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

Центр программной инженерии

ОТЧЕТ

к лабораторной работе №2 по дисциплине:

|  |
| --- |
| «Объектно-ориентированное программирование» |
| Разработка классов и их использование |

наименование темы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы: | ИСТб-19-2 |  |  |  | Карбушев И.В. |
|  | шифр группы |  | подпись |  | Фамилия ИО |
| Проверил: | доцент |  |  |  | Маланова Т.В. |
|  | должность |  | подпись |  | Фамилия ИО |

Иркутск. 2020 г.

Оглавление

[Оглавление 2](#_Toc51891759)

[1 Постановка задачи 3](#_Toc51891760)

[2 Структура классов 4](#_Toc51891761)

[3 Описание методов классов и спецификация локальных переменных 5](#_Toc51891762)

[4 Таблицы тестов 7](#_Toc51891763)

[5 Результаты тестирования 9](#_Toc51891764)

[6 Код программы 10](#_Toc51891765)

[Список использованных источников 14](#_Toc51891766)

1 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 в экземпляр класса решающего задачу должны быть разработаны соответствующие методы.

Вариант №7

a. Отсортируйте слова в массиве по третьей букве, если слово меньше трех букв, то по последней.

b. Определите, входит ли строка, указанная в командной строке, при запуске программы, в предложение.

c. Удалите в предложении каждое третье слово.

2 Структура классов

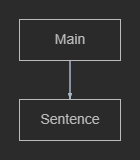


Рисунок 1.1 – Структура классов

3 Описание методов классов и спецификация локальных переменных

Таблица 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Имя переменной | Описание переменных | | Тип данных | |
| **Class Main** | | | | | |
| **Public static void main()** | | | | | |
| 1 | in | Объект класса Scanner для ввода данных | | Scanner | |
| 2 | inputString | Входные данные для работы с 1ой задачей | | String[] | |
| 3 | s1,s2,s3 | Объекты класса StringArraySort с разными входными данными | | StringArraySort | |
| 4 | sentence | Входные данные для работы с 2ой задачей | | String | |
| 5 | subSentence | Входные данные для работы с 2ой задачей | | String | |
| 6 | c1,c2,c3 | Объекты класса StringContains с разными входными данными | | StringContains | |
| 7 | edit | Входные данные для работы с 3ей задачей | | String[] | |
| 8 | e1,e2,e3 | Объекты класса EditString с разными входными данными | | EditString | |
| **Class Sentence** | | | | | |
| 9 | stringArray | Поле класса для работы с методами | | String[] | |
| **Private void swapValues(int ind1, int ind2)** | | | | | |
| 10 | temp | Временная переменная для обмена местами значений массива | | String | |
| 11 | ind1, ind2 | Индексы массива, значения которых нужно поменять местами | | int | |
| **Public void sort()** | | | | | |
| 12 | i,j | Переменные в циклах | | int | |
| 13 | str2, str1 | Текущий элемент массива и первый не отсортированный элемент массива | | String | |
| **Public Boolean ifContains(String subStr)** | | | | | |
| 14 | i | Переменная в цикле | | int | |
| 15 | count | Переменная-счетчик для проверки, содержится ли подстрока subSentence в строке sentence | | int | |
| 16 | sent | Строка из поля класса stringArray для удобной реализации метода | | String | |
| 17 | sentence | Глобальная переменная класса для удобной работы с методами | | String[] | |
| **Public void delWords(int number)** | | | | | |
| 18 | count | | Счетчик слов, чтобы удалять каждое 3 слово в строке | | int |
| 19 | number | | Номер удаляемых слов в предложении | | int |

|  |  |  |  |
| --- | --- | --- | --- |
| @Override  public String toString() | | | |
| 20 | out | Строка из объекта класса Sentence | String |

Продолжение таблицы 1

**Class Sentence:**

1. Public Sentence() – конструктор без любых входных данных
2. Public Sentence(String[] stringArray) – конструктор с входным массивом
3. Public Sentence(Sentence sent) – конструктор копирования с входным объектом этого же класса
4. Private void swapValues(int ind1, int ind2) – метод для того, чтобы менять значения массива местами
5. Public void sort() – главный метод класса, отвечает за сортировку массива по правилам из задания 1
6. Public boolean ifContains(String subStr) – определяет, содержится ли подстрока subStr в строке sentence
7. Public void delWords(int number) – удаляет каждое number-е слово в строке
8. @Override Public String toString() – геттер для переменной sentence

