



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, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И
СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Интеллектуальные системы анализа,
обработки и интерпретации больших данных

ОТЧЕТ

по лабораторной работе № 7

Вариант 13

Название: Строки и регулярные выражения

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-22М

(Группа)

(Подпись, дата)

В.А.Ловцов

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2024

Цель: изучить работу со строками и регулярными выражениями.

Задание 1: в тексте после буквы Р, если она не последняя в слове, ошибочно напечатана буква А вместо О. Внести исправления в текст.

Код:

```
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Third_1 {
    public static String replaceLetterAfterR(String text)
    {
        Pattern pattern = Pattern.compile("pa",
Pattern.CASE_INSENSITIVE);
        Matcher matcher = pattern.matcher(text);

        StringBuffer correctedText = new StringBuffer();
        while (matcher.find()) {
            matcher.appendReplacement(correctedText,
"po");
        }
        matcher.appendTail(correctedText);

        return correctedText.toString();
    }

    public static void main(String[] args) {
        String text = "partenship asap after apartment
past";
        // String text = "Partenship ASAP after apArtment
PAst";
        // String text = "Пример текста с ошибками: Раз РАЗ
РА арАбота каР А";
```

```

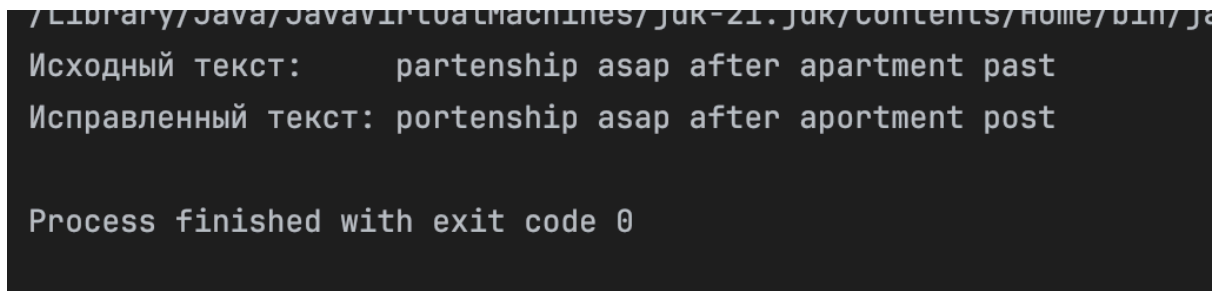
        String correctedText = replaceLetterAfterR(text);

        System.out.print("Исходный текст:\t\t");
        System.out.println(text);
        System.out.print("Исправленный текст:\t");
        System.out.println(correctedText);
    }

}

```

Работа программы показана на рисунке 1.



```

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java
Исходный текст:      partenship asap after apartment past
Исправленный текст: portenship asap after aportment post

Process finished with exit code 0

```

Рисунок 1 – Работа программы

Задание 2: в тексте слова заданной длины заменить указанной подстрокой, длина которой может не совпадать с длиной слова.

Код:

```

import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Fourth_1 {
    public static void main(String[] args) {
        String text = " this line tests regex algo";
        int wordLength = 5;
        String replacement = "*****";
    }
}

```

```

        Pattern pattern = Pattern.compile("\\b\\w{" +
wordLength + "}\\b");
        Matcher matcher = pattern.matcher(text);

        StringBuffer result = new StringBuffer();
        while (matcher.find()) {
            matcher.appendReplacement(result,
replacement);
        }
        matcher.appendTail(result);

        System.out.println(result.toString());
    }
}

```

Работа программы показана на рисунке 2.

