Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт Информационных Технологий и Анализа Данных

Кафедра вычислительной техники

**Название работы** – Разработка классов и их использование

Отчет по лабораторной работе № 2

Вариант 2

по дисциплине Объектно-ориентированное программирование

Выполнил

Студент, номер группы ИСМб-19-1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. Д. Солопов

Дата: \_\_\_\_\_\_\_\_\_\_

Принял

Должность \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.В. Маланова

Дата: \_\_\_\_\_\_\_\_\_\_

Иркутск 2020

Содержание

[1 Постановка задачи 3](#_Toc51516400)

[2 Описание структуры классов 4](#_Toc51516401)

[3 Описание методов класса и спецификация локальных переменных 6](#_Toc51516402)

[4 Таблица тестов 8](#_Toc51516403)

[5 Листинг исходного кода 12](#_Toc51516404)

[6 Заключение 16](#_Toc51516405)

[7 Список использованных источников 17](#_Toc51516406)

1 Постановка задачи

Провести объектную декомпозицию задачи из индивидуального задания и разработать класс, содержащий соответствующие поля для хранения необхо-димых данных и методы, обеспечивающие достаточную для решения задачи функциональность класса. Создание объекта класса решающего задание и вы-зов его методов должны осуществляться из главного класса. Ввод и вывод дан-ных осуществляется как в предыдущей лабораторной работе, для передачи ис-ходных данных в экземпляр класса решающего задачу должны быть разработа-ны соответствующие методы.

Индивидуальное задание выполняется на основе заданий к лабораторной работе 1.

**a.** Определите, является ли одна строка подстрокой другой.

**b.** Найти длину указанного слова в предложении.

**c.** Выведите из строки, содержащей слова, разделенные пробелами и запятыми, центральное слово (если в предложении два централь-ных слова, выведите оба).

2 Описание структуры классов

Таблица 1 – Описание структуры классов

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Имя метода** | **Назначение** | **Класс** |
| 1 | main(String[]) | Точка входа в программу | Main |
| 2 | substringInString(string) | Функция, решающая задачу а | StringClass |
| 3 | lengthDefineWord(int) | Функция, решающая задачу b | StringClass |
| 4 | centralWords() | Функция, решающая задачу c | StringClass |
| 5 | setText(string) | Функция, добавляющая текст в поле класса text | StringClass |
| 6 | getText() | Функция, возвращающая данные поля text | StringClass |
| 7 | StringClass() | Конструктор по умолчанию. | StringClass |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 8 | StringClass(string) | Конструктор с параметром | StringClass |
| 9 | StringClass(StringClass) | Конструктор копирования | StringClass |

**Словесное описание класса Main:**

Класс Main содержит в себе один статический метод main(), который является точкой входа в программу.

**Словесное описание класса StringClass:**

Класс не статический, содержит одно поле типа String – text, которое является исходной строкой, с которой методы класса осуществляют операции с целью решить поставленную задачу согласно индивидуальному варианту, пять методов: substringInString(string) – решает задачу a, lengthDefineWord(number) – решает задачу b, centralWords() – решает задачу c, setText(string) – метод set для text(записывает данные в поле text), getText() – метод get для text(возвращает данные text) и 3 конструктора: StringClass() – конструктор по умолчанию, StringClass(string) – конструктор с параметром, StringClass(StringClass) - конструктор копирования.

3 Описание методов класса и спецификация локальных переменных

Таблица 2 – Описание методов класса и спецификация локальных переменных

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Имя** | **Спецификатор** | **Входные данные** | **Назначение** | **Тип** | **Выходные данные** | **Назначение** | **Тип** |
| **Класс Main** | | | | | | | | |
| 1 | main | public | args | Аргументы командной строки | String[] | answers | Результат выполнения работы программы | String[] |
| **Класс StringClass** | | | | | | | | |
| 1 | StringClass | public | - | - | - | - | - | - |
| 2 | StringClass | public | \_text | Строка, для копирование в поле класса text | String | text | Строка, для работы объекта класса | String |
| 3 | StringClass | public | obj | Объект, для копирования | StringClass | text | Строка, для работы объекта класса | String |
| 4 | setText | public | \_text | Строка, для копирование в поле класса text | String | text | Строка, для работы объекта класса | String |