4 Таблицы тестов

1 задача

|  |  |  |
| --- | --- | --- |
| **№** | **Входные данные** | **Данные на выходе** |
| 1 | I am keen on programming on java //стандартные данные  Raz dva tri chetire aa //входные данные  I am keen on programming on java //данные с конструктора копирования | I keen am on on programming java  dva pyat chetire shest tri raz  I keen am on on programming java |

2 задача

|  |  |  |
| --- | --- | --- |
| **№** | **Входные данные** | **Данные на выходе** |
| 1 | I am keen on programming on java  dva t //стандартные данные  raz dva tri chetire pyat shest  dva, t //входные данные  I am keen on programming on java  dva t //данные с конструктора копирования | Нет, "I am keen on programming on java " не содержит "dva t"  Да, "raz dva tri chetire pyat shest " содержит "dva t"  Нет, "I am keen on programming on java " не содержит "dva t" |

3 задача

|  |  |  |
| --- | --- | --- |
| **№** | **Входные данные** | **Данные на выходе** |
| 1 | I keen am on on programming java  dva pyat chetire shest tri raz  I keen am on on programming java  //все данные с 1 задачи | I keen on on java  dva pyat shest tri  I keen on on java |

5 Результаты тестирования

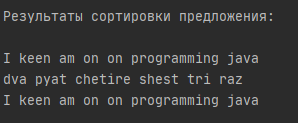


Рисунок 2.1 – Тест 1.1

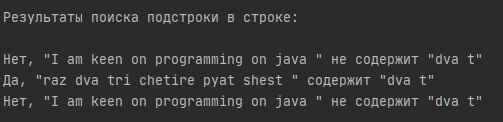


Рисунок 2.2 – Тест 2.1

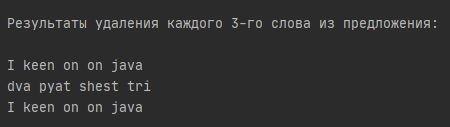


Рисунок 2.3 – Тест 3.1

6 Код программы

**Код класса Main**

package com.company;  
  
import java.util.Scanner;  
  
class Main{  
 static Scanner *in* = new Scanner(System.*in*);  
  
 public static void main(String[] args) {  
 System.*out*.println("Введите предложение для последующих операций над ним");  
 String[] inputData = *in*.nextLine().split(" ");  
 System.*out*.println("Введите искомую подстроку для 2 задачи");  
 String subStr = *in*.nextLine();  
  
 Sentence sent1 = new Sentence();  
 Sentence sent2 = new Sentence(inputData);  
 Sentence sent3 = new Sentence(sent1);  
  
 //решение второй задачи  
 System.*out*.println("\nРезультаты поиска подстроки в строке:\n");  
 if (sent1.ifContains(subStr)) System.*out*.println("Да, \"" + sent1.toString() + "\"" + " содержит \"" + subStr + "\"");  
 else System.*out*.println("Нет, \"" + sent1.toString() + "\"" + " не содержит \"" + subStr + "\"");  
 if (sent2.ifContains(subStr)) System.*out*.println("Да, \"" + sent2.toString() + "\"" + " содержит \"" + subStr + "\"");  
 else System.*out*.println("Нет, \"" + sent2.toString() + "\"" + " не содержит \"" + subStr + "\"");  
 if (sent3.ifContains(subStr)) System.*out*.println("Да, \"" + sent3.toString() + "\"" + " содержит \"" + subStr + "\"");  
 else System.*out*.println("Нет, \"" + sent3.toString() + "\"" + " не содержит \"" + subStr + "\"");  
  
 //решение первой задачи  
 System.*out*.println("\nРезультаты сортировки предложения:\n");  
 sent1.sort();  
 sent2.sort();  
 sent3.sort();  
  
