



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.

**О Т Ч Е Т**

по лабораторной работе № 7

Вариант № 4

**Название:** строки и регулярные выражения

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

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

А.А.Клушина

(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 слова заданной длины заменить указанной подстрокой, длина которой может не совпадать с длиной слова.

Код класса Main:

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

public class WordReplacement {

    public static void main(String[] args) {
        String text = "Это моя седьмая лабораторная по
python";

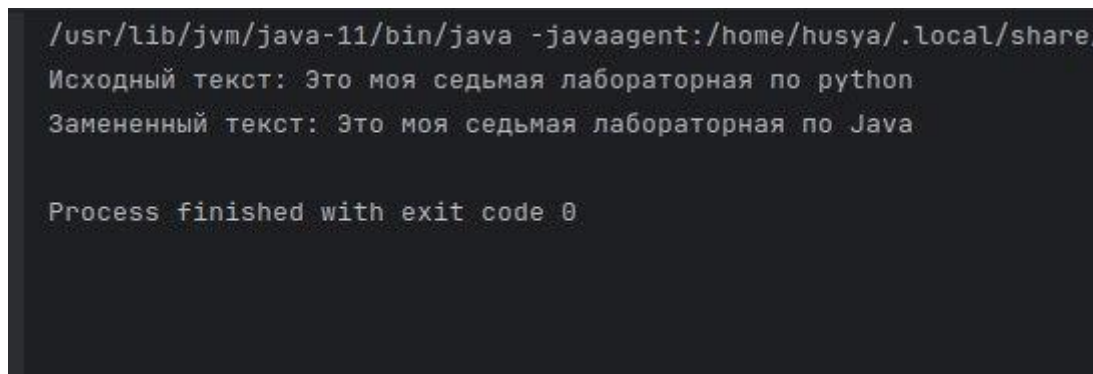
        int n = 6; // Длина слов, которые нужно заменить
        String replacement = "Java"; // Подстрока, на
которую нужно заменить
        String regex = "\\b\\w{" + n + "}\\b";

        Pattern pattern = Pattern.compile(regex);
        Matcher matcher = pattern.matcher(text);

        String replacedText =
matcher.replaceAll(replacement);

        System.out.println("Исходный текст: " + text);
        System.out.println("Замененный текст: " +
replacedText);
    }
}
```

Работа программы показана на рисунке 1.



```
/usr/lib/jvm/java-11/bin/java -javaagent:/home/husya/.local/share
Исходный текст: Это моя седьмая лабораторная по python
Замененный текст: Это моя седьмая лабораторная по Java

Process finished with exit code 0
```

Рисунок 1 – Работа программы

**Задание 2:** Найти, каких букв, гласных или согласных, больше в каждом предложении текста.

Код класса Main:

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

public class VowelConsonantCounter {
    public static void main(String[] args) {
        String text = "Это предложение номер один. Это предложение номер два.";
        Pattern pattern = Pattern.compile("\\p{Punct}");
        String[] sentences = text.split(pattern.pattern());
        // Разделение текста на предложения

        // Проходим по каждому предложению и подсчитываем количество гласных и согласных букв
        for (String sentence : sentences) {
            int vowelsCount = 0;
            int consonantsCount = 0;

            // Приводим предложение к нижнему регистру для удобства подсчета
```

```

String lowercaseSentence =
sentence.toLowerCase();

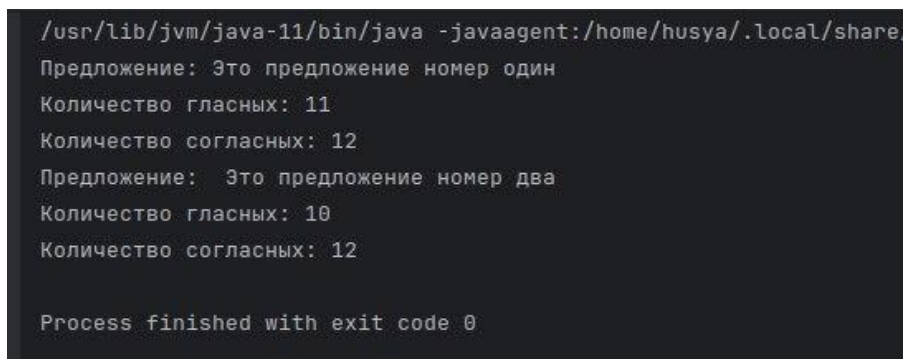
// Подсчет гласных и согласных букв
for (char c : lowercaseSentence.toCharArray()) {
    if (c == 'a'  c == 'e'  c == 'i'  c == 'o'
c == 'u'  c == 'y'  c == 'а'  c == 'о'  c == 'и'  c == 'е'  c ==
'ё'  c == 'э'  c == 'ы'  c == 'у'  c == 'ю'  c == 'я') {
        vowelsCount++;
    } else if (c >= 'a' && c <= 'z'  c >= 'а' &&
c <= 'я') {
        consonantsCount++;
    }
}

System.out.println("Предложение: " + sentence);
System.out.println("Количество гласных: " +
vowelsCount);

System.out.println("Количество согласных: " +
consonantsCount);
}
}
}

