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.**

О Т Ч Е Т

по лабораторной работе № 7

Вариант 6

Название: Строки и регулярные выражения

Дисциплина: Языки программирования для работы с большими данными

Студент ИУ6-23М (Группа) _____ (Подпись, дата) В.А. Елисеев (И.О. Фамилия)

Преподаватель _____ (Подпись, дата) П.В. Степанов (И.О. Фамилия)

Москва, 2022

Цель работы: получение навыков работы со строками и регулярными выражениями в Java.

Задание 1:

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

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.

Подзадача 1.

Код программы:

```java
public class App1 {
    Run | Debug
    public static void main(String[] args) throws Exception {
        String text = "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, qui

        String suffix = args[0]; // То, что заменяем
        String change = args[1]; // То, на что заменяем

        Matcher m = Pattern.compile(suffix + "[ .,!?:;]").matcher(text);

        ArrayList<Integer> list = new ArrayList<Integer>();

        while (m.find()) {
            list.add(m.end() - 1);
        }

        Collections.reverse(list);

        for (int idx : list) {
            text = text.substring(beginIndex: 0, idx) + change + text.substring(idx);
        }

        System.out.println(text);
    }
}
```

Результат выполнения программы:

```
um slava
Lorem ipsumslava dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
 commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillumslava dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit
 anim id est laborumslava.
```

Подзадача 2.

Код программы:

```java
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;

public class App2 {
    Run | Debug
    public static void main(String[] args) throws IOException {
        BufferedWriter writer = new BufferedWriter(new FileWriter(fileName: "out_text.txt"));

        int rule = Integer.parseInt(args[0]);
        String symbol = args[1];
        int k = 0;

        if (args.length == 3) k = Integer.parseInt(args[2]);

        try (BufferedReader br = new BufferedReader(new InputStreamReader(new FileInputStream(new File(pathname: "text.txt")), charsetName: "UTF-8"))) {
            for (String line; (line = br.readLine()) != null;) {
                switch (rule) {
                    case 0:
                        writer.write(line.replaceAll(symbol, replacement: "") + "\n");
                    break;

                    default:
                        line = line.substring(beginIndex: 0, k) + symbol + line.substring(k);
                        writer.write(line + "\n");
                    break;
                }
            }
        } catch (Exception e) {
            System.out.println(x: "File not found");
            System.exit(status: 0);
        }

        writer.close();
    }
}
```

Результат выполнения программы:

text.txt

```
1   Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
2   Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
3   Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
4   Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
```

```
0 m
```

out_text.txt

```
1   Lore ipsu dolor sit aet, consectetur adipiscing elit, sed do eiusod tepor incididunt ut labore et dolore agna aliqua.
2   Ut eni ad ini venia, quis nostrud exercitation ullaco laboris nisi ut aliquip ex ea coodo consequat.
3   Duis aute irure dolor in reprehenderit in voluptate velit esse cillu dolore eu fugiat nulla pariatur.
4   Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt ollit ani id est laboru.
```

```
1 x 7
```

out_text.txt

```
1   Lorem ixpsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
2   Ut enimx ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
3   Duis auxte irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
4   Exceptexur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
```

Задание 2:

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

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

Подзадача 1.

Код программы:

```java
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.ArrayList;

public class App1 {
    Run | Debug
    public static void main(String[] args) throws Exception {
        String text = new String(Files.readAllBytes(Paths.get(first: "poem.txt")));
        text = text.replaceAll(regex: "\n", replacement: " ");

        ArrayList<String> list = new ArrayList<String>();
        for (String word : text.split(regex: " ")) {
            if (word.charAt(index: 0) == word.charAt(word.length() - 1) && !list.contains(word)) list.add(word);
        }

        System.out.println(list);
    }
}
```

Результат работы программы:

poem.txt

```
Mary had a little lamb,
Little lamb, little lamb,
Mary had a little lamb,
Its fleece was white as snow,
And every where that Mary went,
Mary went, Mary went,
Everywhere that Mary went,
The lamb was sure to go;
He followed her to school one day,
School one day, school one day,
He followed her to school one day,
Which was against the rule;
It made the children laugh and play,
Laugh and play, laugh and play,
It made the children laugh and play,
To see a lamb at school,
And so the teacher turned him out,
Turned him out, turned him out,
So the teacher turned him out,
But still he lingered near,
And waited patiently about,
Patiently about, patiently about,
Waited patiently about,
Till Mary did appear;
“Why does the lamb love Mary so?
Mary so, Mary so,
Why does the lamb love Mary so?”
The eager children cried;
“Why Mary loves the lamb, you know,
Lamb you know, lamb you know,
Why Mary loves the lamb, you know”
The teacher did reply;
Mary had a little lamb,
Little lamb, little lamb,
Mary had a little lamb,
Its fleece was white as snow.
```

`[a, that, did]`

Подзадача 2.

