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

Вариант 6

**Название:** Строки. Регулярные выражения.

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

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

Москва, 2023

**ВВЕДЕНИЕ**

**Задание:**

* 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.
  3. Напечатать без повторения слова текста, у которых первая и последняя буквы совпадают.
  4. В тексте найти и напечатать все слова максимальной и все слова минимальной длины.
  5. 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.
  6. 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.
  7. Вывести в заданном тексте все слова, расположив их в алфавитном порядке.
  8. Подсчитать, сколько слов в заданном тексте начинается с прописной буквы.

**ОСНОВНАЯ ЧАСТЬ**

**Задание 1.1**

После каждого слова текста, заканчивающегося заданной подстрокой, вставляется указанное слово.

Код программы приведен в листинге 1. Результат выполнения программы представлен на рисунке 1.

Листинг 1 – Код класса Main

import java.util.Scanner;  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
  
 System.out.print("Введите текст: ");  
 String text = scanner.nextLine();  
  
 System.out.print("Введите подстроку, которая ищется: ");  
 String substring = scanner.nextLine();  
  
 System.out.print("Введите слово-вставку: ");  
 String insert = scanner.nextLine();  
  
 String[] words = text.split("\\s+");  
  
 StringBuilder result = new StringBuilder();  
  
 for (int i = 0; i < words.length; i++) {  
 result.append(words[i]);  
 if (words[i].endsWith(substring)) {  
 result.append(" " + insert);  
 }  
 result.append(" ");  
 }  
  
 System.out.println(result.toString());  
 }  
}

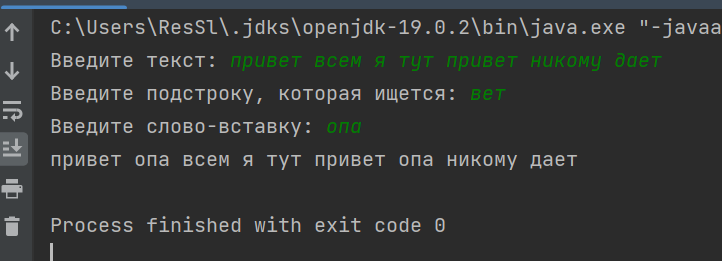


Рисунок 1 – Результат работы программы 1.1

**Задание 1.2**

В зависимости от признака (0 или 1) в каждой строке текста удаляется указанный символ везде, где он встречается, или вставляется после k-гo символа

Код программы приведен в листинге 2. Результат выполнения программы представлен на рисунке 2.

Листинг 2 – Код класса Main

import java.util.Scanner;  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Введите текст: ");  
 String text = scanner.nextLine();  
  
 System.*out*.print("Введите символ для удаления или вставки: ");  
 char symbol = scanner.next().charAt(0);  
  
 System.*out*.print("Введите признак действия (0 для удаления, 1 для вставки): ");  
 int action = scanner.nextInt();  
  
 // Если вставляем, то задаем позицию  
 int position = -1;  
 if (action == 1) {  
 System.*out*.print("Введите позицию для вставки символа: ");  
 position = scanner.nextInt();  
 }  
  
 StringBuilder result = new StringBuilder();  
  
 for (int i = 0; i < text.length(); i++) {  
 char c = text.charAt(i);  
  
 if (action == 0 && c == symbol) {  
 continue;  
 } else {  
 result.append(c);  
 }  
 }  
 if (action == 1) result.insert(position + 1, symbol);  
 System.*out*.println(result.toString());  
 }  
}

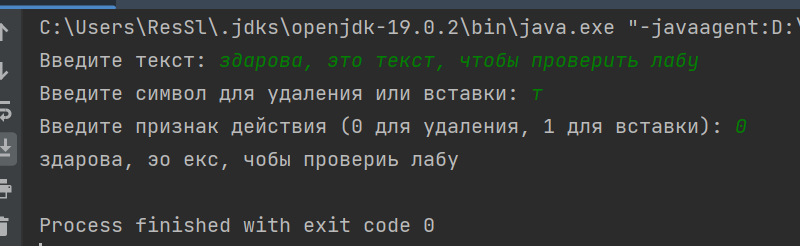


Рисунок 2 – Результат работы программы 1.2

**Задание 1.3**

Печатаются без повторения слова текста, у которых первая и последняя буквы совпадают.

Код программы приведен в листинге 3. Результат выполнения программы представлен на рисунке 3.

