Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

|  |
| --- |
| **Отчет** |
| по лабораторной работе №1 по дисциплине:  «Объектно-ориентированное программирование»  «Разработка консольного приложения» |
| наименование темы  Вариант 11 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  |  |  |  |  |  |
|  |  | шифр |  | подпись |  | И.О. Фамилия |
| Проверил |  |  |  |  |  |  |
|  |  |  |  | подпись |  | И.О. Фамилия |

Иркутск 2021

# Оглавление

Содержание

[Оглавление 2](#_Toc93889002)

[1 Постановка задачи 3](#_Toc93889003)

[2 Блок-схема алгоритма 4](#_Toc93889004)

[3 Таблица спецификации 10](#_Toc93889005)

[4 Таблица тестов 11](#_Toc93889006)

[5 Результаты тестирования 11](#_Toc93889007)

[6 Исходный код 13](#_Toc93889008)

[Список использованных источников 15](#_Toc93889009)

# Постановка задачи

Разработать консольное приложение на языке Java, реализующее решение задачи в соответствии с индивидуальным заданием. Исходные данные для задачи вводятся как параметры при запуске программы из командной строки, либо – как значения, инициализирующие переменные. Вывод результатов осуществляется в консоль.

Вариант №11

1. проверка слова в строке на число;
2. найти символ в строке, который встречается наибольшее число раз;
3. удалить из строки все знаки препинания;
4. в матрице, элементами которой являются строки, проверить все ли её элементы являются числами. Результат выдать в виде матрицы с 0 и 1 на соответствующих местах.

# Блок-схема алгоритма



Рисунок 1.1 – Иерархия модулей



Рисунок 1.2 – Блок-схема метода main



Рисунок 1.3 – Блок-схема taskA



Рисунок 1.4 – Блок-схема taskB



Рисунок 1.5 – Блок-схема taskС



Рисунок 1.7 – Блок-схема taskD

# Таблица спецификации

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя  переменной | Описание переменных | Тип  данных |
| **public static void main() – точка входа в программу** | | | |
| — | | | |
| **public static void taskA()** | | | |
| 1 | words | Введённый массив слов | String[] |
| 2 | number | Число | int |
| **public static void taskB()** | | | |
| 1 | text | Введённый текст | String |
| 2 | chars | Массив символов | char[] |
| 3 | maxCountChar | Символ, встречающийся наибольшее количество раз | char |
| 4 | maxCount | Счётчик одинаковых символов | int |
| 5 | tempChar | Временный символ | char |
| 6 | tempCount | Временный счётчик одинаковых символов | int |
| **public static void taskС()** | | | |
| 1 | text | Введённый текст | String |
| 2 | result | Результат | String |
| **public static void taskD()** | | | |
| 1 | matrixWords | Введённая матрица слов | String[][] |
| 2 | matrixResult | Результирующая матрица | int[][] |

# Таблица тестов

|  |  |  |
| --- | --- | --- |
| **№** | **Входные данные** | **Данные на выходе** |
| 1 | asda 12 vfd 1h df1 101 | 12 является числом  101 является числом |
| 2 | adsa[pvsdfvmsdn | [aadddfmnpsssvv  В строке 3 символов d |
| 3 | dsadsad....asd,,ngbg::a;;sd | dsadsadasdngbgsd |
| 4 | 15 fv 1 0 s  1 se 123 1g  j1 0 | [1, 0, 1, 1, 0]  [1, 0, 1, 0]  [0, 1] |

# Результаты тестирования

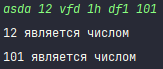
****

Рисунок 2.1 – Результат теста 1

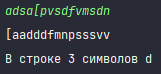


Рисунок 2.2 – Результат теста 2



Рисунок 2.3 – Результат теста 3

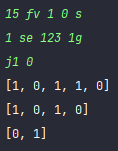


Рисунок 2.4 – Результат теста 4

# Исходный код

package com.company;  
  
import java.util.Arrays;  
import java.util.Scanner;  
  
public class Main {  
 public static Scanner *scanner* = new Scanner(System.*in*);  
  
 public static void main(String[] *args*) {  
 *taskA*();  
 System.*out*.println();  
 *taskB*();  
 System.*out*.println();  
 *taskC*();  
 System.*out*.println();  
 *taskD*();  
 }  
  
 public static void taskA() {  
 *//слова разделяются пробелом или запятой* String[] words = *scanner*.nextLine().split("[ ,]");  
 for (String word : words) {  
 try {  
 int number = Integer.*parseInt*(word);  
 System.*out*.printf("%s является числом\n", number);  
 } catch (NumberFormatException *ignored*) {  
 }  
 }  
 }  
  
 public static void taskB() {  
 char[] chars = *scanner*.nextLine().toCharArray();  
 Arrays.*sort*(chars);  
 String text = new String(chars);  
 System.*out*.println(text);  
  
 char maxCountChar = text.charAt(0);  
 int maxCount = text.lastIndexOf(maxCountChar) + 1;  
  
 for (int i = text.lastIndexOf(maxCountChar) + 1; i < text.length(); i++) {  
 char tempChar = text.charAt(i);  
 int tempCount = text.lastIndexOf(tempChar) - text.indexOf(tempChar) + 1;  
 if (maxCount < tempCount) {  
 maxCountChar = tempChar;  
 maxCount = tempCount;  
 }  
  
 i = text.lastIndexOf(tempChar);  
 }  
  
 System.*out*.printf("В строке %s символов %s\n", maxCount, maxCountChar);  
 }  
  
 public static void taskC() {  
 String text = *scanner*.nextLine();  
 String result = text.replaceAll("[.,:;?!-]", "");  
 System.*out*.println(result);  
 }  
  
 public static void taskD() {  
 String[][] matrixWords = new String[3][];  
 int[][] matrixResult = new int[3][];  
  
 *//строка разделяется на слова через пробел* for (int i = 0; i < matrixWords.length; i++) {  
 matrixWords[i] = *scanner*.nextLine().split(" ");  
 }  
  
 for (int i = 0; i < matrixWords.length; i++) {  
 matrixResult[i] = new int[matrixWords[i].length];  
 for (int j = 0; j < matrixWords[i].length; j++) {  
 try {  
 Integer.*parseInt*(matrixWords[i][j]);  
 matrixResult[i][j] = 1;  
 } catch (NumberFormatException *ignored*) {  
 }  
 }  
 }  
  
 for (int[] row : matrixResult) {  
 System.*out*.println(Arrays.*toString*(row));  
 }  
 }  
}

# Список использованных источников

1. Объектно-ориентированное программирование. Метод. указания по выполнению лабораторных работ / сост.: В.Л. Аршинский. – Иркутск : Изд-во ИРНИТУ, 2017. – 24 c.