Код программы:

```java
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class App2 {
    Run | Debug
    public static void main(String[] args) throws IOException {
        String text = new String(Files.readAllBytes(Paths.get(first: "text.txt")));
        text = text.replaceAll(regex: "\n", replacement: " ");
        Matcher m = Pattern.compile(regex: "[^A-z][A-z]+[^A-z]").matcher(text);

        int minLen = 99999;
        int maxLen = 0;

        while (m.find()) {
            String word = m.group().replaceAll(regex: "[ .,!?;:\n]", replacement: "");

            if (word.length() < minLen) minLen = word.length();
            if (word.length() > maxLen) maxLen = word.length();
        }

        Matcher mMin = Pattern.compile("[^A-z][A-z]{" + minLen + "}[^A-z]").matcher(text);
        Matcher mMax = Pattern.compile("[^A-z][A-z]{" + maxLen + "}[^A-z]").matcher(text);

        ArrayList<String> listMin = new ArrayList<String>();
        ArrayList<String> listMax = new ArrayList<String>();

        while (mMin.find()) {
            String word = mMin.group().replaceAll(regex: "[ .,!?;:\n]", replacement: "");;
            if (!listMin.contains(word)) listMin.add(word);
        }

        while (mMax.find()) {
            String word = mMax.group().replaceAll(regex: "[ .,!?;:\n]", replacement: "");;
            if (!listMax.contains(word)) listMax.add(word);
        }

        System.out.println(listMin);
        System.out.println(listMax);
    }
}
```

Результат работы программы:

text.txt

```
The Matoran or Bionicle alphabet is used in the Lego series Bionicle. It is used to write things in the constructed language, Matoran, and can also be used to write in English.
All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.
(Article 1 of the Universal Declaration of Human Rights)
```

```
[a]
[constructed, brotherhood, Declaration]
```

Задание 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.

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.

Подзадача 1.

Код программы:

```java
public class App1 {
    public static void main(String[] args) throws Exception {
        String sentence = "На клетчатом листе бумаги закрашена часть клеток. Выделить все различные фигуры, которые образовались при этом. Фигурой считается набор закрашенных кл

        LinkedList<String> list = new LinkedList<String>();

        for (String word : sentence.split(" ")) {
            list.add(word);
        }

        list.add(list.removeFirst());

        System.out.println(String.join(" ", list));
    }
}
```

Результат работы программы:

```
клетчатом листе бумаги закрашена часть клеток. Выделить все различные фигуры, которые образовались при этом. Фигурой считается набор закрашенных клеток, достижимых друг из друга при движении в четырёх направлениях. Две фигуры являются различными, если их нельзя совместить поворотом на угол, кратный 90 градусам, и параллельным переносом. Используйте класс HashSet. На
```

Подзадача 2.

Код программы:

```java
public class App2 {
    Run | Debug
    public static void main(String[] args) {
        String text = "На плоскости задано N отрезков. Найти точку пересечения двх отрезков, имеющую минимальную абсциссу. Использовать класс TreeMap. ";
        String codedText = "";

        for (int i = 0; i < text.length(); i += 3) {
            codedText += text.charAt(i);
        }

        for (int i = 1; i < text.length(); i += 3) {
            codedText += text.charAt(i);
        }

        for (int i = 2; i < text.length(); i += 3) {
            codedText += text.charAt(i);
        }

        System.out.println("Начальный текст: " + text);
        System.out.println("Закодированный текст: " + codedText);
    }
}
```

Результат работы программы:

```
Начальный текст: На плоскости задано N отрезков. Найти точку пересечения двх отрезков, имеющую минимальную абсциссу. Использовать класс TreeMap.
Закодированный текст: Нпсс до рк.аиоуесеявоео еумилуацс пьвьлсгМ.алктза оео й ч рен хтзвиюсимыобиуИоза а еа  ооианNтзвНтткпечид рк,мщ нан сс.слоткстер
```

Задание 4:

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

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

Подзадача 1.

Код программы:

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

public class App1 {
    Run | Debug
    public static void main(String[] args) throws Exception {
        String text = "На клетчатом листе бумаги закрашена часть клеток. Выделить все различные фигуры, которые образовались при этом. Фигурой считается набор закрашенных клето
        Matcher m = Pattern.compile(regex: "[^А-я]*[А-я]+[^А-я]*").matcher(text);

        ArrayList<String> list = new ArrayList<String>();

        while (m.find()) {
            System.out.println(m.group());
            String word = m.group().replaceAll(regex: "[ .,?!;:]", replacement: "");;
            list.add(word);
        }

        Collections.sort(list);
        System.out.println(list);
    }
}
```

Результат работы программы:

```
[Выделить, Две, Используйте, На, Фигурой, бумаги, в, все, градусам, движении, достижимых, друг, друга, если, закрашена, закрашенных, и, из, их, классHashSet, клеток, клеток, клетчатом, которые, кратный90, листе, на, н
абор, направлениях, нельзя, образовались, параллельным, переносом, поворотом, при, при, различные, различными, совместить, считается, угол, фигуры, фигуры, х, часть, четырё, этом, являются]
```

Подзадача 2.

Код программы:

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

public class App2 {
    Run | Debug
    public static void main(String[] args) {
        String text = "А Вы Бы Не Хотели";
        Matcher m = Pattern.compile(regex: "[^А-я]*[А-Я][А-я]*[^А-я]*").matcher(text);

        int a = 0;
        while (m.find()) {
            a++;
        }
        System.out.println(a);
    }
}
```

Результат работы программы:

```
5
```

Ссылка на программное решение:

https://github.com/ArMaxik/BigDataLanguages/tree/main/lr7

Вывод: в ходе лабораторной работы были получены навыки работы со строками и регулярными выражениями в Java.