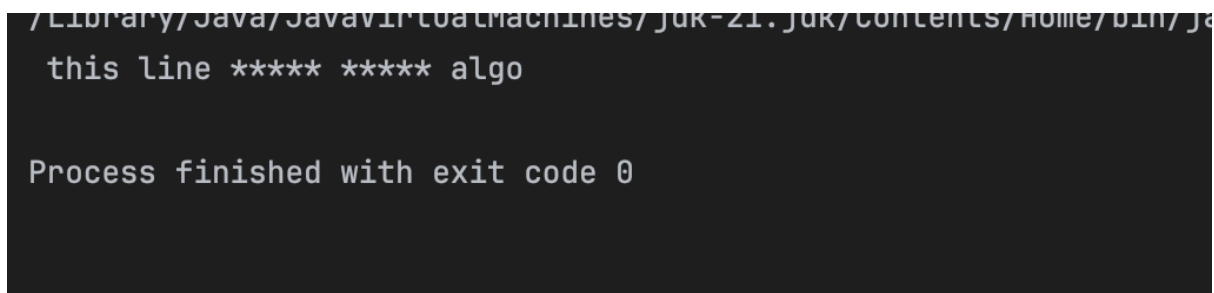


Рисунок 2 – Работа программы

Задание 3: в тексте найти и напечатать n символов (и их количество), встречающихся наиболее часто.

Код:

```

import java.util.HashMap;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

```

```

public class Third_2 {
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter n: ");
        int n = scanner.nextInt();
        String text = "Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Sed do eiusmod tempor incididunt ut labore et
dolore magna aliqua.";

        Map<Character, Integer> charFrequency = new
HashMap<>();
        Pattern pattern = Pattern.compile("\\S");
        Matcher matcher = pattern.matcher(text);

        while (matcher.find()) {
            char c = matcher.group().charAt(0);
            charFrequency.put(c, charFrequency.getOrDefault(c,
0) + 1);
        }

        for (int i = 0; i < n; i++) {
            int maxCount = 0;
            char mostFrequentChar = ' ';
            for (Entry<Character, Integer> entry :
charFrequency.entrySet()) {
                if (entry.getValue() > maxCount) {
                    maxCount = entry.getValue();
                    mostFrequentChar = entry.getKey();
                }
            }
        }
    }
}

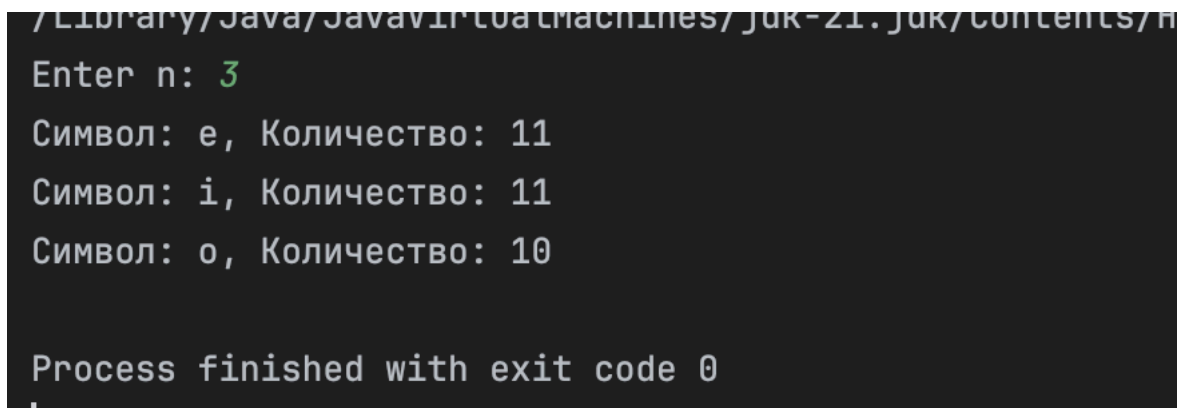
```

```

        System.out.println("Символ: " + mostFrequentChar +
", Количество: " + maxCount);
        charFrequency.remove(mostFrequentChar);
    }
}
}

```

Работа программы показана на рисунке 3.



```

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home
Enter n: 3
Символ: е, Количество: 11
Символ: і, Количество: 11
Символ: о, Количество: 10

Process finished with exit code 0

```

Рисунок 3 – Работа программы

Задание 4: найти, каких букв, гласных или согласных, больше в каждом предложении текста.