```

Работа программы показана на рисунке 2.



```

/usr/lib/jvm/java-11/bin/java -javaagent:/home/husya/.local/share
Предложение: Это предложение номер один
Количество гласных: 11
Количество согласных: 12
Предложение: Это предложение номер два
Количество гласных: 10
Количество согласных: 12

Process finished with exit code 0

```

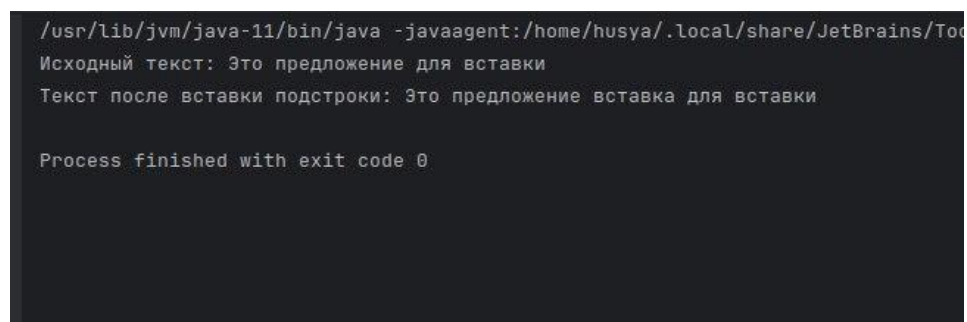
Рисунок 2 – Работа программы

**Задание 3:** В тексте после k-го символа вставить заданную подстроку.

### Код класса Main:

```
public class SubstringInsertion {  
    public static void main(String[] args) {  
        String text = "Это предложение для вставки";  
        int k = 16; // Позиция, после которой нужно вставить  
        подстроку  
        String substringToInsert = "вставка "; // Заданная  
        подстрока для вставки  
  
        // Проверка, что k находится в пределах длины текста  
        if (k < 0 || k > text.length()) {  
            System.out.println("Ошибка: позиция вставки  
        находится вне диапазона текста.");  
            return;  
        }  
  
        String newText = text.substring(0, k) +  
        substringToInsert + text.substring(k);  
  
        System.out.println("Исходный текст: " + text);  
        System.out.println("Текст после вставки подстроки: "  
        + newText);  
    }  
}
```

Работа программы показана на рисунке 3.



```
/usr/lib/jvm/java-11/bin/java -javaagent:/home/husya/.local/share/JetBrains/Tool  
Исходный текст: Это предложение для вставки  
Текст после вставки подстроки: Это предложение вставка для вставки  
  
Process finished with exit code 0
```

Рисунок 3 – Работа программы

**Задание 4:** В стихотворении найти количество слов, начинающихся и заканчивающихся гласной буквой.

Код класса Main:

```
public class VowelWordsCounter {
    public static void main(String[] args) {
        String poem = "Мороз и солнце, день чудесный, еще ты
дремлешь, друг прелесный";

        // Разделение стихотворения на отдельные слова
        String[] words = poem.split("\\s+"); // Разделение
по пробелам

        int vowelWordsCount = 0;

        // Перебор каждого слова из стихотворения
        for (String word : words) {
            char firstChar =
Character.toLowerCase(word.charAt(0));
            char lastChar =
Character.toLowerCase(word.charAt(word.length() - 1));

            // Проверка, начинается ли и заканчивается ли
слово гласной буквой
            // Гласные буквы: а, е, и, о, у, а, о, и, е,
ё, э, ы, у, ю, я
            if (isVowel(firstChar) && isVowel(lastChar)) {
                vowelWordsCount++;
            }
        }

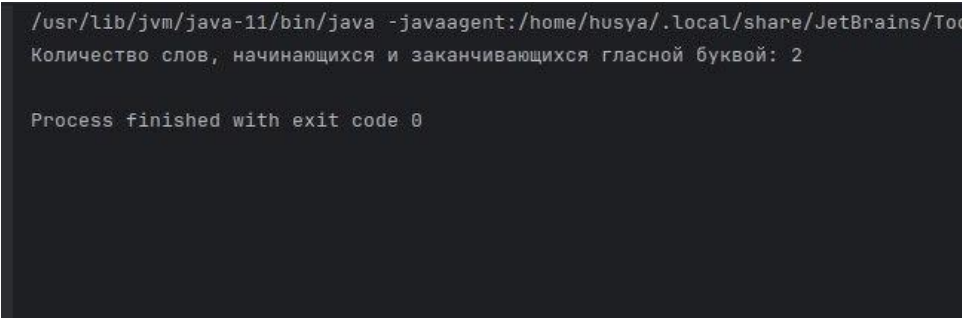
        System.out.println("Количество слов, начинающихся и
заканчивающихся гласной буквой: " + vowelWordsCount);
    }
}
```

```

        // Метод для проверки, является ли символ гласной буквой
        private static boolean isVowel(char c) {
            return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c ==
'u' || c == 'y' || c == 'а' || c == 'о' || c == 'и' || c == 'е' || c == 'ё'
|| c == 'э' || c == 'ы' || c == 'у' || c == 'ю' || c == 'я';
        }
    }
}

