|  |  |
| --- | --- |
| 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 |

**Название:** Строки, регулярные выражения

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-22М |  |  | Р.В. Баканов | |
|  | (Группа) |  | (Подпись, дата) | | (И.О. Фамилия) |
|  |  |  |  | |  |
| Преподаватель |  |  |  | | П.В. Степанов |
|  |  |  | (Подпись, дата) | | (И.О. Фамилия) |

Москва, 2024

**Цель работы:** освоить базовые принципы работы со строками и регулярными выражениями в Java.

**Вариант: 2.**

**Задание 1:** В русском тексте каждую букву заменить ее порядковым номером в алфавите. При выводе в одной строке печатать текст с двумя пробелами между буквами, в следующей строке внизу под каждой буквой печатать ее номер.

Код решения приведен в листинге 1.

Листинг 1 — Задание 1

import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class Main {  
 public static void main(String[] args) {  
 String input = "Не пора ль нам, братия начать";  
  
 StringBuilder result1 = new StringBuilder();  
 StringBuilder result2 = new StringBuilder();  
  
 Pattern pattern = Pattern.*compile*("(?U)\\w");  
 Matcher matcher = pattern.matcher(input);  
  
 while (matcher.find()) {  
 result1.append(String.*format*("%c ", input.charAt(matcher.start())));  
 result2.append(String.*format*("%d ", (int) input.charAt(matcher.start())));  
 }  
  
 System.*out*.println(input);  
 System.*out*.println("Result 1: " + result1.toString());  
 System.*out*.println("Result 2: " + result2.toString());  
 }  
}

**Задание 2:** В тексте после буквы Р, если она не последняя в слове, ошибочно напечатана буква А вместо О. Внести исправления в текст.

Код решения приведен в листинге 2.

Листинг 2 — Задание 2

import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class Main {  
 public static void main(String[] args) {  
 String input = "Три закона работехники в научной фантастике — обязательные правила поведения для работов, впервые сформулираванные Айзеком Азимовым в рассказе «Хоравод»";  
  
 String result = input.replaceAll("ра", "ро");  
  
 System.out.println("Input: " +input);  
 System.out.println("Result: " + result);  
 }  
}

**Задание 3:** Найти и напечатать, сколько раз повторяется в тексте каждое слово, которое встречается в нем.

Код решения приведен в листинге 3.

Листинг 3 – Задание 3

import java.util.HashMap;  
  
public class Main {  
 public static void main(String[] args) {  
 var input = "Олень-северное животное.В летнее время оленям в тайге жарко,а в горах даже в июле холодно.Олень как бы создан для северных просторов,жёсткого ветра,длинных морозных ночей.Олень легко бежит вперёд по тайге,подминает под себя кусты,переплывает быстрые реки.Олень не тонет,потому что каждая его шерстинка-это длинная трубочка,которую внутри наполняет воздух..Нос у оленя покрыт серебристой шёрсткой.Если бы шерсти на носу не было,олень бы его отморозил.";  
  
 var words = input.split("(?U)\\W+");  
  
 var wordCount = new HashMap<String, Integer>();  
  
 for (var word : words) {  
 var lowerCaseWord = word.toLowerCase();  
  
 var count = wordCount.containsKey(lowerCaseWord)  
 ? wordCount.get(lowerCaseWord) + 1  
 : 1;  
  
 wordCount.put(lowerCaseWord, count);  
 }  
  
 System.*out*.println(input);  
  
 for (var word : wordCount.keySet()) {  
 System.*out*.printf("%s: %d\n", word, wordCount.get(word));  
 }  
 }  
}

**Задание 4:** В тексте найти и напечатать n символов (и их количество), встречающихся наиболее часто.

Код решения приведен в листинге 4.

Листинг 4 — Задание 4

import java.util.ArrayList;  
import java.util.HashMap;  
  
public class Main {  
 public static void main(String[] args) {  
 var input = "Олень-северное животное.В летнее время оленям в тайге жарко,а в горах даже в июле холодно.Олень как бы создан для северных просторов,жёсткого ветра,длинных морозных ночей.Олень легко бежит вперёд по тайге,подминает под себя кусты,переплывает быстрые реки.Олень не тонет,потому что каждая его шерстинка-это длинная трубочка,которую внутри наполняет воздух..Нос у оленя покрыт серебристой шёрсткой.Если бы шерсти на носу не было,олень бы его отморозил.";  
 var lowerCaseInput = input.replace(" ", "").toLowerCase();  
  
 var characterCount = new HashMap<Character, Integer>();  
 var maxCount = 0;  
  
 for (int i = 0; i < lowerCaseInput.length(); i++) {  
 var character = lowerCaseInput.charAt(i);  
  