Код:

```

import java.util.HashMap;
import java.util.Map;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Fourth_2 {
    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.";
    }
}

```

```

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

for (String sentence : sentences) {
    Map<Character, Integer> vowelFrequency = new
HashMap<>();
    Map<Character, Integer> consonantFrequency = new
HashMap<>();
    Pattern pattern = Pattern.compile("[a-zA-Z]");
    Matcher matcher = pattern.matcher(sentence);

    while (matcher.find()) {
        char c =
Character.toLowerCase(matcher.group().charAt(0));
        if ("aeiou".contains(String.valueOf(c))) {
            vowelFrequency.put(c,
vowelFrequency.getOrDefault(c, 0) + 1);
        } else {
            consonantFrequency.put(c,
consonantFrequency.getOrDefault(c, 0) + 1);
        }
    }

    int vowelCount =
vowelFrequency.values().stream().mapToInt(Integer::intValue).s
um();

    int consonantCount =
consonantFrequency.values().stream().mapToInt(Integer::intValu
e).sum();

    System.out.println("Предложение: " + sentence);
    if (vowelCount > consonantCount) {
        System.out.println("Больше гласных букв");
    } else if (consonantCount > vowelCount) {

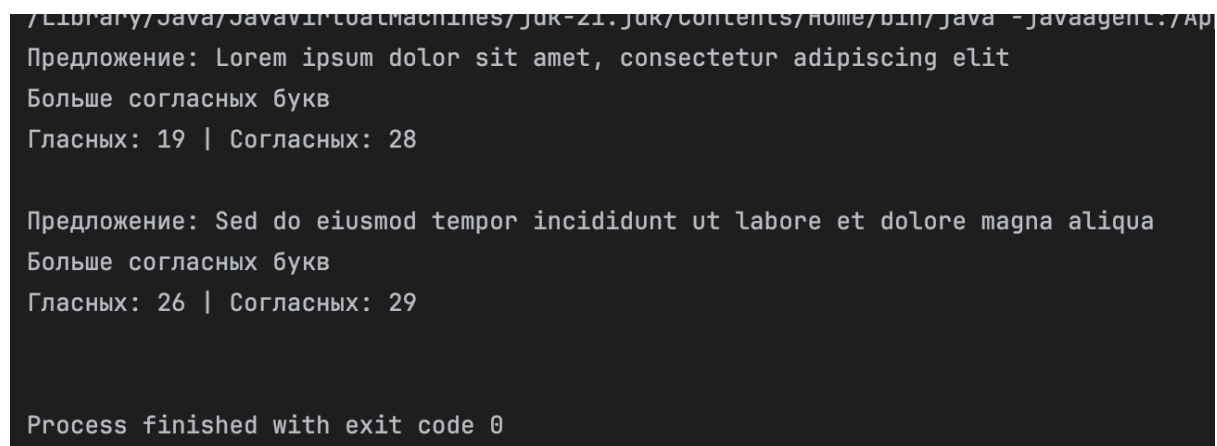
```

```

        System.out.println("Больше согласных букв");
    } else {
        System.out.println("Гласных и согласных букв
одинаковое количество");
    }
    System.out.println("Гласных: " + vowelCount + " |
Согласных: " + consonantCount);
    System.out.println();
}
}
}

```

Работа программы показана на рисунке 4.



```

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:7Ar
Предложение: Lorem ipsum dolor sit amet, consectetur adipiscing elit
Больше согласных букв
Гласных: 19 | Согласных: 28

Предложение: Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua
Больше согласных букв
Гласных: 26 | Согласных: 29

Process finished with exit code 0

```

Рисунок 4 – Работа программы

Задание 5: найти такое слово в первом предложении, которого нет ни в одном из остальных предложений.

