



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)

А.М. Панфилкин

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Весь приведенный ниже код также доступен в следующем репозитории:

<https://github.com/SilkSlime/iu6plfbd>

**Задание 1:** В тексте после k-го символа вставить заданную подстроку.

Листинг 1 – Задание 1

```
package l7;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class e1 {

    /**
     * Вариант 1. Задача 5.
     * В тексте после k-го символа вставить заданную подстроку.
     */
    public static void main(String[] args) {
        String text = readFile("l7/sometext.txt");
        String subs = "--Hello, world!--";
        int k = 5;
        String result = text.substring(0, k) + subs + text.substring(k);
        System.out.println(result);
    }

    public static String readFile(String filename) {
        String text = "";
        try (BufferedReader br = new BufferedReader(new FileReader(filename))) {
            String line;
            while ((line = br.readLine()) != null) {
                text += line + '\n';
            }
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
        return text;
    }
}
```

**Задание 2:** После каждого слова текста, заканчивающегося заданной подстрокой, вставить указанное слово.

Листинг 2 – Задание 2

```
package l7;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
```

```

public class e2 {

    /**
     * Вариант 1. Задача 6.
     * После каждого слова текста, заканчивающегося заданной подстрокой, вставить указанное
     слово.
     */
    public static void main(String[] args) {
        String text = readFile("l7/sometext.txt");
        String subs = "lorem";
        String word = "IPSUM!";

        String result = text.replaceAll("(?i) (" + subs + ")\\b", "$1 " + word);

        System.out.println(result);

    }

    public static String readFile(String filename) {
        String text = "";
        try (BufferedReader br = new BufferedReader(new FileReader(filename))) {
            String line;
            while ((line = br.readLine()) != null) {
                text += line + '\n';
            }
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
        return text;
    }
}

```

**Задание 3:** В стихотворении найти количество слов, начинающихся и заканчивающихся гласной буквой.

Листинг 3 – Задание 3

```

package l7;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class e3 {

    /**
     * Вариант 2. Задача 5.
     * В стихотворении найти количество слов, начинающихся и заканчивающихся гласной буквой.
     */
    public static void main(String[] args) {

```

```

String text = readFile("l7/sometext.txt");
// Разделитель - любой символ, кроме буквы
String words[] = text.split("[^a-zA-Z]+");

int count = 0;
for (String word : words) {
    // Объяснение шаблона:
    // ^ - начало строки
    // [aeiouyAEIOUY] - любая гласная буква
    // .* - любое количество любых символов
    // [aeiouyAEIOUY]$ - любая гласная буква в конце строки
    if (word.matches("^[aeiouyAEIOUY].*[aeiouyAEIOUY]$")) {
        count++;
        // System.out.println(word);
    }
}

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

public static String readFile(String filename) {
    String text = "";
    try (BufferedReader br = new BufferedReader(new FileReader(filename))) {
        String line;
        while ((line = br.readLine()) != null) {
            text += line+'\n';
        }
    } catch (IOException e) {
        System.out.println("Error: " + e.getMessage());
    }
    return text;
}
}

```

**Задание 5:** В каждом предложении текста поменять местами первое слово с последним, не изменяя длины предложения.

#### Листинг 5 – Задание 5

```

package l7;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class e5 {

    /**
     * Вариант 3. Задача 5.
     * В каждом предложении текста поменять местами первое слово с последним, не изменяя длины

```

```

предложения.
    */
    public static void main(String[] args) {
        String text = readFile("17/sometext.txt");
        // Разделитель - знаки конца предложения
        String sentences[] = text.split("[.!?]+");

        String result = "";

        for (String sentence : sentences) {
            String[] words = sentence.trim().split("[^a-zA-Z]+");
            if (words.length == 0) {
                continue;
            } else if (words.length == 1 & words[0].equals("")) {
                continue;
            }

            String temp = words[0];
            words[0] = words[words.length-1];
            words[words.length-1] = temp;

            // rebuild sentence
            String newSentence = "";
            for (String word : words) {
                newSentence += word + " ";
            }
            newSentence = newSentence.trim();
            result += newSentence + ". ";
        }

        System.out.println(result);
    }

    public static String readFile(String filename) {
        String text = "";
        try (BufferedReader br = new BufferedReader(new FileReader(filename))) {
            String line;
            while ((line = br.readLine()) != null) {
                text += line+'\n';
            }
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
        return text;
    }
}

```

**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.

#### Листинг 6 – Задание 6

```
package l7;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class e6 {

    /**
     * Вариант 3. Задача 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 static void main(String[] args) {
        String text = readFile("l7/sometext.txt");
        // Разбить по пробельным символам
        String[] words = text.split("\\s+");
        // Циклический сдвиг массива на 1 элемент вправо
        String[] newWords = new String[words.length];
        for (int i = 0; i < words.length; i++) {
            newWords[(i+1)%words.length] = words[i];
        }
        // Собрать строку из массива
        String result = String.join(" ", newWords);
        System.out.println(result);
    }

    public static String readFile(String filename) {
        String text = "";
        try (BufferedReader br = new BufferedReader(new FileReader(filename))) {
            String line;
            while ((line = br.readLine()) != null) {
                text += line + '\n';
            }
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
        return text;
    }
}
```

**Задание 7:** Заменить все одинаковые рядом стоящие символы в тексте одним символом.

**Листинг 7 – Задание 7**

```
package l7;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class e7 {

    /**
     * Вариант 4. Задача 5.
     * Заменить все одинаковые рядом стоящие символы в тексте одним символом.
     */
    public static void main(String[] args) {
        String text = readFile("l7/sometext.txt");

        // Заменить все одинаковые рядом стоящие символы в тексте одним символом.
        String result = text.replaceAll("(.)\\1+", "$1");
        System.out.println(result);
    }

    public static String readFile(String filename) {
        String text = "";
        try (BufferedReader br = new BufferedReader(new FileReader(filename))) {
            String line;
            while ((line = br.readLine()) != null) {
                text += line + '\n';
            }
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
        return text;
    }
}
```

**Задание 8:** Вывести в заданном тексте все слова, расположив их в алфавитном порядке.

**Листинг 8 – Задание 8**

```
package l7;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Arrays;
```

```

public class e8 {

    /**
     * Вариант 4. Задача 5.
     * Вывести в заданном тексте все слова, расположив их в алфавитном порядке.
     */
    public static void main(String[] args) {
        String text = readFile("l7/sometext.txt");
        // Разбить по всем символам, кроме букв и цифр
        String[] words = text.split("[^a-zA-Z0-9]+");

        // Сделать уникальными
        // words = Arrays.stream(words).distinct().toArray(String[]::new);

        // Сортировка через библиотеку Arrays
        Arrays.sort(words);

        // Вывод
        for (String word : words) {
            System.out.println(word);
        }
    }

    public static String readFile(String filename) {
        String text = "";
        try (BufferedReader br = new BufferedReader(new FileReader(filename))) {
            String line;
            while ((line = br.readLine()) != null) {
                text += line + '\n';
            }
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
        return text;
    }
}

```



**Вывод:** В ходе лабораторной работы мы изучили работу со строками и регулярными выражениями в Java. Были выполнены задания по вставке подстрок в текст, замене символов и слов, поиску количества слов с гласными началом и концом, выводу уникальных слов и сортировке их в алфавитном порядке. В процессе выполнения заданий мы получили опыт работы с методами работы со строками и регулярными выражениями в Java, что поможет нам в дальнейшей разработке программ на этом языке.