|  |  |
| --- | --- |
| 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.Н. Зыкин |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Цель:** освоить принципы арифметических операций на языке программирования Java.

**Задание 1:** ввести n строк с консоли. Вывести на консоль те строки, длина которых меньше (больше) средней, а также длину.

Код класса Main:

import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Comparator;  
import java.util.Date;  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.in);  
 System.out.print("Введите количество строк: ");  
 int n = in.nextInt();  
 in.nextLine();  
 System.out.println("Введите строки:");  
 ArrayList<String> strings = new ArrayList<>();  
 int sum = 0;  
 for(int i = 0; i < n; i++) {  
 strings.add(in.nextLine());  
 sum += strings.get(i).length();  
 }  
 int avg = sum/n;  
 System.out.println("Средняя длина строк: " + avg);  
 System.out.println("Строки, с длиной больше средней: ");  
 strings.stream().filter(s-> s.length() > avg).forEach(s-> System.out.printf("%s : %d\n", s, s.length()));  
 System.out.println("Строки, с длиной меньше средней: ");  
 strings.stream().filter(s-> s.length() < avg).forEach(s-> System.out.printf("%s - %d\n", s, s.length()));  
 printInfo();  
  
 }  
  
 private static void printInfo() {  
 System.out.printf("-----------------------------\nРазработчик: Зыкин\nДата и время получения задания: 09.02.2024 13:50\nДата и время сдачи задания: %s",  
 new SimpleDateFormat("dd.MM.yyyy HH:mm").format(new Date()));  
 }  
}

Результат работы программы представлен на рисунке 1.

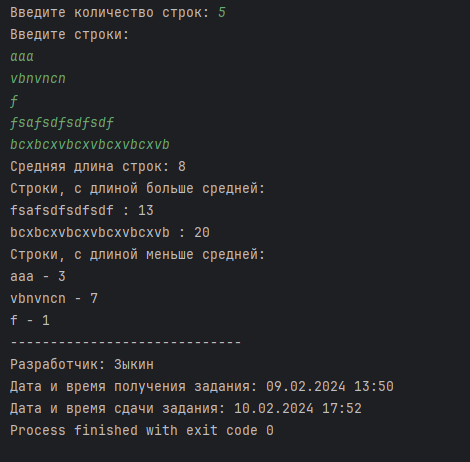


Рисунок 1 – Результат работы программы

**Задание 2:** ввести n слов с консоли. Найти слово, в котором число различных символов минимально. Если таких слов несколько, найти первое из них.

Код класса Main:

import java.text.SimpleDateFormat;  
import java.util.Arrays;  
import java.util.Date;  
import java.util.Scanner;  
import java.util.stream.Collectors;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.in);  
 System.out.print("Введите количество строк: ");  
 int n = in.nextInt();  
 in.nextLine();  
 System.out.println("Введите строки:");  
 String result = "";  
 int minCount = -1;  
 for(int i = 0; i < n; i++) {  
 String newStr = in.nextLine();  
 int count = getUniqueCharCount(newStr);  
 if(i == 0) {  
 minCount = count;  
 result = newStr;  
 } else if(count < minCount) {  
 minCount = count;  
 result = newStr;  
 }  
 }  
 System.out.println("Слово с минимальным количеством различных символов:");  
 System.out.println(result);  
  
 printInfo();  
  
 }  
  
 private static int getUniqueCharCount(String str) {  
 return Arrays.stream(str.split(""))  
 .collect(Collectors.toSet())  
 .size();  
 }  
  
 private static void printInfo() {  
 System.out.printf("-----------------------------\nРазработчик: Зыкин\nДата и время получения задания: 09.02.2024 13:50\nДата и время сдачи задания: %s",  
 new SimpleDateFormat("dd.MM.yyyy HH:mm").format(new Date()));  
 }  
}

Результат работы программы показан на рисунке 2.

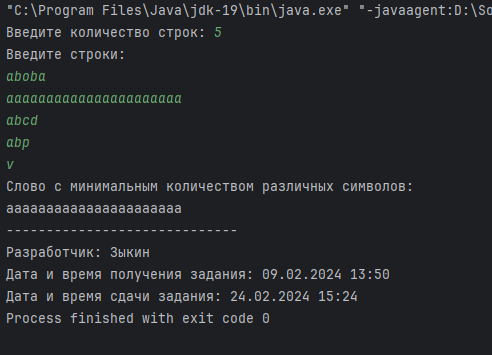


Рисунок 2 – Результат работы программы

**Задание 3:** найти и вывести наибольшее число возрастающих (убывающих) элементов матрицы, идущих подряд.

Код модуля Main:

import java.text.SimpleDateFormat;  
import java.util.\*;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.in);  
 System.out.print("Введите размер матрицы: ");  
 int n = in.nextInt();  
 Random r = new Random();  
 int [][] a = new int[n][n];  
  
 //заполнение массива  
 for(int i = 0; i < n; i++) {  
 for(int j = 0; j < n; j++) {  
 a[i][j] = r.nextInt(-n,n);  
 }  
 }  
  
 System.out.println("Исходный массив:");  
 printArr(a);  
  
 int maxUp = 0;  
 int currCountUp = 1;  
 int maxDown = 0;  
 int currCountDown = 1;  
 int prev = a[0][0]; //первый элемент можно не пропускать, так как нужно строгое убывание/возрастание  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 //возрастание  
 if (a[i][j] > prev) {  
 currCountUp++;  
 } else {  
 if(currCountUp > maxUp) maxUp = currCountUp;  
 currCountUp = 1;  
 }  
  