 System.*out*.println(sent1.toString());  
 System.*out*.println(sent2.toString());  
 System.*out*.println(sent3.toString());  
  
 //решение третьей задачи  
 System.*out*.println("\nРезультаты удаления каждого 3-го слова из предложения:\n");  
 sent1.delWords(3);  
 sent2.delWords(3);  
 sent3.delWords(3);  
  
 System.*out*.println(sent1.toString());  
 System.*out*.println(sent2.toString());  
 System.*out*.println(sent3.toString());  
 }  
}

**Код класса Sentence**

package com.company;  
  
public class Sentence {  
  
 private String[] stringArray;  
  
 public Sentence() { //базовый конструктор  
 this.stringArray = new String[]{"I", "am", "keen", "on", "programming", "on", "java"};  
 }  
  
 public Sentence(String[] stringArray) { //конструктор с входными данными  
 this.stringArray = stringArray;  
 }  
  
 public Sentence(Sentence sent){  
 this.stringArray = sent.stringArray;  
 }  
  
 public void swapValues(int ind1, int ind2) { //обмен переменных массива по индексам  
 String temp = stringArray[ind2];  
 stringArray[ind2] = stringArray[ind1];  
 stringArray[ind1] = temp;  
 }  
  
 public void sort() { //метод сортировки массива  
 // по третьей букве, если слово меньше трех букв, то по последней.  
 for (int i = 0; i < stringArray.length; i++)  
 {  
 for (int j = i; j < stringArray.length; j++)  
 {  
 String str1 = stringArray[i]; //слова из массива строк, чтобы  
 String str2 = stringArray[j]; //удобнее обращаться к их символам  
  
 if (str1.length() >= 3) //если 1 слово длиннее или равно 3 буквам  
 {  
 //если 2 слово длиннее или равно 3 буквам и 3-ья буква 1-го слово больше, чем 2-ая  
 if (str2.length() >= 3 && str1.charAt(2) > str2.charAt(2)) swapValues(i, j);  
 //иначе если 3-я буква 1-го слова больше последней буквы 2-го слова  
 else if (str2.length() < 3 && str1.charAt(2) > str2.charAt(str2.length() - 1)) swapValues(i ,j);  
 }  
 else  
 {  
 //аналогично, как выше  
 if (str2.length() >= 3 && str1.charAt(str1.length() - 1) > str2.charAt(2)) swapValues(i, j);  
 else if (str2.length() < 3 && str1.charAt(str1.length() - 1) > str2.charAt(str2.length() - 1)) swapValues(i ,j);  
 }  
 }  
 }  
 }  
  
 public boolean ifContains(String subStr) //метод на проверку, содержит ли строка подстроку  
 {  
 int count = 0; //счетчик для того, чтобы считать текущий индекс подстроки  
 String sent = new Sentence(stringArray).toString(); //получение строки из объекта, потому что здесь нужна именно строка, а не массив строк  
 for (int i = 0; i < sent.length(); i++)  
 {  
 if (sent.charAt(i) == subStr.charAt(count)) //если текущий символ строки совпадает с символом подстроки  
 {  
 count++;  
 //если подстрока полностью содержится в строке, т.е. количество совпадающих символов в строке и подстроке равно длине подстроки  
 if (count == subStr.length())  
 {  
 return true;  
 }  
 }  
 else count = 0; //иначе подстрока не содержится полностью и индекс подстроки обнуляется  
 }  
 return false;  
 }  
  
 public void delWords(int number) //удаление каждого number-го слова в массиве  
 {  
 int count = 1;  
 for (int i = 0; i < stringArray.length; i++){  
 if (count++ % number == 0) stringArray[i] = "";  
 }  
 }  
  
 @Override //переопределение метода toString() для текущего класса  
 public String toString() {  
 String out = "";  
 for (String str : stringArray) {  
 out += str;  
 if (!str.equals("")) out += " ";  
 }  
 return out;  
 }  
}

Список использованных источников

1. Объектно-ориентированное программирование. Метод. указания по

выполнению лабораторных работ / сост.: В.Л. Аршинский. – Иркутск : Изд-во

ИРНИТУ, 2017. – 24 c.