 var count = characterCount.containsKey(character)  
 ? characterCount.get(character) + 1  
 : 1;  
  
 if (count > maxCount) {  
 maxCount = count;  
 }  
  
 characterCount.put(character, count);  
 }  
  
 var maxCountCharacters = new ArrayList<Character>();  
  
 for (var character : characterCount.keySet()) {  
 var count = characterCount.get(character);  
  
 if (count == maxCount) {  
 maxCountCharacters.add(character);  
 }  
 }  
  
 System.out.println(input);  
  
 for (var character : maxCountCharacters) {  
 System.out.printf("%c: %d\n", character, characterCount.get(character));  
 }  
 }  
}

**Задание 5:** Найти наибольшее количество предложений текста, в которых есть одинаковые слова.

Код решения приведен в листингах 5 и 6.

Листинг 5 – Задание 5 (1)

import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.HashSet;  
  
public class Main {  
 public static void main(String[] args) {  
 var input = "В такой точно день охотился я однажды за тетеревами в Чернском уезде, Тульской губернии. Я нашел и настрелял довольно много дичи; наполненный ягдташ немилосердно резал мне плечо; но уже вечерняя заря погасала, и в воздухе, еще светлом, хотя не озаренном более лучами закатившегося солнца, начинали густеть и разливаться холодные тени, когда я решился, наконец, вернуться к себе домой. Быстрыми шагами прошел я длинную «площадь» кустов, взобрался на холм и, вместо ожиданной знакомой равнины с дубовым леском направо и низенькой белой церковью в отдалении, увидал совершенно другие, мне не известные места. У ног моих тянулась узкая долина; прямо, напротив, крутой стеной возвышался частый осинник.";  
  
 var sentences = input.toLowerCase().split("(\\.\\.\\.)|\\.|!|\\?");  
  
 var sentenceWords = new ArrayList<HashSet<String>>();  
  
 for (var sentence : sentences) {  
 var words = new HashSet<>(Arrays.*asList*(sentence.trim().split("(?U)\\W++")));  
  
 sentenceWords.add(words);  
 }  
  
 var commonWordsFinder = new CommonWordsFinder(sentenceWords);  
 commonWordsFinder.findSentencesWithCommonWords();  
  
 System.*out*.println(commonWordsFinder.getMaxDepth());  
 }  
}

Листинг 6 – Задание 5 (2)

import java.util.ArrayList;  
import java.util.HashSet;  
  
public class CommonWordsFinder {  
 private final ArrayList<HashSet<String>> \_sentenceWords;  
 private int \_maxDepth;  
  
 public CommonWordsFinder(ArrayList<HashSet<String>> sentenceWords) {  
 \_sentenceWords = sentenceWords;  
 \_maxDepth = 1;  
 }  
  
 public void findSentencesWithCommonWords() {  
 for (int i = 0; i < \_sentenceWords.size(); i++) {  
 findSentencesWithCommonWords(i, new HashSet<>());  
 }  
 }  
  
 public int getMaxDepth() {  
 return \_maxDepth;  
 }  
  
 private void findSentencesWithCommonWords(int i, HashSet<Integer> checkedSentences) {  
 checkedSentences.add(i);  
  
 var achievedDepth = checkedSentences.size();  
  
 if (achievedDepth > \_maxDepth) {  
 \_maxDepth = achievedDepth;  
 }  
  
 for (int j = 0; j < \_sentenceWords.size(); j++) {  
 if (checkedSentences.contains(j)) {  
 continue;  
 }  
  
 var intersection = new HashSet<>(\_sentenceWords.get(i));  
 var currentSentenceWords = \_sentenceWords.get(j);  
  
 intersection.retainAll(currentSentenceWords);  
  
 if (intersection.size() > 0) {  
 findSentencesWithCommonWords(j, checkedSentences);  
 }  
 }  
  
 checkedSentences.remove(i);  
 }  
}

**Задание 6:** Найти такое слово в первом предложении, которого нет ни в одном из остальных предложений.

Код решения приведен в листинге 7.

Листинг 7 – Задание 6

import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.HashSet;  
  
public class Main {  
 public static void main(String[] args) {  
 var input = "Dummy test sentence. Second sentence to exclude word. It is funny word this 'dummy'.";  
  
 var sentences = input.toLowerCase().split("(\\.\\.\\.)|\\.|!|\\?");  
  
 var sentenceWords = new ArrayList<HashSet<String>>();  
  
 for (var sentence : sentences) {  
 var words = new HashSet<>(Arrays.asList(sentence.trim().split("(?U)\\W++")));  
  
 sentenceWords.add(words);  
 }  
  
 var firstSentenceWords = sentenceWords.getFirst();  
  
 for (int i = 1; i < sentenceWords.size(); i++) {  
 firstSentenceWords.removeAll(sentenceWords.get(i));  
 }  
  
 for (var word : firstSentenceWords) {  
 System.out.println(word);  
 }  
 }  
}

**Задание 7:** Ввести текст и список слов. Для каждого слова из заданного списка найти, сколько раз оно встречается в тексте, и рассортировать слова по убыванию количества вхождений.