Код:

```

import java.util.HashSet;
import java.util.Set;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

```



```

public class Third_3 {
    public static void main(String[] args) {
        String text = "Java is a popular akk programming
language. It is used for developing various applications. Java
provides many features for developers.";

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

        Set<String>    uniqueWordsFirstSentence    =    new
HashSet<>();
        Set<String>    uniqueWordsOtherSentences    =    new
HashSet<>();

        // Находим уникальные слова в первом предложении
        Pattern                wordPattern                =
Pattern.compile("\\b\\w+\\b");
        Matcher                matcher                =
wordPattern.matcher(sentences[0]);
        while (matcher.find()) {

uniqueWordsFirstSentence.add(matcher.group().toLowerCase());

        }

        // Находим уникальные слова в остальных
предложениях
        for (int i = 1; i < sentences.length; i++) {
            matcher = wordPattern.matcher(sentences[i]);
            while (matcher.find()) {

uniqueWordsOtherSentences.add(matcher.group().toLowerCase());

            }
        }
    }
}

```

```

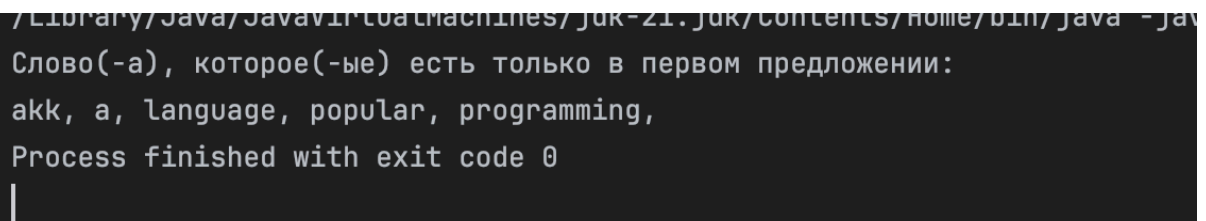
        // Находим слово, которое есть только в первом
        предложении

        uniqueWordsFirstSentence.removeAll(uniqueWordsOtherSentences);

        if (!uniqueWordsFirstSentence.isEmpty()) {
            System.out.println("Слово(-а), которое(-ые)
        есть только в первом предложении: ");
            for (String element :
        uniqueWordsFirstSentence) {
                System.out.print(element + ", ");
            }
            // uniqueWordsFirstSentence.iterator().next();
        } else {
            System.out.println("В первом предложении нет
        слов, которых нет ни в одном из остальных предложений.");
        }
    }
}

```

Работа программы показана на рисунке 5.



```

/ Library/Java/JavaVM-Toolchains/jdk-21.jdk/Contents/Home/bin/java -jav
Слово(-а), которое(-ые) есть только в первом предложении:
akk, a, language, popular, programming,
Process finished with exit code 0
|

```

Рисунок 5 – Работа программы

Задание 6: во всех вопросительных предложениях текста найти и напечатать без повторений слова заданной длины.

Код:

```

import java.util.HashSet;
import java.util.Scanner;

```

```

import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Fourth_3 {
    public static HashSet<String> findUniqueWords(String
text, int wordLength) {
        HashSet<String> uniqueWords = new HashSet<>();

        Pattern                sentencePattern          =
Pattern.compile("[^?!\\.] *\\?");
        Matcher                sentenceMatcher          =
sentencePattern.matcher(text);

        while (sentenceMatcher.find()) {
            String sentence = sentenceMatcher.group();
            Pattern                wordPattern            =
Pattern.compile("\\b\\w{" + wordLength + "}\\b");
            Matcher                wordMatcher            =
wordPattern.matcher(sentence);

            while (wordMatcher.find()) {
                uniqueWords.add(wordMatcher.group());
            }
        }

        return uniqueWords;
    }

    public static void main(String[] args) {
        String text = "What is your name? How are you doing?
Where do you live? Java is a popular akk programming language";

        //          int wordLength = 3; // Заданная длина слова

