|  |  |
| --- | --- |
| 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 **09.04.01/12 Интеллектуальный анализ больших**

**данных в системах поддержки принятия решений.**

**Отчет**

**По лабораторной работе №7**

**Название:** Строки, регулярные выражения

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

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

Москва, 2023

**Задание 1:** в тексте после k-го символа вставить заданную подстроку.

Решение задания представлено в листинге 1.

Листинг 1 – Программный код модуля Main.java для задания 1

public class Main {

public static void main(String[] args) {

String text = "Клара у Карла украла кораллы.";

int k = 10; // Позиция, после которой будет вставлена подстрока

String substring = "ВСТАВИЛ";

String result = insertSubstring(text, k, substring);

System.out.println(result);

}

public static String insertSubstring(String text, int k, String substring) {

StringBuilder builder = new StringBuilder(text);

if (k <= text.length()) {

builder.insert(k, substring);

} else {

builder.append(substring);

}

return builder.toString();

}

}

**Задание 2:** после каждого слова текста, заканчивающегося заданной подстрокой, вставить указанное слово.

Решение задания представлено в листинге 2.

Листинг 2 – Программный код модуля Main.java для задания 2

public class Main {

public static void main(String[] args) {

String text = "Я хочу кушать. Я хочу пить.";

String targetSubstring = "хочу";

String insertedWord = "ВСТАВИЛ";

String result = insertWordAfterSubstring(text, targetSubstring, insertedWord);

System.out.println(result);

}

public static String insertWordAfterSubstring(String text, String targetSubstring, String insertedWord) {

StringBuilder builder = new StringBuilder();

String[] words = text.split("\\s+");

for (String word : words) {

if (word.endsWith(targetSubstring)) {

builder.append(word).append(" ").append(insertedWord).append(" ");

} else {

builder.append(word).append(" ");

}

}

return builder.toString().trim();

}

}

**Задание 3:** в стихотворении найти количество слов, начинающихся и заканчивающихся гласной буквой.

Решение задания представлено в листинге 3.

Листинг 3 – Программный код модуля Main.java для задания 3

public class Main {

public static void main(String[] args) {

String poem = "Сжала руки под тёмной вуалью\n" +

"Отчего ты сегодня бледна?\n" +

"Оттого, что я терпкой печалью\n" +

"Напоила его допьяна.";

int count = countWordsWithVowel(poem);

System.out.println("Количество слов, начинающихся и заканчивающихся гласной буквой: " + count);

}

public static int countWordsWithVowel(String poem) {

String[] words = poem.split("\\s+");

int count = 0;

for (String word : words) {

if (startsWithVowel(word) && endsWithVowel(word)) {

count++;

}

}

return count;

}

public static boolean startsWithVowel(String word) {

String firstChar = word.substring(0, 1).toLowerCase();

return firstChar.matches("[аеёиоуыэюя]");

}

public static boolean endsWithVowel(String word) {

String lastChar = word.substring(word.length() - 1).toLowerCase();

return lastChar.matches("[аеёиоуыэюя]");

}

}

**Задание 4:** напечатать без повторения слова текста, у которых первая и последняя буквы совпадают.

Решение задания представлено в листинге 4.

Листинг 4 – Программный код модуля Main.java для задания 4

import java.util.HashSet;

public class Main {

public static void main(String[] args) {

String text = "Олово поп рука гриб арфа олово поп";

printWordsWithMatchingFirstAndLastLetter(text);

}

public static void printWordsWithMatchingFirstAndLastLetter(String text) {

String[] words = text.split("\\s+");

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

for (String word : words) {

if (word.length() > 1 && word.toLowerCase().charAt(0) == word.toLowerCase().charAt(word.length() - 1)) {

uniqueWords.add(word.toLowerCase());

}

}

for (String uniqueWord : uniqueWords) {

System.out.println(uniqueWord);

}

}

}

**Задание 5:** в каждом предложении текста поменять местами первое слово с последним, не изменяя длины предложения.

Решение задания представлено в листинге 5.

Листинг 5 – Программный код модуля Main.java для задания 5

public class Main {

public static void main(String[] args) {

String text = "У Лукоморья дуб зеленый, златая цепь на дубе том. И днем и ночью кот учены, все ходит по цепи кругом.";

String result = swapFirstAndLastWords(text);

System.out.println(result);

}

public static String swapFirstAndLastWords(String text) {

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

StringBuilder builder = new StringBuilder();

for (String sentence : sentences) {

String[] words = sentence.trim().split("\\s+");

if (words.length > 1) {

String firstWord = words[0];

String lastWord = words[words.length - 1];

words[0] = lastWord;

words[words.length - 1] = firstWord;

}

String swappedSentence = String.join(" ", words);

builder.append(swappedSentence).append(". ");

}

return builder.toString().trim();

}

}

**Задание 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.

Решение задания представлено в листинге 6.

Листинг 6 – Программный код модуля Main.java для задания 6

public class Main {

public static void main(String[] args) {

String sentence = "Вася ест булочку у подъезда и пьет сок.";

String transformedSentence = transformSentence(sentence);

System.out.println("Исходное предложение: " + sentence);

System.out.println("Преобразованное предложение: " + transformedSentence);

}

public static String transformSentence(String sentence) {

String[] words = sentence.split("\\s+");

int n = words.length;

// Сохраняем последнее слово для использования в начале преобразованного предложения

String lastWord = words[n - 1];

// Сдвигаем каждое слово влево, начиная с последнего

for (int i = n - 1; i > 0; i--) {

words[i] = words[i - 1];

}

// Помещаем последнее слово на место первого

words[0] = lastWord;

return String.join(" ", words);

}

}

**Задание 7:** заменить все одинаковые рядом стоящие символы в тексте одним символом.

Решение задания представлено в листинге 7.

Листинг 7 – Программный код модуля Main.java для задания 7

public class Main {

public static void main(String[] args) {

String text = "Онн хоооддииит на боккккс";

String transformedText = replaceRepeatedCharacters(text);

System.out.println("Исходный текст: " + text);

System.out.println("Преобразованный текст: " + transformedText);

}

public static String replaceRepeatedCharacters(String text) {

return text.replaceAll("(.)\\1+", "$1");

}

}

**Задание 8:** заменить все одинаковые рядом стоящие символы в тексте одним символом.

Решение задания представлено в листинге 8.

Листинг 8 – Программный код модуля Main.java для задания 8

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Collections;

public class Main {

public static void main(String[] args) {

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

ArrayList<String> words = extractWords(text);

Collections.sort(words);

System.out.println("Слова в алфавитном порядке:");

for (String word : words) {

System.out.println(word);

}

}

public static ArrayList<String> extractWords(String text) {

String[] wordsArray = text.split("\\W+");

ArrayList<String> words = new ArrayList<>(Arrays.asList(wordsArray));

return words;

}

}

**Вывод:** в ходе выполнения данной лабораторной работы были изучены приемы работы со строковыми данными на языке Java, а также применены регулярные выражения для фильтрования строковых данных.