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 №2 по дисциплине:  «Объектно-ориентированное программирование»  «Разработка классов и их использование» |
| наименование темы  Вариант 11 |

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

Иркутск 2021

# Оглавление

Содержание

[Оглавление 2](#_Toc94308659)

[1 Постановка задачи 3](#_Toc94308660)

[1 Структура классов 4](#_Toc94308661)

[2 Таблица спецификации 5](#_Toc94308662)

[3 Таблица тестов 7](#_Toc94308663)

[4 Результаты тестирования 7](#_Toc94308664)

[5 Исходный код 8](#_Toc94308665)

[Список использованных источников 12](#_Toc94308666)

# Постановка задачи

Провести объектную декомпозицию задачи из индивидуального задания и разработать класс, содержащий соответствующие поля для хранения необходимых данных и методы, обеспечивающие достаточную для решения задачи функциональность класса. Создание объекта класса, решающего задание и вызов его методов, должны осуществляться из главного класса. Ввод и вывод данных осуществляется как в предыдущей лабораторной работе, для передачи исходных данных экземпляру класса, решающему задачу, должны быть разработаны соответствующие методы.

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.

# Структура классов



Рисунок 1 – Структура классов

Описание классов

1. Класс Main – главный класс, содержит метод main, являющийся точкой запуска прораммы
2. Класс Text – класс, содержащий переменную text типа String и методы для выполнения задания A, B и C

Состояния объектов данного класса определяется полями (атрибутами):

* String text

Поведение объектов данного класса определяется методами:

* String getText()
* void wordsIsNumber()
* String charMaxCount()
* String removeAllPunctuation()
* Text clone()

Начальное состояние объектов определяются с помощью конструкторов:

* Text()
* Text(String text)
* Text(Text other)

1. Класс WordMatrix – класс, содержащий двухмерный массив слов matrixWords типа String, двухмерный массив matrixResult типа int, и метод для выполнения задания D

Состояния объектов данного класса определяется полями (атрибутами):

* String[][] matrixWords
* int[][] matrixResult

Поведение объектов данного класса определяется методами:

* void printMatrixWords()
* void printMatrixResult()
* void checkWordsOnNumber()
* WordMatrix clone()

Начальное состояние объектов определяются с помощью конструкторов:

* WordMatrix()
* WordMatrix(String[][] matrixWords)
* WordMatrix(WordMatrix other)

# Таблица спецификации

Таблица 1 – Описание полей классов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Имя | Назначение | Тип данных | Тип доступа | ОДЗ |
| Класс Text | | | | | |
| 1 | text | текст | String | private | [a..Z],[а..Я],[,. ],[0..9] |
| Класс WordMatrix | | | | | |
| 2 | matrixWords | Матрица слов | String[][] | private | [a..Z],[а..Я],[,. ],[0..9] (для элементов) |
| 3 | matrixResult | Результирующая матрица | int[][] | private | [-2147483648, 2147483647]  (для элементов) |

Таблица 2 – Описание методов класса Text

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Имя | Назначение | Тип данных | Тип доступа | Параметры |
| 1 | getText | геттер text | String | public | — |
| 2 | wordsIsNumber | Определение какие слова являются числами | void | private | — |
| 3 | charMaxCount | Определение символа, который встречается наибольшее число раз | String | public | — |
| 4 | removeAllPunctuation | Удаление всех знаков препинания из строки | String | private | — |

Таблица 3 – Описание методов класса WordMatrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Имя | Назначение | Тип данных | Тип доступа | Параметры |
| 1 | printMatrixWords | Печать матрицы слов | void | public | — |
| 2 | printMatrixResult | Печать результирующей матрицы | void | public | — |
| 3 | checkWordsOnNumber | Проверка слов на число в матрице | void | public | — |

# 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

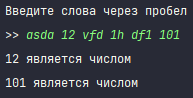
****

Рисунок 4.1 – Тест 1

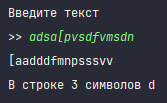


Рисунок 4.2 – Тест 2

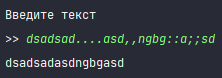


Рисунок 4.3 – Тест 3

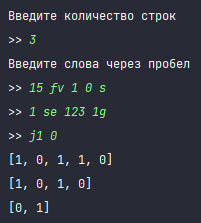


Рисунок 4.4 – Тест 4

# Исходный код

класс Main

public class Main {  
 public static Scanner *scanner* = new Scanner(System.*in*);  
  
 public static void main(String[] *args*) {  
 System.*out*.print("Введите слова через пробел\n>> ");  
 Text text = new Text(*scanner*.nextLine());  
 text.wordsIsNumber();  
 System.*out*.println();  
  
