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 |
| Кафедра вычислительной техники |
| наименование кафедры |

|  |
| --- |
| **Отчет** |
| по лабораторной работе №2 по дисциплине «Разработка классов и их использование» |
| наименование темы  Вариант №4 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | АСУб-19-1 |  |  |  | Аласаев П.М |
|  |  | шифр |  | подпись |  | И.О. Фамилия |
| Проверил |  |  |  |  |  | Маланов Т.В. |
|  |  |  |  | подпись |  | И.О. Фамилия |
| Работа защищена с оценкой | | | |  | | |

Иркутск 2020 г.

**Содержание**

[1 Постановка задачи 3](#_Toc53221512)

[2 Описание структуры классов 4](#_Toc53221513)

[3 Описание методов классов и спецификация локальных переменных 5](#_Toc53221514)

[4 Таблица тестов 8](#_Toc53221515)

[5 Результаты тестирования 9](#_Toc53221516)

[6 Исходный код 10](#_Toc53221518)

[Список использованных источников 13](#_Toc53221519)

1 Постановка задачи

1. Найдите символ в строке, который встречается наибольшее число раз.
2. Найдите самое длинное слово в предложении.
3. Удалите в предложении каждое третье слово.

Провести объектную декомпозицию задачи из индивидуального задания и разработать класс, содержащий соответствующие поля для хранения не обходимых 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.

2 Описание структуры классов

Программа разбита на два класса. Один главный класс, в котором содержится метод main являющийся точкой входа в программу, класс Strill в котором содержатся необходимые методы реализующие функционал для обработки строки, и класс Array содержащий функционал для обработки массива из строк

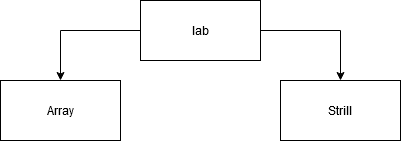


Рисунок 1 – Структура классов

1. *Чем является объект класса Array?* Array -Это экземпляр класса Array
2. *Чем является объект класса Strill?* Strill - Это экземпляр класса Strill
3. *Какими атрибутами определяется состояние объекта класса Array?* Array – private String[] words
4. *Какими атрибутами определяется состояние объекта класса Strill?* Strill – private String string
5. *Какие методы изменяют состояние объекта?* Cеттеры(set): setWords, setString.
6. *В чем смысл инкапсуляции и как реализован этот принцип в Вашей работе?* Инкапсуляция признак разганичения доступа. В моей работе поле string[] words объявлем с мадификатором доступа private, что обозначает, что он доступен только в коде класса Array.
7. *Для чего нужны конструкторы?* Конструкторы нужны для задания начального значения полям (атрибутом) класса при создании объекта.
8. *Что будет при создании объекта какого-либо, если разработчик этого класса не создаст ни одного конструктора?* Всем полям задается начальное значение. Например, для int значение будет равно 0.

3 Описание методов классов и спецификация локальных переменных

Таблица 1 – Описание методов класса Main

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Назначение | Возвращаемый тип | Модификатор | Аргументы |
| main | Точка входа в программу. Создает объекты класса задач и выполняет их методы | - | Public static | - |

Таблица 2 – Описание метода класса Strill

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Назначение | Возвращаемый тип | Модификатор | Аргументы |
| Strill | Конструктор по умолчанию. Устанавливает поля объекта значениями по умолчанию | Strill | Public | - |
| Strill | Перегрузка конструктора, принимает аргументы для заполнения полей объекта переданными значениями | Strill | Public | String |
| Strill | Конструктор копирования. Создает копию объекта переданного в аргументе | Strill | public | Strill |
| setString | Метод для установки значения | void | public | String |
| getString | Метод для получения поля | String | public | - |

Продолжение таблицы 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Назначение | Возвращаемый тип | Модификатор | Аргументы |
| mostLengthWord | Метод для получения самого длинного слова | String | public | - |
|  |  | String | public | - |

Таблица 3 – Описание методов класса Array

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Назначение | Возвращаемый тип | Модификатор | Аргументы |
| Array | Перегрузка конструктора, принимает аргументы для заполнения полей объекта переданными значениями | Array | public | String[] |
| Array | Конструктор по умолчанию. Устанавливает поля объекта значениями по умолчанию | Array | Public | - |
| Array | Конструктор копирования. Возвращает копию объекта переданного в аргументе | Array | public | Array |
| setWords | Метод для установки свойства | Void | Public | String[] |
| getWords | Возвращает массив | String[] | Public | - |
| delete | Удаляет каждое третье слово в предложении | void | Public | - |

Таблица 4 – Описание полей класса Strill

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Назначение | Тип | Модификаторы |
| String | Хранит строку над которой будут производиться различные действия | string | private |

Таблица 5 – Описание полей класс Array

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Назначение | Тип | Модификаторы |
| Words | Хранит строку над которой будут производиться различные действия | String[] | private |

Таблица 6 – Спецификация локальных переменных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Имя | Назначение | Тип | Одз |
| 1 | iChar | Строка | char | [0;2\* 109] |
| 2 | isFirst | Проверка | Boolean | [0;2\* 109] |
| 3 | max | Вывод символа, который встречается больше всего раз | string | [0;2\* 109] |
| 4 | max | Вывод самого длинного слова в предложении | String | [0;2\* 109] |
| 5 | mas | Строка | String | [0;2\* 109] |
| 6 | works | Строка | String | [0;2\* 109] |
| 7 | answer | Строка вывода предложения без третьего слова | String | [0;2\* 109] |
| 8 | vvod | Ввод строки | String | [0;2\* 109] |
| 9 | aline | Вывод строки a | String | [0;2\* 109] |
| 10 | bline | Вывод строки b | String | [0;2\* 109] |
| 11 | cline | Вывод строки с | String | [0;2\* 109] |
| 12 | count | Счет символов | int | [0;2\* 109] |

