**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 №1

**Название:** Арифметические операции

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

Студент ИУ6-22М (Группа) _____ (Подпись, дата) Д.А. Арещенков (И.О. Фамилия)

Преподаватель _____ (Подпись, дата) П.В. Степанов (И.О. Фамилия)

Москва, 2023

**Задание 1**. Ввести n строк с консоли, найти самую короткую и самую длинную строки. Вывести найденные строки и их длину. Упорядочить и вывести строки в порядке возрастания (убывания) значений их длины. Вывести в конце программы фамилию разработчика, дату и время получения задания, а также дату и время сдачи задания. Для получения последней даты и времени следует использовать класс Date.

Код написанной программы представлен в листинге 1 и в репозитории по ссылке https://github.com/Ore111ek/JavaPractice.

Листинг 1 – Программа для работы со строками

```
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Введите количество строк: ");
        int str_num = in.nextInt();
        String[] array = new String[str_num];
        System.out.print("Введите строки: ");
        in.nextLine();
        for(int i = 0; i < str_num; i++){
            array[i] = in.nextLine();
        }
        // Нахождение самых длинной и короткой
        String max = array[0];
        String min = array[0];
        for(int i = 1; i < str_num; i++){
            if(array[i].length()>max.length()){max = array[i];}
            if(array[i].length()<min.length()){min = array[i];}
        }
        System.out.println("\nСамая длинная строка(" +
max.length() + " символов): " + max);
        System.out.println("Самая короткая строка(" +
min.length() + " символов): " + min);
        // Упорядочивание
        for(int j = str_num-1; j > 0; j--){
```

```
            for(int i = 0; i < j; i++) {
                if (array[i].length() > array[i+1].length()) {
                    String t = array[i];
                    array[i] = array[i+1];
                    array[i+1] = t;
                }
            }
        }
        System.out.print("\nУпорядоченные строки:\n");
        for(int i = 0; i < str_num; i++){
            System.out.print(array[i] + "\n");
        }
        System.out.println("\nРазработал:    Арещенков    Дмитрий
Александрович");
        System.out.println("Задание выдано: 17-02-2023 17:25");
        System.out.println("Задание сдано: " + LocalDate.now() +
" " + LocalTime.now());
    }
}
```

Результат выполнения программы продемонстрирован на рисунке 1.

```
Введите количество строк: 4
Введите строки: normal string
The longest string Ever
short
last string

Самая длинная строка(23 символов): The longest string Ever
Самая короткая строка(5 символов): short

Упорядоченные строки:
short
last string
normal string
The longest string Ever

Разработал: Арещенков Дмитрий Александрович
Задание выдано: 17-02-2023 17:25
Задание сдано: 2023-03-03 15:29:18.169804800
```

Рисунок 1 – Результат выполнения программы

**Задание 2**. Ввести с консоли n – размерность матрицы a[n][n]. Задать значения элементов матрицы в интервале значений от -n до n с помощью генератора случайных чисел. Упорядочить строки (столбцы) матрицы в порядке возрастания значений элементов k-го столбца (строки). Выполнить циклический сдвиг заданной матрицы на k позиций вправо (влево, вверх, вниз).

Код написанной программы представлен в листинге 2 и в репозитории по ссылке https://github.com/Ore111ek/JavaPractice.

