|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа,**

**обработки и интерпретации больших данных.**

**Отчет**

|  |
| --- |
| **по лабораторной работе № 7** |
|  |
| **Вариант 16** |

**Название:**

Строки и регулярные выражения

**Дисциплина:** Языки программирования для работы с большими данными

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-23М | |  |  | В.И. Тарасов |
|  | | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  | |  |  |  |  |
| Преподаватель | |  |  |  | П.В. Степанов |
|  | |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Цель работы** –получение навыков работы со строками и регулярными выражениями в языке Java.

**Ход работы**

**Задание 1:**

1. После каждого слова текста, заканчивающегося заданной подстрокой, вставить указанное слово.
2. В зависимости от признака (0 или 1) в каждой строке текста удалить указанный символ везде, где он встречается, или вставить его после k-гo символа.

Листинг 1 – Код задания 1

package src.lab7\_1;

import java.util.Scanner;

public class Program {

public static void main(String[] args) {

// 6. После каждого слова текста, заканчивающегося заданной подстрокой, вставить указанное слово.

String lookup\_word = "re";

String inserted\_word = "WORD";

String text = "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.";

System.out.println(text.replaceAll("(" + lookup\_word + ")\\b[,. ]", "$1 " + inserted\_word + " "));

// 7. В зависимости от признака (0 или 1) в каждой строке текста удалить указанный символ везде, где он встречается, или вставить его после k-гo символа.

String str = "Lorem ipsum dolor sit amet\nconsectetur adipiscing elit\nsed do eiusmod tempor incididunt ut labore\net dolore magna aliqua";

String[] split = str.split("\n");

Scanner console = new Scanner(System.in);

int flag = console.nextInt();

System.out.println("Введите символ");

String ch = console.next();

if (flag == 1){

System.out.println("Введите номер символа");

int k = console.nextInt();

for (String w: split){

System.out.println(w.substring(0, k) + ch + w.substring(k));

}

}

else {

for (String w: split){

System.out.println(w.replaceAll(ch, ""));

}

}

}

}

Результаты выполнения задачи представлены на рисунке 1.

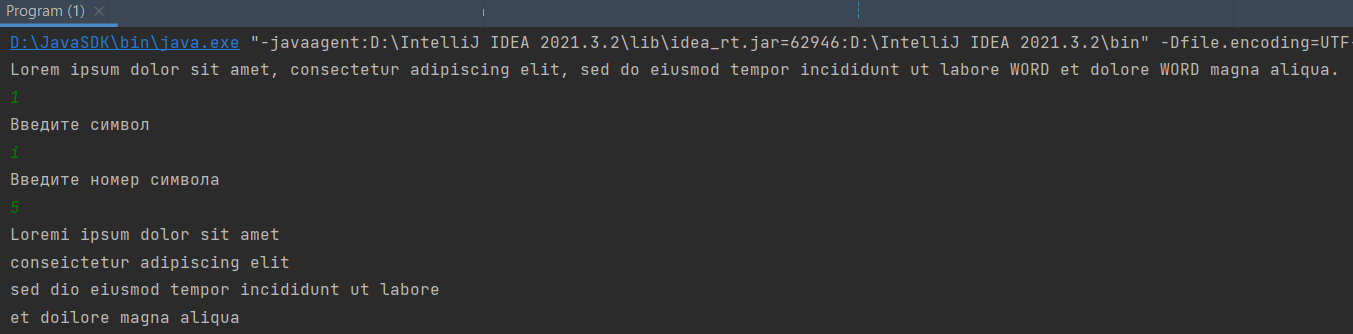


Рисунок 1 – Результат выполнения задания 1

**Задание 2:**

1. Напечатать без повторения слова текста, у которых первая и последняя буквы совпадают.
2. В тексте найти и напечатать все слова максимальной и все слова минимальной длины.