Продолжение таблицы 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | getText | public | - | - | - | text | Строка, для работы объекта класса | String |
| 6 | substringInString | public | substring | Строка для поиска | String | flag | Результат поиска | boolean |
| 7 | lengthDefineWord | public | number | Порядковый номер слова | int | lengthWord | Длина искомого слова | int |
| 8 | centerWords | public | - | - | - | results | Результат поиска | String |

Методы класса StringClass() используют встроенный метод trim(), доступ к которому имеет любой объект класса String. Данный метод используется для удаления символов пробела в начале и в конце строки.

4 Таблица тестов

Таблица 3 – Таблица тестов для решения задачи a

|  |  |  |  |
| --- | --- | --- | --- |
| **N теста** | **Назначение теста** | **Входные данные для теста** | **Выходные данные для теста** |
| 1 | Ввод осмысленного текста(подстрока и длина искомого слова найдена) | text = “Здравствуйте, уважаемые коллеги! Представляю вашему вниманию новый электромобиль!”  substring = “вашему”  number = 4 | Подстрока вашему присутствует в строке Здравствуйте, уважаемые коллеги! Представляю вашему вниманию новый электромобиль!  Слово с порядковым номером 4 имеет длину 11  Центральное слово/слова в предложении: Представляю вашему  (Рис. 1) |
| 2 | Ввод осмысленного текста(подстрока не найдена, но длина искомого слова найдена) | text = “Hello my world! Hello my friend! Hello my car!”  substring = “junior”  number = 5 | Подстрока junior не присутствует в строке Hello my world! Hello my friend! Hello my car!  Слово с порядковым номером 5 имеет длину 2  Центральное слово/слова в предложении: my (Рис. 2) |

Продолжение таблицы 3

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | Ввод осмысленного текста(подстрока не найдена и длина искомого слова равна 0) | text = “Какую книгу вы читаете?”  substring = “центр”  number = 5 | Подстрока центр не присутствует в строке Какую книгу вы читаете?  Слово с порядковым номером 5 имеет длину 0  Центральное слово/слова в предложении: книгу вы (Рис. 3) |
| 4 | Ввод не осмысленного текста | text = “AOFBAWUB FAWB An oanf > >,> ,?> ,? >,/> <?Famkan @ #!@$ !@$ !@$ !@$M>! @<$>! @<$!”  substring = “ @<$>!”  number = 15 | Подстрока @<$>! присутствует в строке AOFBAWUB FAWB An oanf > >,> ,?> ,? >,/> <?Famkan @ #!@$ !@$ !@$ !@$M>! @<$>! @<$!  Слово с порядковым номером 15 имеет длину 3  Центральное слово/слова в предложении: > (Рис. 4) |

Продолжение таблицы 3

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | Ввод не осмысленного текста(подстрока не найдена, размер искомого слова – 0) | text = “$%12ma ki91 9 9j ij bhg hbgv u8 n9”  substring = ”hello”  number = “12” | Подстрока hello не присутствует в строке $%12ma ki91 9 9j ij bhg hbgv u8 n9  Слово с порядковым номером 15 имеет длину 0  Центральное слово/слова в предложении: ij (Рис. 5) |