Листинг 2 – Программа для работы с матрицей

```
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Введите размерность матрицы: ");
        int n = in.nextInt();
        Matrix m = new Matrix(n);
        m.output();
        System.out.print("Введите столбец для сортировки: ");
        int col_num = in.nextInt();
        m.sort(col_num);
        m.output();
        System.out.print("Введите сдвиг по горизонтали: ");
        int step = in.nextInt();
        m.shift(step,true);
        m.output();
    }
}
class Matrix {
    int[][] matrix;
    int n;
    // Инициализировать случайными числами от -size до size
    public Matrix(int size) {
        n = size;
        matrix = new int[n][n];
        for(int i = 0; i < n; i++){
```

```
        for(int j = 0; j < n; j++){
            matrix[i][j] = (int)(Math.random()*(2*n+1)) - n;
        }
    }
}
// Вывести матрицу в консоль
public void output() {
    System.out.print("Матрица:");
    for(int i = 0; i < n; i++){
        System.out.println();
        for(int j = 0; j < n; j++){
            System.out.print(matrix[i][j]+" ");
        }
    }
    System.out.println();
}
// Отсортировать строки матрицы,
//   чтобы элементы столбца col_num были по возрастанию
public void sort(int col_num){
    int j;
    int[] tmp = new int[n];
    for(int i = 0; i < n; i++){
        j = i;
        for(int k = i; k < n; k++){
            if(matrix[j][col_num] > matrix[k][col_num]){
                j = k;
            }
        }
        tmp = matrix[i];
        matrix[i] = matrix[j];
        matrix[j] = tmp;
    }
}
// Циклический сдвиг вправо/влево (direction = true),
//                   вверх/вниз (direction = false)
public void shift(int step, boolean direction){
```

```
        int[][] new_m = new int[n][n];
        if(direction) {
            for (int i = 0; i < n; i++) {
                for (int j = 0; j < n; j++) {
                    int prev_j = (j-step)%n;
                    if(prev_j < 0){prev_j += n;}
                    new_m[i][j] = matrix[i][prev_j];
                }
            }
        } else{
            for (int i = 0; i < n; i++) {
                for (int j = 0; j < n; j++) {
                    int prev_i = (i-step)%n;
                    if(prev_i < 0){prev_i += n;}
                    new_m[i][j] = matrix[prev_i][j];
                }
            }
        }
        matrix = new_m;
    }
}
```

Результат выполнения программы продемонстрирован на рисунке 2. При этом был введён размер матрицы, номер столбца, по которому выполняется сортировка, и значения сдвига по горизонтали.

При реализации был написан отдельный класс Matrix для инициализации, хранения и обработки матрицы. Внутри класса кроме сортировки строк по элементам выбранного столбца также реализован сдвиг, которому передаётся значение шага сдвига и выбор направления: горизонтальный или вертикальный. Если значение шага имеет отрицательное значение, то сдвиг выполняется влево и вверх соответственно.

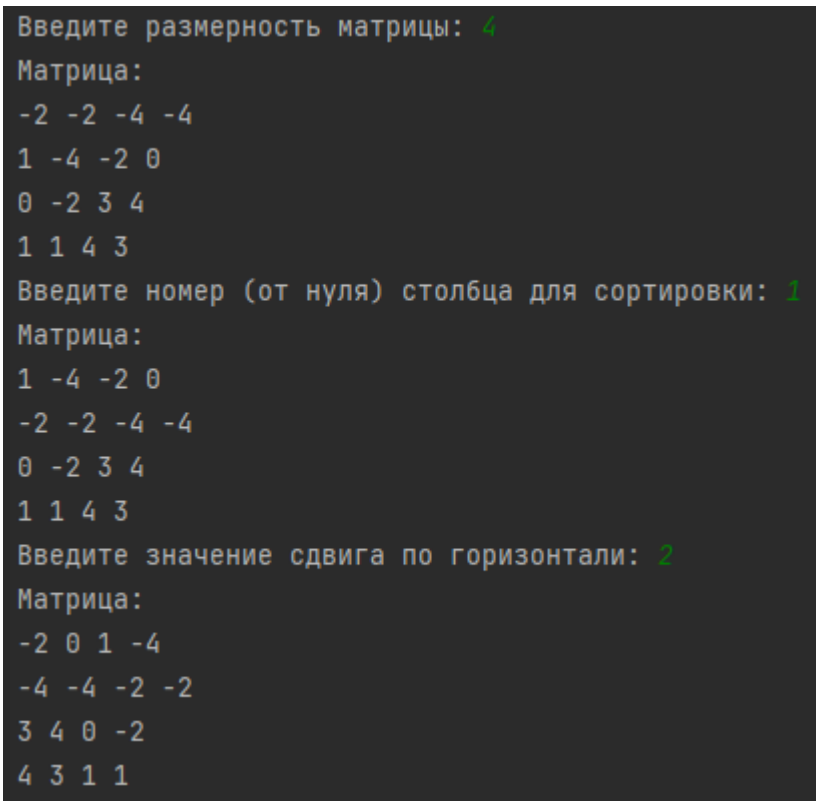


Рисунок 2 – Результат выполнения программы с обработкой массива

**Вывод:** в результате выполнения работы написана программа для упорядочивания и поиска минимальных и максимальных по длине строк, создан класс для хранения и обработки матрицы. Класс позволяет инициализировать матрицу заданного размера случайными числами, выполнять сортировку по заданному столбцу и сдвиг в заданном направлении на выбранный шаг.