Листинг 2 – Код задания 2

package src.lab7\_2;

import java.util.ArrayList;

import java.util.HashSet;

public class Program {

public static void main(String[] args) {

// 6. Напечатать без повторения слова текста, у которых первая и последняя буквы совпадают.

String text = "aaa aba aba bbb baa daa";

String[] split = text.split(" ");

HashSet<String> set = new HashSet<>();

for (String w: split){

if (w.matches("(.).\*\\1")) {

set.add(w);

}

}

System.out.println(set);

// 7. В тексте найти и напечатать все слова максимальной и все слова минимальной длины.

String str = "aaaaaa b c d dd eee ffffff";

String[] splitStr = str.split(" ");

ArrayList<String> maxLen = new ArrayList<>();

ArrayList<String> minLen = new ArrayList<>();

int minLength = 9999;

int maxLength = 0;

for (String w: splitStr){

if (w.length() < minLength) minLength = w.length();

if (w.length() > maxLength) maxLength = w.length();

}

for (String w: splitStr){

if (w.length() == maxLength) maxLen.add(w);

if (w.length() == minLength) minLen.add(w);

}

System.out.println("Минимальная длина:");

System.out.println(minLen);

System.out.println("Максимальная длина:");

System.out.println(maxLen);

}

}

Результаты выполнения представлены на рисунке 2.

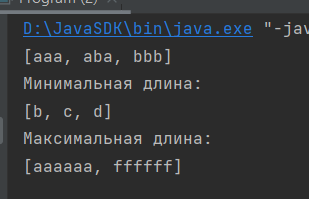


Рисунок 2 – Результат выполнения задания 2

**Задание 3:**

1. В предложении из n слов первое слово поставить на место второго, второе – на место третьего, и т.д., (n-1)-е слово – на место n-го, n-е слово поставить на место первого. В исходном и преобразованном предложениях между словами должны быть или один пробел, или знак препинания и один пробел.
2. Текст шифруется по следующему правилу: из исходного текста выбирается 1, 4, 7, 10-й и т.д. (до конца текста) символы, затем 2, 5, 8, 11-й и т.д. (до конца текста) символы, затем 3, 6, 9, 12-й и т.д. Зашифровать заданный текст.

Листинг 3 – Код задания 3

package src.lab7\_3;

public class Program {

public static void main(String[] args) {

/\* 6.В предложении из n слов первое слово поставить на место второго,

второе – на место третьего, и т.д., (n-1)-е слово – на место n-го, n-е слово поставить на место первого.

В исходном и преобразованном предложениях между словами должны быть или один пробел,

или знак препинания и один пробел.

\*/

String text = """

В предложении из n слов первое слово поставить на место второго,

второе – на место третьего, и т.д., (n-1)-е слово – на место n-го, n-е слово поставить на место первого.

В исходном и преобразованном предложениях между словами должны быть или один пробел,

или знак препинания и один пробел.""";

String res = "";

// Индекс последнего слова в тексте

int pos = text.lastIndexOf(" ");

int length = text.length();

// Буква, которая раньше была первой в тексте

String oldFirstLetter = String.valueOf(text.charAt(0)).toLowerCase();

// Буква, которая теперь будет первой в тексте

String newFirstLetter = String.valueOf(text.charAt(pos + 1)).toUpperCase();

if (String.valueOf(text.charAt(pos - 1)).matches("\\p{Punct}")) {

// Если есть знак препинания перед последним словом

res = newFirstLetter // Первая буква последнего слова

+ text.substring(pos + 2, length - 1) // Последнее слово до точки

+ text.charAt(pos - 1) // Знак препинания, который был перед последним словом

+ " "

+ oldFirstLetter // Первая буква текста

+ text.substring(1, pos - 1) // Весь оставшийся текст

+ text.substring(length - 1); // Последняя точка

}

else {

// Если перед последним словом просто пробел

res = newFirstLetter // Первая буква последнего слова

+ text.substring(pos + 2, length - 1) // Последнее слово до точки

+ " "

+ oldFirstLetter // Первая буква текста

+ text.substring(1, pos) // Весь оставшийся текст

+ text.substring(length - 1); // Последняя точка

}

System.out.println(res);

/\* 7. Текст шифруется по следующему правилу: из исходного текста выбирается 1, 4, 7, 10-й и т.д. (до конца текста) символы,

затем 2, 5, 8, 11-й и т.д. (до конца текста) символы, затем 3, 6, 9, 12-й и т.д. Зашифровать заданный текст.