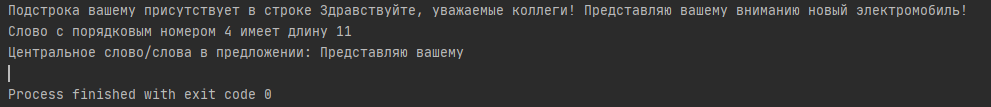


Рисунок 1 – Результат теста №1 из таблицы тестов №3

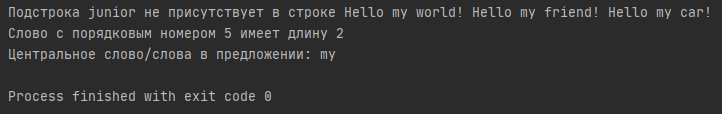


Рисунок 2 – Результат теста №2 из таблицы тестов №3

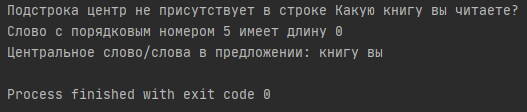


Рисунок 3 – Результат теста №3 из таблицы тестов №3

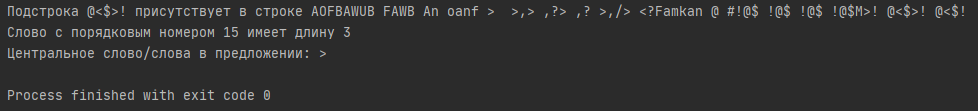


Рисунок 4 – Результат теста №4 из таблицы тестов №3

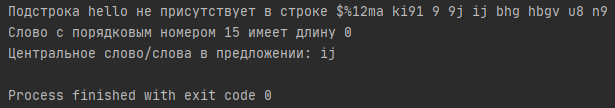


Рисунок 5 – Результат теста №5 из таблицы тестов №3

5 Листинг исходного кода

**Класс StringClass:**

package com.company;

public class StringClass {

private String text;

public StringClass(){ //Конструктор по умолчанию

text = "";

}

public StringClass(String \_text){ //Конструктор с параметром

\_text = \_text.trim();

text = \_text;

}

public StringClass(StringClass obj){ //Конструктор копирования

text = obj.text;

}

public void setText(String \_text){ //Метод для задания данных

\_text = \_text.trim();

text = \_text;

}

public String getText(){ //Метод для взятия данных

return text;

}

public boolean substringInString(String substring) {

/\*

a. Определите, является ли одна строка подстрокой другой.

\*/

substring = substring.trim();

boolean flag = false;

for(int i = 0; (i < text.length()) && (!flag); i++)

if(text.charAt(i) == substring.charAt(0)){

int count = 0;

for(int k = 0; k < substring.length(); k++)

if(substring.charAt(k) == text.charAt(k + i))

count++;

if(count == substring.length())

flag = true;

}

return flag;

}

public int lengthDefineWord(int number) {

/\*

b. Найти длину указанного слова в предложении.

\*/

int counter = 1;

int lengthWord = 0;

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

if((i >= 1) && ((text.charAt(i-1) == ' ') || (text.charAt(i-1) == ','))

&& ((text.charAt(i) != ' ') && (text.charAt(i) != ',')))

counter++;

if((counter == number) && ((text.charAt(i) != ' ') && (text.charAt(i) != ',')))

lengthWord++;

}

return lengthWord;

}

public String centerWords () {

//c. Выведите из строки, содержащей слова, разделенные пробелами и запятыми, центральное слово

// (если в предложении два централь-ных слова, выведите оба).

String results = "";

int countWords = 1;

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

if(((text.charAt(i - 1) == ' ') || (text.charAt(i-1) == ','))

&& ((text.charAt(i) != ' ') && (text.charAt(i) != ',')))

countWords++;

int number = (int)(countWords / 2), counter = 1;

if((countWords % 2) != 0)

number++;

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

if((i >= 1) && ((text.charAt(i-1) == ' ') || (text.charAt(i-1) == ','))

&& ((text.charAt(i) != ' ') && (text.charAt(i) != ',')))

counter++;

if((counter == number) && ((text.charAt(i) != ' ') && (text.charAt(i) != ',')))

results += text.charAt(i);

}

if((countWords % 2) == 0){

results += ' ';

counter = 1;

number++;

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

if((i >= 1) && ((text.charAt(i-1) == ' ') || (text.charAt(i-1) == ','))