```

Работа программы показана на рисунке 4.



```

/usr/lib/jvm/java-11/bin/java -javaagent:/home/nusya/.local/share/JetBrains/Toolbox/scripts/java-11/bin/java -jar ...
Количество слов, начинающихся и заканчивающихся гласной буквой: 2

Process finished with exit code 0

```

Рисунок 4 – Работа программы

**Задание 5:** Во всех вопросительных предложениях текста найти и напечатать без повторений слова заданной длины.

Код класса Main:

```

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

public class QuestionWordsByLengthFinder {
    public static void main(String[] args) {
        String text = "Это вопросительное предложение? Можно ли найти слова заданной длины в нем? Во всех вопросительных предложениях, а не в других?";

        int len = 3; // Искомая длина слова
    }
}

```

```

Pattern pattern = Pattern.compile("(^[^!?]+[.!?])");
Matcher matcher = pattern.matcher(text);

Set<String> uniqueWords = new HashSet<>();

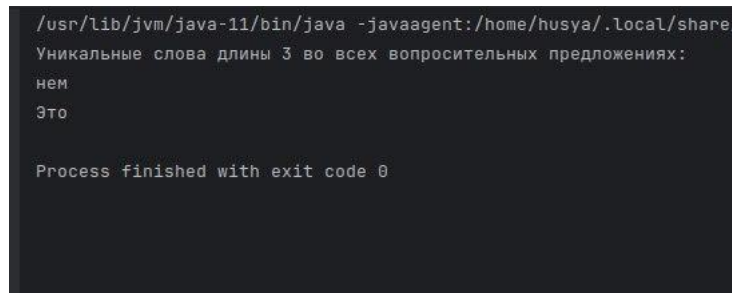
while (matcher.find()) {
    String part = text.substring(matcher.start(),
matcher.end());
    if (part.endsWith("?")) {
        for (String word: part.split(" ")) {
            String tempStr =
word.replaceAll("[.,?!:;]+$", "");
            if ((tempStr.length() == len) &&
!(uniqueWords.contains(tempStr))) {
                uniqueWords.add(tempStr);
            }
        }
    }
}

System.out.println("Уникальные слова длины " + len + "
во всех вопросительных предложениях:");
for (String uniqueWord : uniqueWords) {
    System.out.println(uniqueWord);
}
}
}

```



Работа программы показана на рисунке 5.



```
/usr/lib/jvm/java-11/bin/java -javaagent:/home/husya/.local/share...
Уникальные слова длины 3 во всех вопросительных предложениях:
нем
Это

Process finished with exit code 0
```

Рисунок 5 – Работа программы

**Задание 6:** В каждом предложении текста поменять местами первое слово с последним, не изменяя длины предложения.

Код класса Main:

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

public class SwapFirstAndLastWords {
    public static void main(String[] args) {
        String text = "Меня зовут Настя. Я учусь в МГТУ им
Баумана. Делаю лабораторные по Java.";

        Pattern pattern = Pattern.compile("([^.!?]+[.!?])");
        Matcher matcher = pattern.matcher(text);

        while (matcher.find()) {
            String part = text.substring(matcher.start(),
matcher.end()).trim();

            String[] words = part.split(" ");
            String first_word =
words[0].replaceAll("[.,?!:;]+$", "");
            String last_word = words[words.length -
1].replaceAll("[.,?!:;]+$", "");
```

```

        words[0] = words[0].replaceAll(first_word,
last_word);

        words[words.length - 1] = words[words.length -
1].replaceAll(last_word, first_word);

        System.out.println(String.join(" ", words));

    }

}

```

Работа программы показана на рисунке 6.

