|  |  |
| --- | --- |
| 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**

**Отчет**

**по лабораторной работе № 7**

**Название:** Регулярные выражения

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-21М |  |  | Д.Ю. Ермошин |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | П.В. Степанов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2020

**Задание**

Научиться работать со строками и регулярными выражениями в языке Java.

**Выполнение работы**

**Вариант 1**

В тексте после k-го символа вставить заданную подстроку.

|  |
| --- |
| package lab7;  /\*\*  \* 5. В тексте после k-го символа вставить заданную подстроку.  \*/  public class Variant1 {  private final String text;  public Variant1(final String text) {  this.text = text;  }  public String work(final int position, final String substring) {  if (position > text.length())  return text + substring;  return text.substring(0, position) + substring + text.substring(position);  }  } |

**Вариант 2**

Найти и напечатать, сколько раз повторяется в тексте каждое слово, которое встречается в нем.

|  |
| --- |
| package lab7;  import java.util.HashMap;  import java.util.Map;  /\*\*  \* 2. Найти и напечатать, сколько раз повторяется в тексте каждое слово, которое встречается в нем.  \*/  public class Variant2 {  private final String text;  private final Map<String, Integer> wordsCounter = new HashMap<>();  public Variant2(final String text) {  if (text.isEmpty())  throw new IllegalArgumentException("Input text must not be empty");  this.text = text;  }  public void work() {  countWords();  printResult();  }  private void countWords() {  final String words = text.replaceAll("[!?,.]", "");  for (final String word : words.split("\\s+")) {  if (!wordsCounter.containsKey(word))  wordsCounter.put(word, 1);  else  wordsCounter.put(word, wordsCounter.get(word) + 1);  }  }  private void printResult() {  wordsCounter.forEach((key, value) -> System.out.println(key + ": " + value));  }  } |

**Вариант 3**

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.

|  |
| --- |
| package lab7;  import java.util.Arrays;  import java.util.List;  /\*\*  \* 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.  \*/  public class Variant3 {  private final String sentence;  private final char sentenceEnd;  private final List<String> words;  public Variant3(String sentence) {  this.sentence = sentence;  this.sentenceEnd = sentence.charAt(sentence.length() - 1);  this.words = Arrays.asList(sentence.replaceAll("[:;,]", "").split("\\s+"));  }  public String work() {  final String lastWord = words.get(words.size() - 1);  // Get last word without .?!  final StringBuilder builder = new StringBuilder(lastWord.substring(0, lastWord.length() - 1));  int wordEndPosition = 0;  for (int i = 0; i < words.size() - 1; i++) {  final int wordBeginPosition = sentence.indexOf(words.get(i), wordEndPosition);  wordEndPosition = wordBeginPosition + words.get(i).length();  builder.append(sentence, wordEndPosition, sentence.indexOf(words.get(i + 1), wordEndPosition));  builder.append(words.get(i));  }  return builder.append(sentenceEnd).toString();  }  } |

**Вариант 4**

Заменить все одинаковые рядом стоящие символы в тексте одним символом.

|  |
| --- |
| package lab7;  /\*\*  \* 5. Заменить все одинаковые рядом стоящие символы в тексте одним символом.  \*/  public class Variant4 {  private final String text;  private final char[] characterArray;  public Variant4(String text) {  this.text = text;  characterArray = text.toCharArray();  }  public String work() {  final char[] resultArray = new char[text.length()];  int j = 0;  for (int i = 0; i < characterArray.length; i++) {  if (i + 1 < text.length() && characterArray[i] == characterArray[i + 1]) {  int pos = i + 1;  while (characterArray[pos] == characterArray[pos + 1])  pos++;  i = pos;  }  resultArray[j] = characterArray[i];  j++;  }  return new String(resultArray, 0, j);  }  } |

**Заключение**

В данной лабораторной работе мы научились работать со строками и регулярными выражениями в языке Java.