&& ((text.charAt(i) != ' ') && (text.charAt(i) != ',')))

counter++;

if((counter == number) && ((text.charAt(i) != ' ') && (text.charAt(i) != ',')))

results += text.charAt(i);

}

}

return results;

}

}

**Класс Main:**

package com.company;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

System.out.print("Введите исходный текст: ");

String text = scan.nextLine();

System.out.print("Введите подстроку, которую нужно найти в исходном тексте: ");

String substring = scan.nextLine();

System.out.print("Введите номер слова, длину которого нужно найти в исходном тексте\n(номер считать слева направо от 1): ");

int number = scan.nextInt();

while((number <= 0) || (number >= text.length())){

System.out.print("Введите корректный номер слова: ");

number = scan.nextInt();

}

scan.close();

StringClass cls = new StringClass(text);

String[] answers = new String[3];

if(cls.substringInString(substring))

answers[0] = "Подстрока " + substring + " присутствует в строке " + text;

else

answers[0] = "Подстрока " + substring + " не присутствует в строке " + text;

answers[1] = "Слово с порядковым номером " + String.valueOf(number) + " имеет длину "

+ String.valueOf(cls.lengthDefineWord(number));

answers[2] = "Центральное слово/слова в предложении: " + cls.centerWords();

System.out.println("\nВывод результата: ");

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

System.out.println(answers[i]);

}

}

**Класс Main с демонстрацией работы конструкторов класса(конструктора по умолчанию, конструктора с параметром и конструктора копирования):**

package com.company;

import java.util.Scanner;

public class Main{

public static void main(String[] args){

StringClass obj1 = new StringClass(); // Конструктор по умолчанию

StringClass obj2 = new StringClass("Hello world!");// Конструктор с параметром

obj1.setText("Car"); // Присвоение полю text объекта obj1 значения "Car"

StringClass obj3 = new StringClass(obj1); // Конструктор копирования

obj1.setText("House"); // Присвоение полю text объекта obj1 значения "House"

// Функция getText() класса StringClass возвращает текущее значение поля text

System.out.println(obj1.getText()); // Вывод на консоль: House

System.out.println(obj2.getText()); // Вывод на консоль: Hello world!

System.out.println(obj3.getText()); // Вывод на консоль: Car

}

}

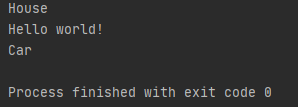


Рисунок 6 – Результат выполнения кода для демонстрации работы конструкторов

6 Заключение

Знакомство с интегрированной средой разработки приложений IDE NetBeans прошло успешно. На практике были применены элементы объектного подхода: применена объектная декомпозиция, инкапсуляция механизма реализации объекта. Был получен практический опыт написания и реализации классов, создания объектов, вызова их методов, работа с модификаторами доступа. Все тесты были успешно пройдены, ошибок или не корректной обработки данных не выявлено.

7 Список использованных источников

1. Java. Экспресс-курс [электронный ресурс] // Сайт Александра Климова [сайт], URL: <http://developer.alexanderklimov.ru/android/java/java.php> (дата обращения 12.09.20)

2. API Specification for the Java 7 SE. [официальный сайт] URL: <http://docs.oracle.com/javase/7/docs/api/> (дата обращения 12.09.20)

3. Гради Буч, Роберт А. Максимчук, Майкл У. Энгл, Бобби Дж. Янг, Джим Коналлен, Келли А. Хьюстон. Объектно-ориентированны анализ и проек-тирование с примерами приложений. Третье издание. М.: "Вильямс", 2010, -720 с.

4. Хабибуллин И.Ш. Java 7: для программистов / И. Ш. Хабибуллин. – Санкт-Петербург : БХВ–Петербург, 2014. – 768 с.

5. Васильев А. Н. Java. Объектно-ориентированное программирование: для магистров и бакалавров. Базовый курс по объектно-ориентированному программированию / А. Н. Васильев . – СПб.: Питер, 2012. – 395 с.