



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 6

Название: Строки, регулярные выражения

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

Г.Л. Кушнир

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2023

**Цель:** Изучить работу со строками и регулярными выражениями.

**Вариант 1 задание 6:** После каждого слова текста, заканчивающегося заданной подстрокой, вставить указанное слово.

**Вариант 1 задание 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.

**Вариант 2 задание 6:** Напечатать без повторения слова текста, у которых первая и последняя буквы совпадают.

**Вариант 2 задание 7:** В тексте найти и напечатать все слова максимальной и все слова минимальной длины.

**Вариант 3 задание 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.

**Вариант 3 задание 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.

**Вариант 4 задание 6:** Вывести в заданном тексте все слова, расположив их в алфавитном порядке.

**Вариант 4 задание 7:** Подсчитать, сколько слов в заданном тексте начинается с прописной буквы

Код основного класса Main:

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Scanner in = new Scanner(System.in);  
  
        // Вариант 1 задание 6
```

```

System.out.print("Input some text: ");
String text = in.nextLine();
System.out.print("Input substring to compare: ");
String substring = in.nextLine();
System.out.print("Input word to insert: ");
String insertWord = in.nextLine();

String[] words = text.split("\\s+");
StringBuilder changedText = new StringBuilder();
for (String word : words) {
    changedText.append(word).append(" ");
    if (word.replaceAll("[^a-zA-Za-яА-Я]", "").endsWith(substring)) {
        changedText.append(insertWord).append(" ");
    }
}
System.out.println("Sorted text by words: " + changedText);

System.out.println("-----");

// Вариант 1 задание 7

System.out.print("Input some text: ");
text = in.nextLine();
System.out.print("Input choice (0/1): ");
int choice = Integer.parseInt(in.nextLine());

if (choice == 0) {
    System.out.print("Input symbol: ");
    String symbol = in.nextLine();
    String newText = text.replaceAll(String.valueOf(symbol), "");
    System.out.println("Changed text: " + newText);
} else if (choice == 1) {
    System.out.print("Input position: ");
    int pos = Integer.parseInt(in.nextLine());
    System.out.print("Input symbol: ");
    String symbol = in.nextLine();
    StringBuilder newText = new StringBuilder();
    for (int i = 0; i < text.length(); i++) {
        newText.append(text.charAt(i));
        if ((i + 1) % pos == 0) {
            newText.append(symbol);
        }
    }
    System.out.println("Changed text: " + newText);
} else {
    System.out.println("Wrong choice. Text doesn't changed");
}

System.out.println("-----");

// Вариант 2 задание 6

System.out.print("Input some text: ");
text = in.nextLine();

Set<String> uniqueWords = Arrays.stream(text.split(" "))
    .map(String::toLowerCase)
    .filter(word -> word.length() > 1 && word.charAt(0) ==
word.charAt(word.length() - 1))
    .collect(Collectors.toSet());

System.out.println("Unique words with the same first and last letter:

```

```

" + uniqueWords);

    System.out.println("-----");
    -----");

    // Вариант 2 задание 7

    System.out.print("Input some text: ");
    text = in.nextLine();
    words = text.replaceAll("[^a-zA-Za-яA-Я]", "").split("\\s+");

    int minLength = Arrays.stream(words)
        .mapToInt(String::length)
        .min()
        .orElse(0);

    String[] minLengthWords = Arrays.stream(words)
        .filter(word -> word.length() == minLength)
        .toArray(String[]::new);

    int maxLength = Arrays.stream(words)
        .mapToInt(String::length)
        .max()
        .orElse(0);

    String[] maxLengthWords = Arrays.stream(words)
        .filter(word -> word.length() == maxLength)
        .toArray(String[]::new);

    System.out.println("Words with minimum length: " +
Arrays.toString(minLengthWords));
    System.out.println("Words with maximum length: " +
Arrays.toString(maxLengthWords));

    System.out.println("-----");
    -----");

    // Вариант 3 задание 6

    System.out.print("Input some text: ");
    text = in.nextLine();
    words = text.split("\\s+");
    String changedLine = String.join(" ", Arrays.copyOfRange(words, 1,
words.length)) + " " + words[0];
    System.out.println("Changed text: " + changedLine);

    System.out.println("-----");
    -----");

    // Вариант 3 задание 7

    System.out.print("Input some text: ");
    text = in.nextLine();
    StringBuilder encryptedText = new StringBuilder();
    for (int i = 0; i < text.length(); i += 3) {
        encryptedText.append(text.charAt(i));
    }
    for (int i = 1; i < text.length(); i += 3) {
        encryptedText.append(text.charAt(i));
    }
    for (int i = 2; i < text.length(); i += 3) {
        encryptedText.append(text.charAt(i));
    }
    System.out.println("Encrypted text: " + encryptedText);

```

```

        System.out.println("-----");
        // Вариант 4 задание 6

        System.out.print("Input some text: ");
        text = in.nextLine();

        words = text.replaceAll("[^a-zA-Za-яA-Я]",
        "").toLowerCase().split("\\s+");
        Arrays.sort(words);
        System.out.println("Sorted text by words: " + String.join(" ",
        words));

        System.out.println("-----");

        // Вариант 4 задание 7

        System.out.print("Input some text: ");
        text = in.nextLine();

        words = text.replaceAll("[^a-zA-Za-яA-Я]", "").split("\\s+");
        int count = 0;

        for (String word : words) {
            if (!word.isEmpty() && Character.isUpperCase(word.charAt(0))) {
                count++;
            }
        }

        System.out.println("Count of words that starts with uppercase letter:
        " + count);

    }
}

```

Работа программы представлена на рисунке 1.

```

145
35
16
2
0
0
-1
-21
-45
-----
Karlos
char
integer
lego
lol
memes
something
yandex
-----
Line equation: ( -4,000000 ) * x + ( 1,000000 ) * y + ( 6,000000 ) = 0; Number of crossed points: 2
Line equation: ( -2,000000 ) * x + ( 1,000000 ) * y + ( 0,000000 ) = 0; Number of crossed points: 3
Line equation: ( 1,000000 ) * x + ( 0,000000 ) * y + ( -1,000000 ) = 0; Number of crossed points: 2
Line equation: ( 0,000000 ) * x + ( 1,000000 ) * y + ( -2,000000 ) = 0; Number of crossed points: 2
Line equation: ( -1,000000 ) * x + ( 1,000000 ) * y + ( -2,000000 ) = 0; Number of crossed points: 2
Line equation: ( 1,000000 ) * x + ( 0,000000 ) * y + ( -2,000000 ) = 0; Number of crossed points: 2
Line equation: ( -1,500000 ) * x + ( 1,000000 ) * y + ( -1,500000 ) = 0; Number of crossed points: 2
Line equation: ( 1,000000 ) * x + ( 1,000000 ) * y + ( -4,000000 ) = 0; Number of crossed points: 2
-----
Intersection point: (2.0, 3.0)
Intersection point: (4.0, 5.0)
Intersection point: (4.0, 1.0)
Intersection point with minimum abscissa: (2.0, 3.0)

```

Рисунок 1 – Работа программы

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