4 Таблица тестов

Таблица 7 - Тесты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Входные данные | | Выходные данные | |
| Проверка на копирование объекта | | | | |
| 1 | - | | - | |
| Задание A | | | | |
| 1 | adjkbhas ajcnvsdfi sdfnvids | | s | |
| 2 | aaa cccc kk | | c | |
| 3 | There is my house | | e | |
| Задание B | | | | |
| 4 | vkb aosdnf sdfibanciw | | sdfibanciw | |
| 5 | aaaa ccccc sss | | ccccc | |
| 6 | I have to go to the university | | university | |
| Задание C | | | | |
| 7 | aafdaksn saifmia | | - | |
| 8 | aaaa xxxx falsdnf aaaaa xxxx amddasdf | | aaaa xxxx aaaaa xxxx | |
| 9 | There is the house where my family lives | | There is house where family lives | |
| Проверка конструктора по умолчанию | | | | |
| Задание A | | - | | t |
| Задание B | | - | | default |
| Задание С | | - | | default text |

*Добавить в таблицу тестов тесты, связанные с проверкой конструкторов. В качестве входных данных должен быть участок кода Main, который отвечает за создание констркуторов*

5 Результаты тестирования



Рисунок 2 – Тест 1, задание A



Рисунок 3 – Тест 2, задание A



Рисунок 4 – Тест 3, задание A



Рисунок 5 – Тест 1, задание B



Рисунок 6 – Тест 2, задание B



Рисунок 7 – Тест 3, задание B



Рисунок 8 – Тест 1, задание C



Рисунок 9 – Тест 2, задание C

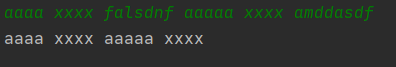


Рисунок 10 – Тест 3, задание C

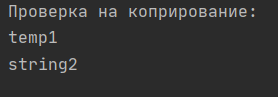


Рисунок 11 – Проверка на копирование

6 Исходный код

1. Класс Main

*Сделать текст черным на белом фоне*

package lab;

import java.util.Scanner;

public class lab {

public static void main(String[] args) {

Scanner vvod = new Scanner(System.in);

String aline = vvod.nextLine();

Strill line = new Strill(aline);

System.out.println(line.maxContainsChar());

String bline = vvod.nextLine();

line.setString(bline);

System.out.println(line.mostLengthWord());

String cline = vvod.nextLine();

Array words = new Array(cline.split(" "));

System.out.println(words.delete());

// Проверка копирования объекта

System.out.println("Проверка на копирование:");

Array ad = new Array("temp1 string2".split(" "));

Array sd = new Array(ad);

ad.setWords("some text 3".split(" "));

String[] wordsSd = sd.getWords();

for (String word :

wordsSd) {

System.out.println(word + " ");

}

}

}

2)Класс Strill

package lab;

import java.util.HashMap;

import java.util.Map;

public class Strill {

private String string;

public Strill() {

this.string = "default text";

}

public Strill(String string) {

this.string = string;

}

public Strill(Strill clone){

this.string = clone.getString();

}

public void setString(String string) {

this.string = string;

}

public String getString() {

return string;

}

public String mostLengthWord(){

String[] mas = string.split(" ");

String max = mas[0];

for (String temp :

mas) {

if (max.length() < temp.length()) max = temp;

}

return max;

}

public char maxContainsChar() {

Map<Character, Integer> map = new HashMap<>();

for (int i = 0; i < string.length(); i++) {

char iChar = string.charAt(i);

if (iChar == ' ') continue;

// получаем число одинаовых букв, если буква встречается первый раз возращаем 0

int count = map.getOrDefault(iChar, 0);

map.put(iChar, ++count);

}

boolean isFirst = true;

Map.Entry<Character, Integer> max = null;

for (Map.Entry<Character, Integer> count : map.entrySet()) {

if (isFirst) {

max = count;

isFirst = false;

}

if (max.getValue() < count.getValue()) max = count;

}

assert max != null;

return max.getKey();

}

}

3)Класс Array

package lab;

public class Array {

private String[] words;

public Array(){

this.words = "defaunt text".split(" ");

}

public Array(String[] words){

this.words = words;

}

public Array(Array clone) {

String[] originWords = clone.getWords();

String[] newWords = new String[originWords.length];

for (int i = 0; i < originWords.length; i++) {

newWords[i] = originWords[i];

}

this.words = newWords;

}

public void setWords(String[] words){

this.words = words;

}

public String[] getWords(){

return words;

}

public String delete(){

StringBuilder answer = new StringBuilder(words[0]).append(" ");

for (int i = 1; i < words.length; i++) {

if ((i + 1) % 3 == 0) continue;

answer.append(words[i]).append(" ");

}

return answer.toString();

}

Список использованных источников

1. Методические указания по выполнению лабораторных работ / сост.: В.Л. Аршинский. – Иркутск : Изд-во ИРНИТУ, 2017. – 24 c.
2. Файн Я. – Программирование на Java для детей, родителей, дедушек и бабушек, 2011