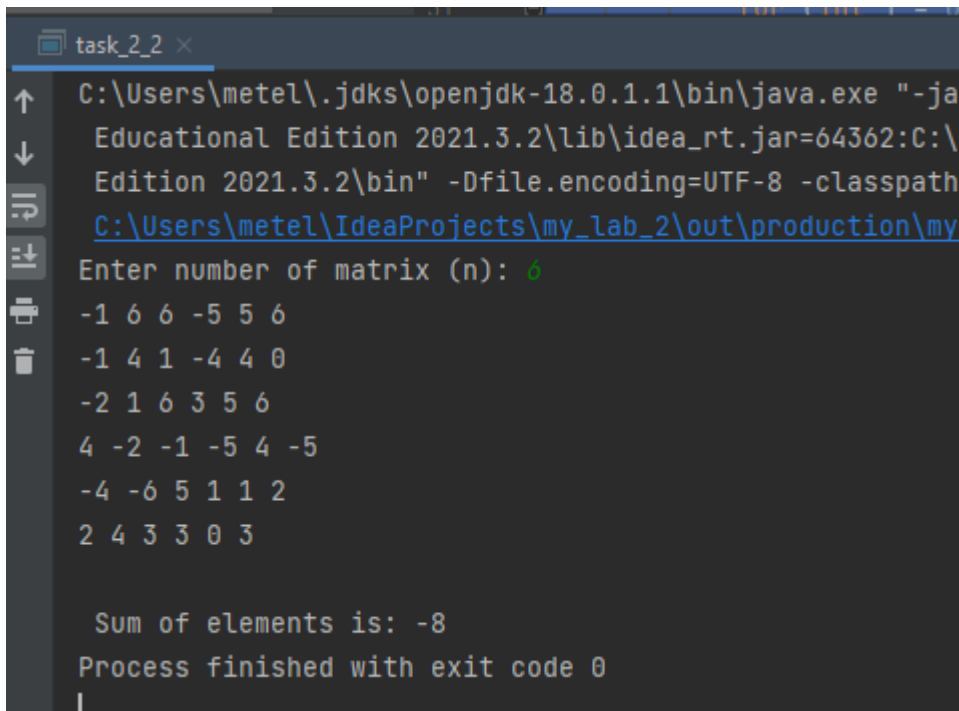
int n = in.nextInt();
int[][] matrix = new int[n][n];

for(int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        matrix[i][j] = ThreadLocalRandom.current().nextInt(-n, n + 1);
        System.out.print(matrix[i][j] + " ");
    }
    System.out.println();
}

int[] matrix_strings = new int[n];
for(int i = 0; i < n; i++) {
    matrix_strings[i] = 0;
    boolean positive_numb_1 = false;
    boolean positive_numb_2 = false;
    int positiv_1=0;
    int positiv_2=0;
    for (int j = 0; j < n-1; j++) {
        if (matrix[i][j] > 0) {
            if (!positive_numb_1) {
                positiv_1 = j;
                positive_numb_1 = true;
            } else if(!positive_numb_2) {
                positiv_2 = j;
                positive_numb_2 = true;
            }
        }
    }
    int rasn = positiv_2 - positiv_1;
    if ((positive_numb_2) &&(rasn>1)) {
        for (int j = positiv_1 + 1; j < positiv_2; j++) {
            matrix_strings[i] += matrix[i][j];
        }
    }
}
int sum=0;
for (int i=0; i<n; i++){
    sum+=matrix_strings[i];
}
System.out.print("\n Sum of elements is: " + sum);
}

```

## Результат выполнения подзадачи 2:



```
task_2_2 x
C:\Users\metel\.jdk\openjdk-18.0.1.1\bin\java.exe "-jav
Educational Edition 2021.3.2\lib\idea_rt.jar=64362:C:\P
Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -classpath
C:\Users\metel\IdeaProjects\my_lab_2\out\production\my_
Enter number of matrix (n): 6
-1 6 6 -5 5 6
-1 4 1 -4 4 0
-2 1 6 3 5 6
4 -2 -1 -5 4 -5
-4 -6 5 1 1 2
2 4 3 3 0 3

Sum of elements is: -8
Process finished with exit code 0
```

Программное решение представлено в репозитории распределённой системы управления версиями Git:

[https://github.com/matvilen/BigDataLanguages/tree/main/my\\_lab\\_2/src/laba\\_2](https://github.com/matvilen/BigDataLanguages/tree/main/my_lab_2/src/laba_2)

**Вывод:** при выполнении лабораторной работы были получены навыки работы с арифметическими операциями языка программирования Java.