```



```

        //          System.out.print("Enter n: ");
        //          int wordLength = scanner.nextInt();
        //          for (String sent : sentences) {
        //                      Pattern pattern =
Pattern.compile("^.+(?:\\?\\?\\$)");
        //          Matcher matcher = pattern.matcher(sent);
        //          if (matcher.matches()) {
        //                      Pattern wordPattern =
Pattern.compile("\\b(\\w+)\\b");
        //          Matcher wordMatcher =
wordPattern.matcher(sent);
        //
        //          while (wordMatcher.find()) {
        //                      String word = wordMatcher.group(1);
        //                      if (word.length() == wordLength) {
        //                      uniqueWords.add(word.toLowerCase());
        //                      }
        //          }
        //          }
        //
        //          }
        //
        //          if (!uniqueWords.isEmpty()) {
        //          System.out.println("Слова заданной длины без
повторений в вопросительных предложениях:");
        //          for (String word : uniqueWords) {
        //          System.out.println(word);
        //          }
        //          } else {
        //          System.out.println("В вопросительных
предложениях нет слов заданной длины.");
        //          }
        //          }

```

```
}
```

Работа программы показана на рисунке 6.

```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -ja
Enter n: 3
Уникальные слова заданной длины во всех вопросительных предложениях:
How
are
you

Process finished with exit code 0
```

Рисунок 6 – Работа программы

Задание 7: 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:

```
import java.util.*;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Third_4 {
    public static List<String>
sortWordsByLengthAndVowels(String text) {
        List<String> words = new ArrayList<>();
        Pattern wordPattern = Pattern.compile("\\b\\w+\\b");
        Matcher matcher = wordPattern.matcher(text);

        while (matcher.find()) {
            words.add(matcher.group());
        }

        words.sort(new Comparator<String>() {
            @Override
```

```

        public int compare(String word1, String word2) {
            if (word1.length() != word2.length()) {
                return Integer.compare(word2.length(),
word1.length());
            } else {
                int vowelsCount1 = countVowels(word1);
                int vowelsCount2 = countVowels(word2);
                return Integer.compare(vowelsCount1,
vowelsCount2);
            }
        }
    });

    return words;
}

public static int countVowels(String word) {
    int count = 0;
    Pattern vowelPattern = Pattern.compile("[aeiouy]",
Pattern.CASE_INSENSITIVE);
    Matcher matcher = vowelPattern.matcher(word);

    while (matcher.find()) {
        count++;
    }

    return count;
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Введите исходный текст:");
    String text = scanner.nextLine();

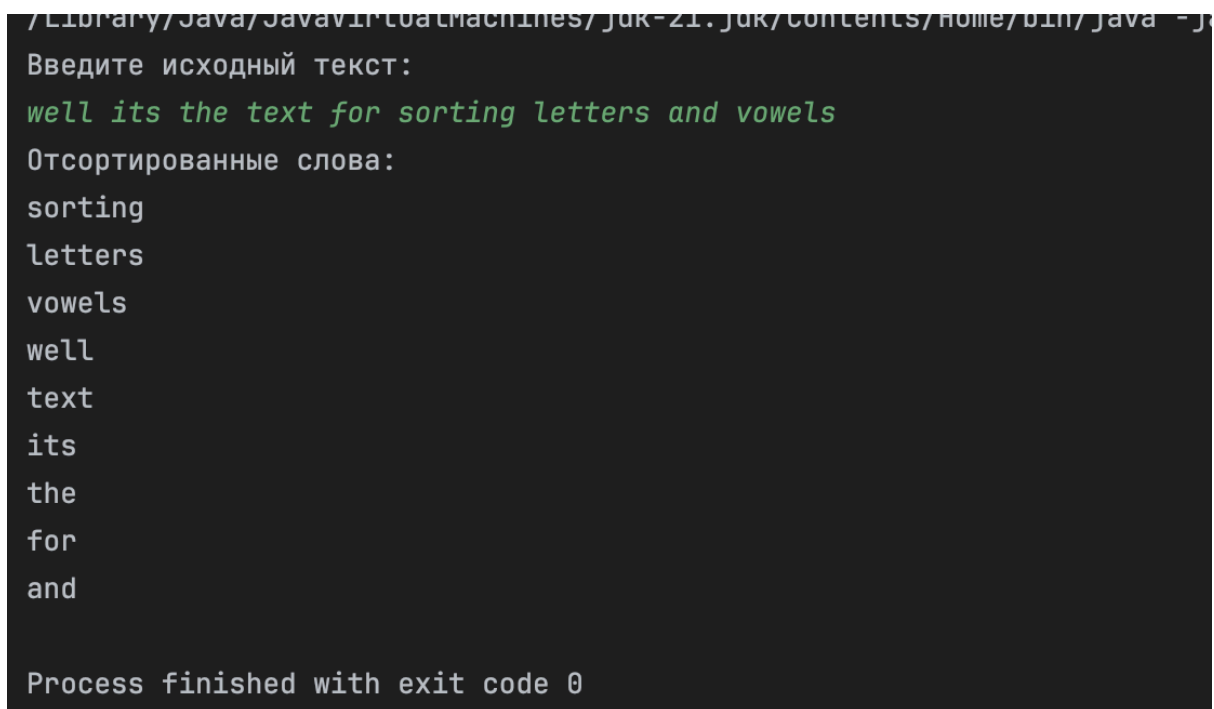
```

```
//      String text = "You can use the text for sortiAg
vowel letters";

      List<String> sortedWords =
sortWordsByLengthAndVowels(text);

      System.out.println("Отсортированные слова:");
      for (String word : sortedWords) {
          System.out.println(word);
      }
  }
}
```

