|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 **09.04.01/12 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений.**

**Отчет**

**по лабораторной работе № 2**

**Вариант № 3**

**Название:** Арифметические операции

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-23М |  |  | В.А. Гаджиев |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Цель работы**

Изучить арифметических операций на языке программирования Java.

**Задание 1** (Вариант 1, 2; Задание 3, 4):

3. Ввести n строк с консоли. Вывести на консоль те строки, длина которых меньше (больше) средней, а также длину.

4. Ввести n слов с консоли. Найти слово, в котором число различных символов минимально. Если таких слов несколько, найти первое из них.

3. Найти и вывести наибольшее число возрастающих (убывающих) элементов матрицы, идущих подряд.

4. Найти сумму элементов матрицы, расположенных между первым и вторым положительными элементами каждой строки.

Листинг программы:

import java.util.OptionalDouble;  
import java.util.Scanner;  
import java.util.stream.IntStream;  
  
public class Main {  
 public static void main(String[] args) {  
 n\_strings(Integer.parseInt(args[0])); //вариант 1 задача 3  
 count\_various\_symbols(Integer.parseInt(args[0])); //вариант 1 задача 4  
 LIS\_in\_matrix(Integer.parseInt(args[0])); //вариант 2 задача 3  
 sum\_in\_matrix(Integer.parseInt(args[0])); //вариант 2 задача 4  
 }  
  
 public static void n\_strings(int n) {  
  
 Scanner inp = new Scanner(System.in);  
 String[] strings = new String[n];  
 int[] len\_strings = new int[n];  
  
 for (int i = 0; i < n; ++i) {  
 if (inp.hasNextLine()) {  
 strings[i] = inp.nextLine();  
 len\_strings[i] = strings[i].length();  
 }  
 }  
  
 double avg;  
 OptionalDouble aver = IntStream.of(len\_strings).average();  
  
 if (aver.isPresent())  
 avg = aver.getAsDouble();  
 else return;  
  
 System.out.printf("Average is %.1f\n", avg);  
  
 System.out.println("\nAbove average:");  
 for (int i = 0; i < n; ++i) {  
 if (len\_strings[i] > avg) {  
 System.out.printf("string: '%s', len: %d\n", strings[i], len\_strings[i]);  
 }  
 }  
  
 System.out.println("\nBelow average:");  
 for (int i = 0; i < n; ++i) {  
 if (len\_strings[i] < avg) {  
 System.out.printf("string: '%s', len: %d\n", strings[i], len\_strings[i]);  
 }  
 }  
  
 }  
  
 public static void count\_various\_symbols(int n) {  
  
 Scanner inp = new Scanner(System.in);  
 String[] strings = new String[n];  
  
 for (int i = 0; i < n; ++i) {  
 if (inp.hasNextLine()) {  
 strings[i] = inp.nextLine();  
 }  
 }  
  
 int min = (int) strings[0].chars().distinct().count();  
 int index = 0;  
 int tmp;  
  
 for (int i = 1; i < n; ++i) {  
 tmp = (int) strings[i].chars().distinct().count();  
 if (tmp < min) {  
 index = i;  
 min = tmp;  
 }  
 }  
  
 System.out.printf("String: '%s', Count: %d", strings[index], min);  
  
 }  
  
 public static int[][] fill\_in\_matrix(int n) {  
 int[][] matrix = new int[n][n];  
 for(int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 int element = (int) (Math.random() \* (2 \* n + 1)) - n;  
 matrix[i][j] = element;  
 }  
 }  
  
 return matrix;  
 }  
  
 public static void print\_matrix(int[][] matrix) {  
 for(int[] row: matrix) {  
 for (int element: row) {  
 System.out.printf("%4d", element);  
 }  
 System.out.println();  
 }  
 }  
  
 public static int[] matrix\_to\_list(int[][] matrix){  
 int n = matrix.length;  
 int[] list = new int[n\*n];  
 int index\_of\_list = 0;  
 for (int[] ints : matrix) {  
 for (int j = 0; j < n; j++) {  
 list[index\_of\_list] = ints[j];  
 index\_of\_list++;  
 }  
 }  
 return list;  
 }  
  
 public static void LIS\_in\_list(int[] input) {  
 int[] lens = new int[input.length];  
 int max = 0;  
  
 for(int i=1; i<input.length; i++)  
 {  
 if (input[i]>input[i-1])  
 lens[i] = lens[i-1]+1;  
 else lens[i] = 0;  
 max = Math.max(lens[i], max);  
 }  
  
 System.out.println(max + 1);  
 for(int i=0; i<input.length; i++)  
 {  
 if (lens[i] == max)  
 {  
 for(int j = i - lens[i]; j <= i; j++)  
 System.out.print(input[j] + " ");  
 System.out.println();  
 }  
 }  
 }  
  
 public static void LIS\_in\_matrix(int n) {  
 int[][] matrix = fill\_in\_matrix(n);  
 print\_matrix(matrix);  
 int[] sequence = matrix\_to\_list(matrix);  
 LIS\_in\_list(sequence);  
 }  
  
 public static void sum\_in\_matrix(int n) {  
 int[][] matrix = fill\_in\_matrix(n);  
 print\_matrix(matrix);  
 int ind = 0;  
 int sum = 0;  
  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 if(matrix[i][j] > 0) {  
 ind++;  
 } else if (matrix[i][j] <= 0 && ind == 1) {  
 sum += matrix[i][j];  
 }  
 }  
 ind = 0;  
 }  
  
 System.out.println(sum);  
 }  
}

**Вывод**

В ходе выполнения лабораторной работы были изучены арифметические операций на языке программирования Java.