 System.*out*.print("Введите текст\n>> ");  
 text = new Text(*scanner*.nextLine());  
 System.*out*.println(text.charMaxCount());  
  
 System.*out*.print("Введите текст\n>> ");  
 text = new Text(*scanner*.nextLine());  
 System.*out*.println(text.removeAllPunctuation());  
 System.*out*.println();  
  
 WordMatrix wordMatrix = new WordMatrix(*inputWordMatrix*());  
 wordMatrix.checkWordsOnNumber();  
 wordMatrix.printMatrixResult();  
 }  
  
 private static String[][] inputWordMatrix() {  
 System.*out*.print("Введите количество строк\n>> ");  
 String[][] matrix = new String[*scanner*.nextInt()][];  
  
 *scanner* = new Scanner(System.*in*);  
 System.*out*.println("Введите слова через пробел");  
 *//строка разделяется на слова через пробел* for (int i = 0; i < matrix.length; i++) {  
 System.*out*.print(">> ");  
 matrix[i] = *scanner*.nextLine().split(" ");  
 }  
 return matrix;  
 }  
}

класс Text

public class Text implements *Cloneable* {  
 private String text;  
  
 public Text() {  
 text = "пустой текст";  
 }  
  
 public Text(String *text*) {  
 this.text = *text*;  
 }  
  
 public Text(Text *other*) {  
 this.text = *other*.text;  
 }  
  
 public String getText() {  
 return text;  
 }  
  
 public void wordsIsNumber() {  
 *//слова разделяются пробелом или запятой* String[] words = text.split(" ");  
 for (String word : words) {  
 try {  
 int number = Integer.*parseInt*(word);  
 System.*out*.printf("%s является числом\n", number);  
 } catch (NumberFormatException *ignored*) {  
 }  
 }  
 }  
  
 public String charMaxCount() {  
 char[] chars = text.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);  
 }  
  
 return String.*format*("В строке %s символов %s\n", maxCount, maxCountChar);  
 }  
  
 public String removeAllPunctuation() {  
 return text.replaceAll("[.,:;?!-]", "");  
 }  
  
 @Override  
 public Text clone() throws CloneNotSupportedException {  
 return (Text) super.clone();  
 }  
}

класс WordMatrix

public class WordMatrix implements *Cloneable* {  
 private String[][] matrixWords;  
 private int[][] matrixResult;  
  
 public WordMatrix() {  
 matrixWords = new String[3][1];  
 matrixWords[0][0] = "вы";  
 matrixWords[1][0] = "не ввели";  
 matrixWords[2][0] = "предложение";  
 matrixResult = new int[3][1];  
 }  
  
 public WordMatrix(String[][] *matrixWords*) {  
 this.matrixWords = *matrixWords*;  
 matrixResult = new int[*matrixWords*.length][];  
 for (int i = 0; i < *matrixWords*.length; i++) {  
 matrixResult[i] = new int[*matrixWords*[i].length];  
 }  
 }  
  
 public WordMatrix(WordMatrix *other*) {  
 this.matrixWords = *other*.matrixWords;  
 this.matrixResult = *other*.matrixResult;  
 }  
  
 public void printMatrixWords() {  
 for (String[] row : matrixWords) {  
 System.*out*.println(Arrays.*toString*(row));  
 }  
 }  
  
 public void printMatrixResult() {  
 for (int[] row : matrixResult) {  
 System.*out*.println(Arrays.*toString*(row));  
 }  
 }  
  
 public void checkWordsOnNumber() {  
 for (int i = 0; i < matrixWords.length; i++) {  
 for (int j = 0; j < matrixWords[i].length; j++) {  
 try {  
 Integer.*parseInt*(matrixWords[i][j]);  
 matrixResult[i][j] = 1;  
 } catch (NumberFormatException *ignored*) {  
 }  
 }  
 }  
 }  
  
 @Override  
 public WordMatrix clone() throws CloneNotSupportedException {  
 return (WordMatrix) super.clone();  
 }  
}

# Список использованных источников

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