Код решения приведен в листингах 8 и 9.

Листинг 8 – Задание 7 (1)

import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.Comparator;  
  
public class Main {  
 public static void main(String[] args) {  
 var input = "Test. Dummy mock. Testing test.";  
 var wordsToFind = new ArrayList<String>();  
  
 wordsToFind.add("test");  
 wordsToFind.add("dummy");  
 wordsToFind.add("mock");  
  
 var wordsWithCount = new ArrayList<WordWithCount>();  
  
 var words = input.toLowerCase().trim().split("(?U)\\W++");  
  
 for (var word : wordsToFind) {  
 var lowerCaseWord = word.toLowerCase();  
  
 var count = Arrays.stream(words).filter(w -> w.equals(lowerCaseWord)).count();  
  
 var wordWithCount = new WordWithCount(word, count);  
  
 wordsWithCount.add(wordWithCount);  
 }  
  
 wordsWithCount.sort(Comparator.comparingLong(w -> -w.Count));  
  
 for (var word : wordsWithCount) {  
 System.out.printf("%s: %d\n", word.Word, word.Count);  
 }  
 }  
}

Листинг 9 – Задание 7 (2)

public class WordWithCount {  
 public String Word;  
 public long Count;  
  
 public WordWithCount(String word, long count) {  
 Word = word;  
 Count = count;  
 }  
}

**Задание 8:** Все слова текста рассортировать в порядке убывания их длин, при этом все слова одинаковой длины рассортировать в порядке возрастания в них количества гласных букв.

Код решения приведен в листингах 10 и 11.

Листинг 10 – Задание 8 (1)

import java.util.ArrayList;  
import java.util.Comparator;  
import java.util.HashMap;  
import java.util.HashSet;  
  
public class Main {  
 public static void main(String[] args) {  
 var input = "Test. Cool dummy mock. Testing test. IKEA.";  
  
 var words = input.trim().split("(?U)\\W++");  
  
 var wordsByLength = new HashMap<Integer, ArrayList<WordWithInfo>>();  
  
 for (var word : words) {  
 var wordWithInfo = new WordWithInfo(word, word.length(), *countVowels*(word));  
  
 if (!wordsByLength.containsKey(wordWithInfo.Length)) {  
 wordsByLength.put(wordWithInfo.Length, new ArrayList<>());  
 }  
  
 wordsByLength.get(wordWithInfo.Length).add(wordWithInfo);  
 }  
  
 for (var length : wordsByLength.keySet().stream().sorted((l, r) -> r - l).toArray(Integer[]::new)) {  
 var wordsWithSameLength = wordsByLength.get(length);  
 wordsWithSameLength.sort(Comparator.*comparingInt*(l -> l.VowelCount));  
  
 for (var word : wordsWithSameLength) {  
 System.*out*.printf("%s ", word.Word);  
 }  
 }  
 }  
  
 private static int countVowels(String word) {  
 var lowerCaseWord = word.toLowerCase();  
 var count = 0;  
 var vowels = *getVowels*();  
  
 for (int i = 0; i < lowerCaseWord.length(); i++) {  
 if (vowels.contains(lowerCaseWord.charAt(i))) {  
 count++;  
 }  
 }  
  
 return count;  
 }  
  
 private static HashSet<Character> getVowels() {  
 var vowels = new HashSet<Character>();  
  
 vowels.add('a');  
 vowels.add('e');  
 vowels.add('i');  
 vowels.add('o');  
 vowels.add('u');  
  
 vowels.add('е');  
 vowels.add('ы');  
 vowels.add('а');  
 vowels.add('о');  
 vowels.add('э');  
 vowels.add('я');  
 vowels.add('и');  
 vowels.add('ю');  
  
 return vowels;  
 }  
}

Листинг 11 – Задание 8 (2)

public class WordWithInfo {  
 public String Word;  
 public int Length;  
 public int VowelCount;  
  
 public WordWithInfo(String word, int length, int vowelCount) {  
 Word = word;  
 Length = length;  
 VowelCount = vowelCount;  
 }  
}

**Вывод:** в результате выполнения лабораторной работы были разработаны несколько программ в соответствие с требованиями задания.