Листинг 3 – Код класса Main

import java.util.HashSet;  
import java.util.Scanner;  
import java.util.Set;  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Введите текст: ");  
 String text = scanner.nextLine();  
  
 String[] words = text.split("\\s+");  
  
 Set<String> wordSet = new HashSet<>();  
  
 for (String word : words) {  
 if (wordSet.contains(word)) {  
 continue;  
 }  
 if (word.charAt(0) == word.charAt(word.length() - 1)) {  
 wordSet.add(word);  
 }  
 }  
  
 for (String word : wordSet) {  
 System.*out*.println(word);  
 }  
 }  
}

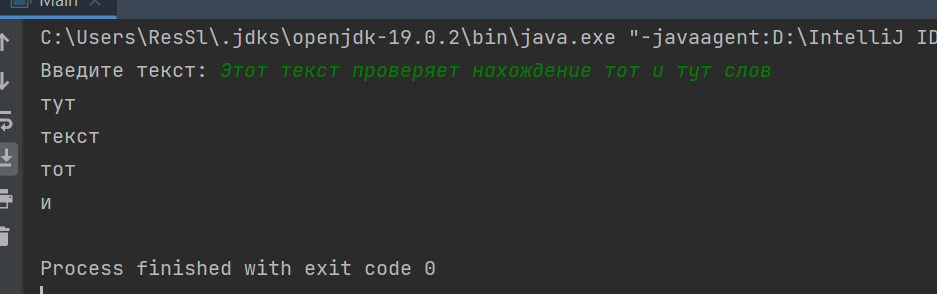


Рисунок 3 – Результат работы программы 1.3

**Задание 1.4**

В тексте находятся и печатаются все слова максимальной и все слова минимальной длины.

Код программы приведен в листинге 4. Результат выполнения программы представлен на рисунке 4.

Листинг 4 – Код класса Main

import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Введите текст: ");  
 String text = scanner.nextLine();  
  
 String[] words = text.split("\\s+");  
  
 int maxLength = 0;  
 int minLength = Integer.*MAX\_VALUE*;  
  
 List<String> longestWords = new ArrayList<>();  
 List<String> shortestWords = new ArrayList<>();  
  
 for (String word : words) {  
 int length = word.length();  
 if (length > maxLength) {  
 maxLength = length;  
 }  
 if (length < minLength) {  
 minLength = length;  
 }  
 }  
  
 for (String word : words) {  
 if (word.length() == maxLength) {  
 longestWords.add(word);  
 }  
  
 if (word.length() == minLength) {  
 shortestWords.add(word);  
 }  
 }  
  
 System.*out*.print("Слова максимальной длины: ");  
 for (String word : longestWords) {  
 System.*out*.print(word + "; ");  
 }  
 System.*out*.println();  
  
 System.*out*.print("Слова минимальной длины: ");  
 for (String word : shortestWords) {  
 System.*out*.print(word + "; ");  
 }  
 System.*out*.println();  
 }  
}

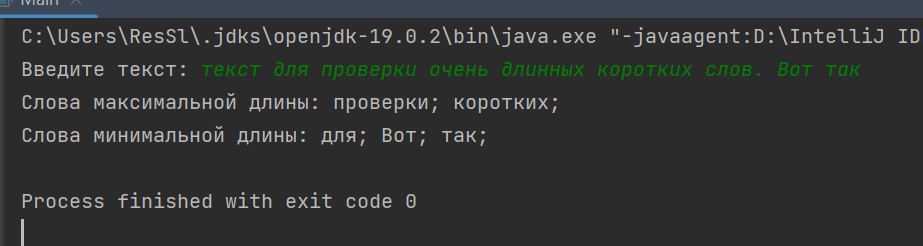


Рисунок 4 – Результат работы программы 1.4

**Задание 1.5**

В предложении из n слов первое слово поставлено на место второго, второе – на место третьего, и т.д., (n-1)-е слово – на место n-го, n-е слово поставлено на место первого. В исходном и преобразованном предложениях между словами один пробел, или знак препинания и один пробел.

Код программы приведен в листинге 5. Результат выполнения программы представлен на рисунке 5.

Листинг 5 – Код класса Main

import java.util.Scanner;  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Введите предложение: ");  
 String sentence = scanner.nextLine();  
  
 String[] words = sentence.split("\\s+");  
  
 String lastWord = words[words.length - 1];  
  
 for (int i = words.length - 1; i >= 1; i--) {  
 words[i] = words[i - 1];  
 }  
  
 words[0] = lastWord;  
  
 String result = String.*join*(" ", words);  
 System.*out*.println("Преобразованное предложение: " + result);  
 }  
}

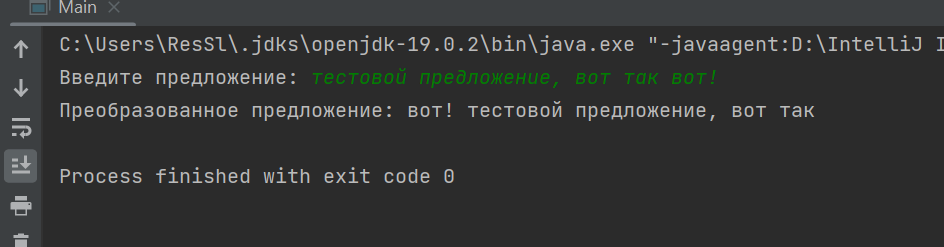


Рисунок 5 – Результат работы программы 1.5

**Задание 1.6**

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. Зашифрованный заданный текст выводится.

