



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 Информатика и вычислительная техника

МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных

О Т Ч Е Т

по лабораторной работе №7

Название: Строки, Регулярные выражения

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-22М

(Группа)

(Подпись, дата)

В.А. Трофимов

(И.О. Фамилия)

Преподаватель

П.В. Степанов

(Подпись, дата)

(И.О. Фамилия)

Москва, 2023

Задания:

1.6. После каждого слова текста, заканчивающегося заданной подстрокой, вставить указанное слово.

1.7. В зависимости от признака (0 или 1) в каждой строке текста удалить указанный символ везде, где он встречается, или вставить его после k-го символа.

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

2.7. В тексте найти и напечатать все слова максимальной и все слова минимальной длины.

3.6. 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.

3.7. 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.

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

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

Код для решения задания 1.6:

```
package bdjava.lab7;

import static java.lang.System.out;

//6. После каждого слова текста, заканчивающегося заданной подстрокой, вставить
указанное слово.
public class lab7_1 {
    public static void main(String[] args) {
        String text = "qsqs sa sdwdsa saeed a sxsa";
        String pattern = "\\b\\w+sa\\b";
        String newWord = "СЛОВО";
        String newText = text.replaceAll(pattern, "$0 " + newWord);

        out.println(newText);
    }
}
```

Код для решения задания 1.7:

```
package bdjava.lab7;

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

//7. В зависимости от признака (0 или 1)
// в каждой строке текста удалить указанный символ везде, где он встречается, или
// вставить его после k-го символа.
public class lab7_2 {
    public static void main(String[] args) {
        String text = "d3d3d3d3d3cccdsaaw33";
        char symbol = '3';
        // это признак
        boolean sign = true;
        int k = 3;

        if (sign) {
            String regex = String.valueOf(symbol);
            String res = text.replaceAll(regex, "");
            System.out.println("Результат: " + res);
        } else {
            String regex = "(.{1," + k + "})";
            String replacement = "$1" + symbol;
            String res = text.replaceAll(regex, replacement);
            System.out.println("Результат: " + res);
        }
    }
}
```

Код для решения задания 2.6:

```
package bdjava.lab7;

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

import static java.lang.System.out;

//6. Напечатать без повторения слова текста, у которых первая и последняя буквы
// совпадают.
public class lab7_3 {
    public static void main(String[] args) {
        String text = "dwed www dwws wdd dd";
        Pattern p = Pattern.compile("\\b(\\w)\\w*\\1\\b");
        Matcher m = p.matcher(text);
        while (m.find()) {
            String word = m.group();
            out.println(word);
        }
    }
}
```

Код для решения задания 2.7:

```
package bdjava.lab7;

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

import static java.lang.System.out;

//7. В тексте найти и напечатать все слова максимальной и все слова минимальной
длинны.
public class lab7_4 {
    public static void main(String[] args) {
        String text = "rwevrgvb3we qwsd sh s wefth4 qw i78 f";
        Pattern pattern = Pattern.compile("\\b\\w+\\b");

        Matcher matcher = pattern.matcher(text);

        int minLength = text.length();
        int maxLength = 0;
        String min = "";
        String max = "";

        int len;
        while (matcher.find()) {
            String word = matcher.group();
            len = word.length();
            if (len < minLength) {
                minLength = len;
                min = word;
            }
            if (len > maxLength) {
                maxLength = len;
                max = word;
            }
        }

        out.println("Минимальное слово: " + min);
        out.println("Максимальное слово: " + max);
    }
}
```

Код для решения задания 3.6:

```
package bdjava.lab7;

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

import static java.lang.System.out;

//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 Part5 {
    public static void main(String[] args) {
        String sentence = "qwe sdf hjk zxc";
        String regex = "\\b(\\w+\\b)";
        Pattern pattern = Pattern.compile(regex);
        Matcher matcher = pattern.matcher(sentence);
        StringBuilder result = new StringBuilder();
        while (matcher.find()) {
            result.append(matcher.group()).append(" ");
        }
        String[] words = result.toString().trim().split(" ");
        String firstWord = words[0];
        for (int i = 0; i < words.length - 1; i++) {
            words[i] = words[i + 1];
        }
        words[words.length - 1] = firstWord;
        String transformedSentence = String.join(" ", words);
        out.println(transformedSentence);
    }
}
```

Код для решения задания 3.7:

```
package bdjava.lab7;

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

import static java.lang.System.out;

///7.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.
public class Part6 {
    public static void main(String[] args) {
        String text = "This is text to be encrypted.";
        out.println("Оригинальный текст: " + text);
        String encryptedText = encryptText(text);
        out.println("Закодированный текст: " + encryptedText);
    }

    public static String encryptText(String text) {
        Pattern pattern = Pattern.compile("(?<=.)");
        Matcher matcher = pattern.matcher(text);
        StringBuilder[] encryptedChars = new StringBuilder[3];

        encryptedChars[0] = new StringBuilder();
        encryptedChars[1] = new StringBuilder();
        encryptedChars[2] = new StringBuilder();

        int count = 0;
        while (matcher.find()) {
```

```

        out.println();
        count++;
        int index = count % 3;
        encryptedChars[index].append(matcher.group());
    }
    StringBuilder sb = new StringBuilder();
    sb.append(encryptedChars[0]);
    sb.append(encryptedChars[1]);
    sb.append(encryptedChars[2]);
    return sb.toString();
}
}

```

Код для решения задания 4.6:

```

package bdjava.lab7;

import java.util.Arrays;

import static java.lang.System.out;

//6.Вывести в заданном тексте все слова, расположив их в алфавитном порядке.
public class Part7 {
    public static void main(String[] args) {
        String text = "это пример текста, который нужно обработать и вывести все слова в алфавитном порядке.";
        String[] words = text.split("[\\P{L}]+");
        Arrays.sort(words);
        for (String word : words) {
            out.println(word);
        }
    }
}

```

Код для решения задания 4.7:

```

package bdjava.lab7;

import static java.lang.System.out;

public class Part8 {
    public static void main(String[] args) {
        String text = "Жццв РРт Дц ц иа";
        String[] words = text.split("\\s+");
        int count = 0;
        for (String word : words) {
            if (word.matches("\\p{Lu}\\p{L}*")) {
                count++;
                out.println(word);
            }
        }
        out.println("Количество слов, начинающихся с прописной буквы: " + count);
    }
}

```

```
}
```

Вывод:

В ходе выполнения данных заданий были применены различные методы работы с текстом, такие как поиск и замена подстрок, разбиение текста на слова, сортировка и подсчет слов. Были использованы различные структуры данных, такие как массивы и списки. Кроме того, задания позволили попрактиковаться в работе с циклами и условными операторами. В результате выполнения заданий были получены различные результаты, такие как измененный текст, список слов без повторений, количество слов с прописной буквы и т.д. В целом, выполнение данных заданий позволило улучшить навыки работы с текстом и программирования в целом.