Работа программы показана на рисунке 7.



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -J
Введите исходный текст:
well its the text for sorting letters and vowels
Отсортированные слова:
sorting
letters
vowels
well
text
its
the
for
and

Process finished with exit code 0
```

Рисунок 7 – Работа программы

Задание 8: в тексте исключить подстроку максимальной длины, начинающуюся и заканчивающуюся заданными символами.

Код:

```
import java.util.Scanner;
import java.util.regex.Matcher;
```



```

import java.util.regex.Pattern;

public class Fourth_4 {
    public static String delMaxSub(String text, char
startChar, char endChar) {
        Pattern pattern = Pattern.compile("\\\\" + startChar +
"(..*?)" + "\\" + endChar);
        Matcher matcher = pattern.matcher(text);

        int maxLength = 0;
        String maxSubstring = "";

        while (matcher.find()) {
            String substring = matcher.group(1);
            if (substring.length() > maxLength) {
                maxLength = substring.length();
                maxSubstring = substring;
            }
        }

        return text.replace(startChar + maxSubstring +
endChar, "");
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Введи стартовый символ: ");
        char startChar = scanner.next().charAt(0);
        System.out.print("Введи стартовый символ: ");
        char endChar = scanner.next().charAt(0);
        //      char startChar = '('; // Заданный символ начала
        подстроки
    }
}

```

```

//          char endChar = ')'; // Заданный символ конца
подстроки
        scanner.nextLine();
        System.out.println("Введите исходный текст:");
        String text = scanner.nextLine();
//          String text = "We need to drop from this (short)
text (the longest) substring";
//          System.out.println(text);

        String modifiedText = delMaxSub(text, startChar,
endChar);

        System.out.println("Текст после исключения
подстроки:");
        System.out.println(modifiedText);
    }
}

```

Работа программы показана на рисунке 8.

```

C:\Library\Java\Javavirtualmachines\jdk-21.jdk\contents\bin
Введи стартовый символ: [
Введи стартовый символ: ]
Введите исходный текст:
you shoul [change this text] to snth else
Текст после исключения подстроки:
you shoul to snth else

Process finished with exit code 0

```

Рисунок 8 – Работа программы

Вывод: во время выполнения лабораторной работы была изучена работа со строками и регулярными выражениями.