```
/usr/lib/jvm/java-11/bin/java -javaagent:/home/husya/.local/share/JetBrains/Toolbox
Настя зовут Меня.
Баумана учусь в МГТУ им Я.
Java лабораторные по Делаю.

Process finished with exit code 0
```

Рисунок 6 – Работа программы

**Задание 7:** В тексте исключить подстроку максимальной длины, начинающуюся и заканчивающуюся заданными символами.

### Код класса Main:

```
public class SubstringInsertion {
    public static void main(String[] args) {
        String text = "Это предложение для вставки";
        int k = 16; // Позиция, после которой нужно вставить
        подстроку

        String substringToInsert = "вставка "; // Заданная
        подстрока для вставки

        // Проверка, что k находится в пределах длины текста
        if (k < 0 || k > text.length()) {
            System.out.println("Ошибка:      позиция      вставки
            находится вне диапазона текста.");
        }
    }
}
```

```

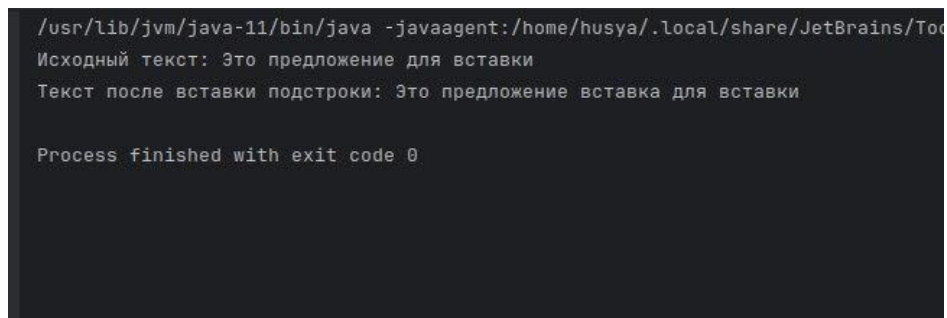
        return;
    }

    String newText = text.substring(0, k) +
substringToInsert + text.substring(k);

    System.out.println("Исходный текст: " + text);
    System.out.println("Текст после вставки подстроки: "
+ newText);
    }
}

```

Работа программы показана на рисунке 7.



```

/usr/lib/jvm/java-11/bin/java -javaagent:/home/husya/.local/share/JetBrains/Toolbox
Исходный текст: Это предложение для вставки
Текст после вставки подстроки: Это предложение вставка для вставки

Process finished with exit code 0

```

Рисунок 7 – Работа программы

**Задание 8:** Заменить все одинаковые рядом стоящие символы в тексте ОДНИМ СИМВОЛОМ.

Код класса Main:

```

public class ReplaceAdjacentChars {

    public static void main(String[] args) {
        String text = "Мееееня зовут нааастя и я деееелаю
лабораторные работы по джаве";

        String result = replaceAdjacentChars(text);

        System.out.println("Текст после замены: " + result);
    }
}

```

```

public static String replaceAdjacentChars(String text) {
    if (text == null || text.isEmpty()) {
        return text;
    }

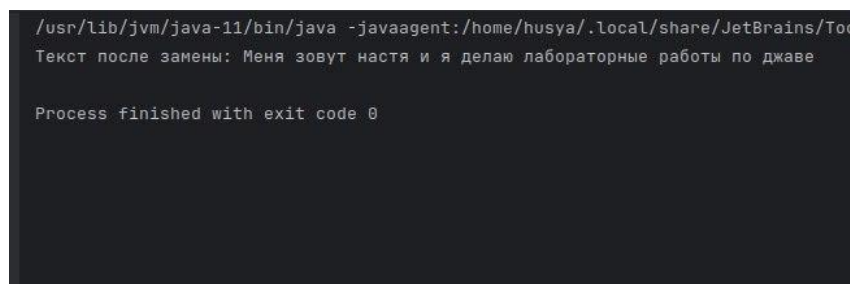
    StringBuilder result = new StringBuilder();
    result.append(text.charAt(0)); // Добавляем первый
    символ в результат

    for (int i = 1; i < text.length(); i++) {
        char currentChar = text.charAt(i);
        char previousChar = text.charAt(i - 1);

        if (currentChar != previousChar) {
            result.append(currentChar); // Добавляем символ
            к результату
        }
    }
}

```

Работа программы показана на рисунке 8.



```

/usr/lib/jvm/java-11/bin/java -javaagent:/home/husya/.local/share/JetBrains/Toolbox
Текст после замены: Меня зовут настя и я делаю лабораторные работы по джаве
Process finished with exit code 0

```

Рисунок 8 – Работа программы

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