



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

## ОТЧЕТ

по лабораторной работе № 7

Название: Строки и регулярные выражения

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

\_\_\_\_\_  
(Подпись, дата)

Э.А. Гаджиев

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022

---

Файл Main.java:

---

```
package Lab71;

import java.util.Scanner;

/*
 * В тексте слова заданной длины заменить указанной подстрокой, длина которой может не
 * совпадать с длиной слова.
 */

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Введите исходный текст: ");
        String text = scanner.nextLine();

        System.out.print("Введите длину заменяемых слов: ");
        int length = Integer.parseInt(scanner.nextLine());

        System.out.print("Введите подстроку, которой нужно заменить слова: ");
        String substr = scanner.nextLine();

        String replacedText = text.replaceAll("\\b[a-яА-Яа-zA-Z]{+" + length + "}\\b", substr);

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

---

Файл Main.java:

---

```
package Lab72;

import java.util.Scanner;

/*
 * В тексте после k-го символа вставить заданную подстроку.
 */

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Введите исходный текст: ");
        String text = scanner.nextLine();

        System.out.print("Введите подстроку, которую нужно вставить: ");
        String substr = scanner.nextLine();

        System.out.print("Введите номер k символа, после которого вставить подстроку: ");
        int kNum = Integer.parseInt(scanner.nextLine());

        String replacedText = text.replaceFirst("^{" + kNum + "}", "$0" + substr);

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

---

Файл Main.java:

---

```
package Lab73;

// Найти, каких букв, гласных или согласных, больше в каждом предложении текста.

import java.util.Arrays;
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.print("Введите текст: ");
        String text = scanner.nextLine();

        String[] splittedText = text.split("[.]\s*");
        System.out.println(Arrays.toString(splittedText));

        Pattern glPattern = Pattern.compile("[уеоэюияы]");
        Pattern soglPattern = Pattern.compile("[а-яА-Я&&[^уеоэюияыУЕОЭЮИЯЫ]]");

        for (String str : splittedText) {
            Matcher matcher = glPattern.matcher(str);

            int glCount = 0;
            int soglCount = 0;

            while (matcher.find())
                glCount++;

            matcher = soglPattern.matcher(str);
            while (matcher.find())
                soglCount++;

            System.out.println("В предложении \"" + str + "\"");
            System.out.println("    гласных: " + glCount + "\n    согласных: " + soglCount);
        }
    }
}
```

```
}
```

---

Файл Main.java:

---

```
package Lab74;

// В стихотворении найти количество слов, начинающихся и заканчивающихся гласной буквой.

import java.util.Arrays;
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 стихотворение:");
        StringBuilder text = new StringBuilder("");
        String line;
        while (!(line = scanner.nextLine()).equals("")) {
            text.append(line).append("\n");
        }

        String vowels = "уеыаоэяиюУЕЫАОЭЯИЮ";
        Pattern vowelsAtBeginAndEndOfWord = Pattern.compile("\\b[" + vowels + "][a-яА-Я]*[" + vowels + "]\\b");
        Matcher matcher = vowelsAtBeginAndEndOfWord.matcher(text.toString());
        int count = 0;
        while (matcher.find()) {
            System.out.print(matcher.group() + " ");
            count++;
        }
        System.out.print("\n" + count);
    }
}
```

---

Файл Main.java:

---

```
package Lab75;

// Во всех вопросительных предложениях текста найти и напечатать без повторений слова
// заданной длины.

import java.util.Arrays;
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;
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:");
        StringBuilder text = new StringBuilder("");
        String line;
        while (!(line = scanner.nextLine()).equals("")) {
            text.append(line).append("\n");
        }

        System.out.println("Длина слов, которые нужно вывести: ");
        int wordLength = Integer.parseInt(scanner.nextLine());

        // Pattern interrogativePattern = Pattern.compile("\\b[a-яА-Я0-9,:\\-()\\s&&[^?!]]*\\b");
        Pattern interrogativePattern = Pattern.compile("\\b[a-яА-Я,\\s-;0-9()]+[?]+");
        Matcher matcher = interrogativePattern.matcher(text.toString());
        while (matcher.find()) {
            System.out.println("В предложении \"" + matcher.group() + "\"");
            Set<String> words = new HashSet<>();
            String sentence = matcher.group();
            Pattern wordPattern = Pattern.compile("\\b[a-яА-Я]{ " + wordLength + " }\\b");
            Matcher wordMatcher = wordPattern.matcher(sentence);
            while (wordMatcher.find()) {
                words.add(wordMatcher.group());
            }
            for (String w : words)
                System.out.println("    " + w);
        }
    }
}
```

---

Файл Main.java:

---

```
package Lab76;

// В каждом предложении текста поменять местами первое слово с последним, не изменяя
// длины предложения.

import java.util.Arrays;
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;
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:");
        StringBuilder text = new StringBuilder("");
        String line;
        while (! (line = scanner.nextLine()).equals("")) {
            text.append(line).append("\n");
        }
        // regex for words matching - \b[a-zA-Z]+\b
        Pattern sentencePattern = Pattern.compile("\\b[a-zA-Z,\\s-;:0-9()]+[^?.!]+");
        Matcher matcher = sentencePattern.matcher(text.toString());
        while (matcher.find()) {
            System.out.println("Предложение \"\" + matcher.group() + "\"");
            System.out.println(matcher.group().replaceAll("^\\b[a-zA-Z]+\\b (.*) (\\b[a-zA-Z]+\\b$)", "$3$2$1"));
        }
    }
}

```

---

Файл Main.java:

---

```

package Lab77;

// В тексте исключить подстроку максимальной длины, начинающуюся и заканчивающуюся
// заданными символами.

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:");
        StringBuilder text = new StringBuilder("");
        String line;
        while (! (line = scanner.nextLine()).equals("")) {
            text.append(line).append("\n");
        }

        System.out.print("Подстрока начинается с символа: ");
        String firstChar = scanner.nextLine();

        System.out.print("Подстроки заканчивается символом: ");
        String lastChar = scanner.nextLine();

        Pattern substringPattern = Pattern.compile("[ " + firstChar + "].*[ " + lastChar + " ]");
        Matcher matcher = substringPattern.matcher(text.toString());

        int maxLength = 0;
        int maxStart = -1;
        int maxEnd = -1;
        while (matcher.find()) {
            System.out.println(matcher.group());
            int length = matcher.end() - matcher.start();
            if (length > maxLength) {
                maxLength = length;
                maxStart = matcher.start();
                maxEnd = matcher.end();
            }
        }

        text.delete(maxStart, maxEnd);
        System.out.println(text);
    }
}

```

---

Файл Main.java:

---

```

package Lab78;

// Заменить все одинаковые рядом стоящие символы в тексте одним символом.

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:");
        StringBuilder text = new StringBuilder("");
        String line;
        while (! (line = scanner.nextLine()).equals("")) {
            text.append(line).append("\n");
        }

        text = new StringBuilder(text.toString().replaceAll("([a-zA-Za-zA-Z])\\1+", "$1"));
        System.out.println(text);
    }
}

```