 //убывание  
 if (a[i][j] < prev) {  
 currCountDown++;  
 } else {  
 if(currCountDown > maxDown) maxDown = currCountDown;  
 currCountDown = 1;  
 }  
 prev = a[i][j];  
 }  
 }  
  
 //отдельно проверяем после последнего элемента, если там была последовательность  
 if(currCountDown > maxDown) maxDown = currCountDown;  
 if(currCountUp > maxUp) maxUp = currCountUp;  
  
 System.out.printf("Наибольшее число возрастающих элементов матриц, идущих подряд: %d.\n", maxUp);  
 System.out.printf("Наибольшее число убывающих элементов матриц, идущих подряд: %d.\n", maxDown);  
   
 printInfo();  
 }  
  
 private static void printArr(int [][] arr) {  
 int rows = arr.length;  
 int cols = arr[0].length;  
  
 for (int i = 0; i < rows; i++) {  
 for (int j = 0; j < cols; j++) {  
 System.out.print(arr[i][j] + " ");  
 }  
 System.out.println();  
 }  
 }  
  
  
 private static void printInfo() {  
 System.out.printf("-----------------------------\nРазработчик: Зыкин\nДата и время получения задания: 09.02.2024 13:50\nДата и время сдачи задания: %s",  
 new SimpleDateFormat("dd.MM.yyyy HH:mm").format(new Date()));  
 }  
}

Результат работы программы представлен на рисунке 3.

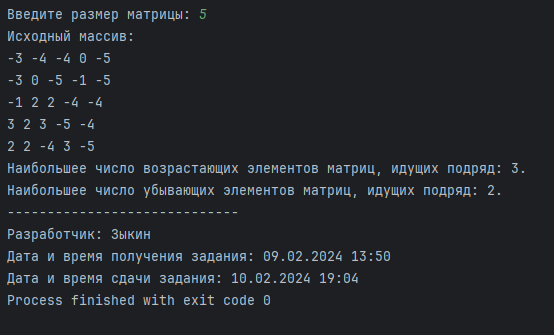
****

Рисунок 3 – Результат работы программы

**Задание 4:** найти сумму элементов матрицы, расположенных между первым и вторым положительными элементами каждой строки.

Код класса Main:

import java.text.SimpleDateFormat;  
import java.util.\*;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.in);  
 System.out.print("Введите размер матрицы: ");  
 int n = in.nextInt();  
 Random r = new Random();  
 int [][] a = new int[n][n];  
  
 //заполнение массива  
 for(int i = 0; i < n; i++) {  
 for(int j = 0; j < n; j++) {  
 a[i][j] = r.nextInt(-n,n);  
 }  
 }  
  
 System.out.println("Исходный массив:");  
 printArr(a);  
  
 int sum = 0;  
 for(int i = 0; i < n; i++) {  
 int indFirst = -1, indSecond = -1;  
 for(int j = 0; j < n; j++) {  
 if(a[i][j] > 0) {  
 if(indFirst == - 1) {  
 indFirst = j;  
 } else {  
 indSecond = j;  
 for(int k = indFirst+1; k < indSecond; k++) {  
 sum += a[i][k];  
 }  
 break; //нет смысла дальше идти по строке  
 }  
 }  
 }  
 }  
   
 System.out.println("Сумма элементов матрицы, расположенных между первым и вторым положительными элементами каждой строки:");  
 System.out.println(sum);  
  
 printInfo();  
 }  
  
 private static void printArr(int [][] arr) {  
 int rows = arr.length;  
 int cols = arr[0].length;  
  
 for(int i = 0; i < rows; i++) {  
 for (int j = 0; j < cols; j++) {  
 System.out.print(arr[i][j] + " ");  
 }  
 System.out.println();  
 }  
 }  
  
  
 private static void printInfo() {  
 System.out.printf("-----------------------------\nРазработчик: Зыкин\nДата и время получения задания: 09.02.2024 13:50\nДата и время сдачи задания: %s",  
 new SimpleDateFormat("dd.MM.yyyy HH:mm").format(new Date()));  
 }  
}

Результат работы программы показан на рисунке 4.

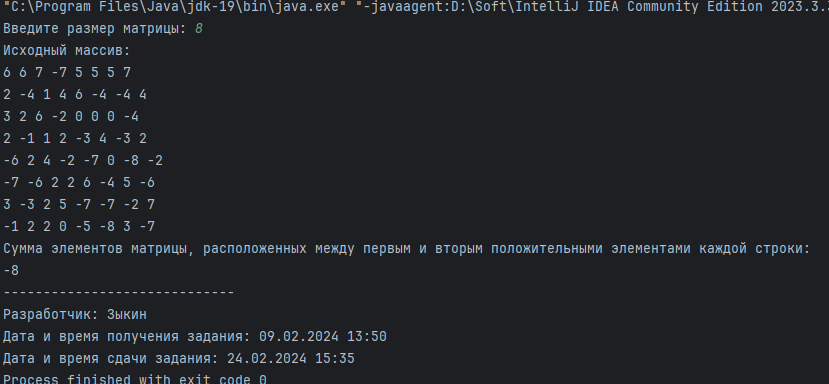


Рисунок 4 – Результат работы программы

**Вывод:** были освоены принципы арифметических операций на языке программирования Java.