\*/

StringBuilder result = new StringBuilder("");

String txt = "Lorem ipsum dolor sit amet";

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

for (int j=i; j < txt.length(); j+=3) {

result.append(txt.charAt(j));

}

}

System.out.println(result);

}

}

Результаты выполнения задания 3 представлены на рисунке 3.

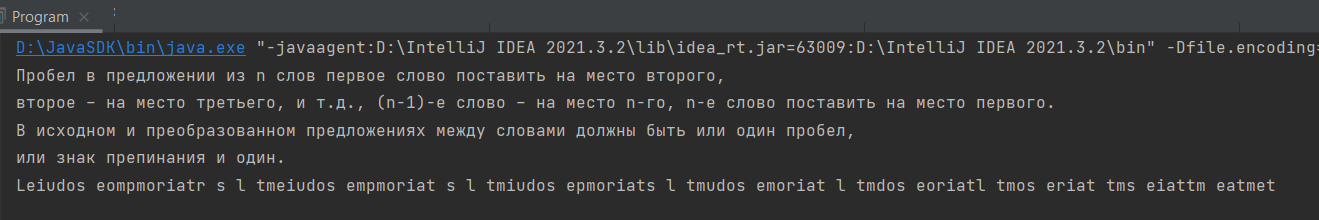


Рисунок 3 – Результаты выполнения задания 3

**Задание 4:**

1. Вывести в заданном тексте все слова, расположив их в алфавитном порядке.
2. Подсчитать, сколько слов в заданном тексте начинается с прописной буквы.

Листинг 4 – Код задания 4

package src.lab7\_4;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Collections;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Program {

public static void main(String[] args) {

// 6. Вывести в заданном тексте все слова, расположив их в алфавитном порядке.

String str = "aaaa bbb hhh tttt ee c d dd eee ffffff";

ArrayList<String> arr = new ArrayList<>(Arrays.asList(str.split(" ")));

Collections.sort(arr);

System.out.println(arr);

// 7. Подсчитать, сколько слов в заданном тексте начинается с прописной буквы.

String text = "Aaaaaa bbbb Ccccc Dddd Eeee fFFf ggg aAAaa";

Pattern pattern = Pattern.compile("\\b[A-ZА-Я]");

Matcher matcher = pattern.matcher(text);

System.out.println(matcher.results().count());

}

}

Результаты выполнения задания 4 представлены на рисунке 4.

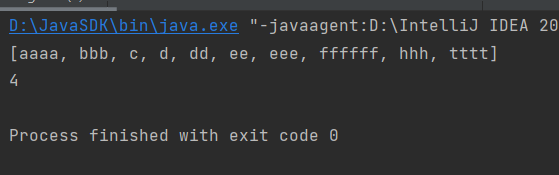


Рисунок 4 – Результат выполнения задания 4

**Местоположение проектных файлов** –файлы проекта расположены в репозитории GitHub по следующим адресам:

<https://github.com/Fregatty/BigDataLanguages/tree/main/src/lab7_1>

<https://github.com/Fregatty/BigDataLanguages/tree/main/src/lab7_2>

<https://github.com/Fregatty/BigDataLanguages/tree/main/src/lab7_3>

<https://github.com/Fregatty/BigDataLanguages/tree/main/src/lab7_4>

**Вывод** – в результате выполнения лабораторной работы были получены навыки работы со строками в языке Java. Были изучены принципы действия регулярных выражений и применена фильтрация на их основе.