Код программы приведен в листинге 6. Результат выполнения программы представлен на рисунке 6.

Листинг 6 – Код класса Main

import java.util.Scanner;  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Введите текст: ");  
 String text = scanner.nextLine();  
  
 StringBuilder resultText = new StringBuilder();  
  
 int n = text.length();  
 for (int i = 0; i < n; i++) {  
 for (int j = i; j < n; j += 3) {  
 resultText.append(text.charAt(j));  
 }  
 }  
  
 System.*out*.println("Зашифрованный текст: " + resultText.toString());  
 }  
}

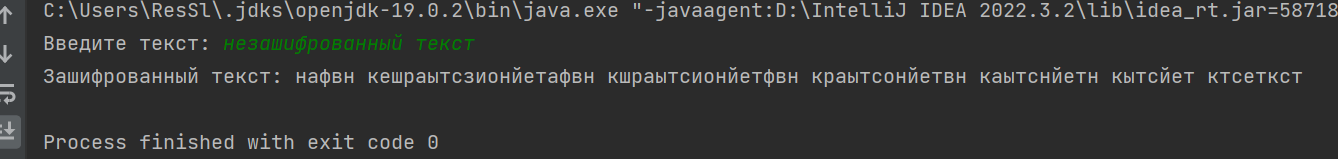


Рисунок 6 – Результат работы программы 1.6

**Задание 1.7**

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

Код программы приведен в листинге 7. Результат выполнения программы представлен на рисунке 7.

Листинг 7 – Код класса Main

import java.util.Arrays;  
import java.util.Comparator;  
import java.util.Scanner;  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Введите текст: ");  
 String text = scanner.nextLine();  
  
 String[] words = text.split("\\s+");  
  
 Comparator<String> alphabetComparator = new Comparator<String>() {  
 @Override  
 public int compare(String s1, String s2) {  
 return s1.compareToIgnoreCase(s2);  
 }  
 };  
  
 Arrays.*sort*(words, alphabetComparator);  
  
 System.*out*.println("Слова в алфавитном порядке: ");  
 for (String word : words) {  
 System.*out*.println(word);  
 }  
 }  
}

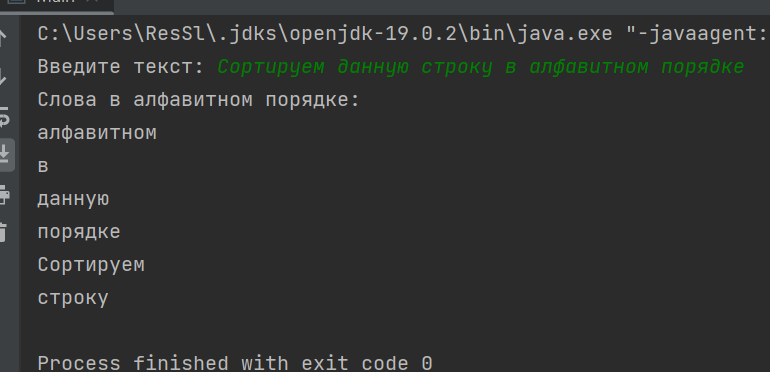


Рисунок 7 – Результат работы программы 1.7

**Задание 1.8**

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

Код программы приведен в листинге 8. Результат выполнения программы представлен на рисунке 8.

Листинг 8 – Код класса Main

import java.util.Scanner;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите текст:");  
 String text = scanner.nextLine();  
  
 Pattern pattern = Pattern.*compile*("\\b[a-z][a-zA-Z]\*\\b");  
 Matcher matcher = pattern.matcher(text);  
 int count = 0;  
 while (matcher.find()) {  
 count++;  
 }  
  
 System.*out*.println("Количество слов, начинающихся с прописной буквы: " + count);  
 }  
}

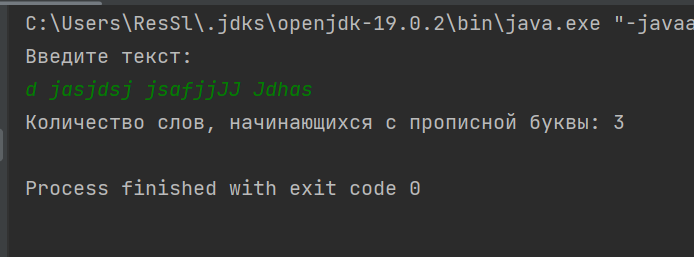


Рисунок 8 – Результат работы программы 1.8

**Вывод**: В результате выполнения лабораторной работы были получены практические навыки для работы с регулярными выражениями и строками.