



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

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Интеллектуальные системы анализа,
обработки и интерпретации больших данных

ОТЧЕТ
по лабораторной работе №7

Название: Строки и регулярные выражения

Дисциплина Языки программирования для работы с большими
данными

Студент

ИУ6–22М
(Группа)

(Подпись, дата)

М.Э.Хабаров

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

2024 г

Цель: освоить принципы строк и регулярных выражений на языке Java.

Задание №1

Формулировка задания и код программы представлены в листинге 1:

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

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

public class Main {

    public static void main(String[] args) {
        String inputFileName = "src/input.txt";
        String outputFileName = "src/output.txt";
        int k = 100; // позиция k, после которой вставляем подстроку
        String substringToInsert = "world"; // подстрока для вставки

        try (BufferedReader br = new BufferedReader(new
FileReader(inputFileName));
            BufferedWriter bw = new BufferedWriter(new
FileWriter(outputFileName))) {
            StringBuilder sb = new StringBuilder();
            String line;
            while ((line = br.readLine()) != null) {
                sb.append(line);
                sb.append("\n");
            }
            String modifiedText = insertSubstring(sb.toString(), k,
substringToInsert);
            bw.write(modifiedText);
            System.out.println("Измененный текст сохранен в файл " +
outputFileName);
        } catch (IOException e) {
            System.err.println("Ошибка при чтении/записи файла: " +
e.getMessage());
        }
    }

    private static String insertSubstring(String text, int k, String sub) {
        if (k < 0 || k > text.length()) {
            return text; // если k выходит за границы текста, вставляем в
конец текста
        }
        StringBuilder sb = new StringBuilder();
        sb.append(text.substring(0, k));
        sb.append(sub);
        sb.append(text.substring(k));
        return sb.toString();
    }
}
```

Задание №2

Формулировка задания и код программы представлены в листинге 2:

```

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.HashSet;

/*
    Из текста удалить все слова заданной длины, начинающиеся на согласную
    букву.
    */

public class Main {

    public static void main(String[] args) {
        String inputFile = "src/input.txt"; // Имя исходного файла
        String outputFile = "src/output.txt"; // Имя файла для записи
результата
        int wordLength = 5; // Длина слова, которое нужно удалить
        HashSet<Character> consonants = new HashSet<>(); // Множество
согласных русских букв
        consonants.add('б');
        consonants.add('в');
        consonants.add('г');
        consonants.add('д');
        consonants.add('ж');
        consonants.add('з');
        consonants.add('й');
        consonants.add('к');
        consonants.add('л');
        consonants.add('м');
        consonants.add('н');
        consonants.add('п');
        consonants.add('р');
        consonants.add('с');
        consonants.add('т');
        consonants.add('ф');
        consonants.add('х');
        consonants.add('ц');
        consonants.add('ч');
        consonants.add('ш');
        consonants.add('щ');

        try (BufferedReader reader = new BufferedReader(new
FileReader(inputFile));
            BufferedWriter writer = new BufferedWriter(new
FileWriter(outputFile))) {

            String line;
            while ((line = reader.readLine()) != null) {
                String[] words = line.split("\\s+");
                StringBuilder modifiedLine = new StringBuilder();

                for (String word : words) {
                    if (word.length() != wordLength ||
!consonants.contains(Character.toLowerCase(word.charAt(0)))) {
                        modifiedLine.append(word).append(" ");
                    }
                }

                writer.write(modifiedLine.toString().trim());
                writer.newLine();
            }
        }
    }
}

```

```

        System.out.println("Слова заданной длины, начинающиеся на
согласную букву, успешно удалены.");
        System.out.println("Результат сохранен в файл " + outputFile);

    } catch (IOException e) {
        System.err.println("Ошибка при чтении/записи файла: " +
e.getMessage());
    }
}
}

```

Задание №3

Формулировка задания и код программы представлены в листинге 3:

```

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

/*
    В тексте найти и напечатать все слова максимальной и все слова
минимальной длины.
*/

public class Main {

    public static void main(String[] args) {
        String inputFile = "src/input.txt"; // Имя исходного файла
        String outputFile = "src/output.txt"; // Имя файла для записи
результата

        List<String> words = new ArrayList<>();
        int minLength = Integer.MAX_VALUE;
        int maxLength = Integer.MIN_VALUE;

        try (BufferedReader reader = new BufferedReader(new
FileReader(inputFile))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] lineWords = line.split("\\s+");
                for (String word : lineWords) {
                    words.add(word);
                    if (word.length() < minLength) {
                        minLength = word.length();
                    }
                    if (word.length() > maxLength) {
                        maxLength = word.length();
                    }
                }
            }

            List<String> minWords = new ArrayList<>();
            List<String> maxWords = new ArrayList<>();

            for (String word : words) {
                if (word.length() == minLength) {
                    minWords.add(word);
                }
            }
        }
    }
}

```

```

    }
    if (word.length() == maxLength) {
        maxWords.add(word);
    }
}

try (BufferedWriter writer = new BufferedWriter(new
FileWriter(outputFile))) {
    writer.write("Слова минимальной длины (" + minLength + "):
");
    for (String minWord : minWords) {
        writer.write(minWord + " ");
    }
    writer.newLine();

    writer.write("Слова максимальной длины (" + maxLength + "):
");
    for (String maxWord : maxWords) {
        writer.write(maxWord + " ");
    }
}

System.out.println("Слова минимальной и максимальной длины
успешно найдены.");
System.out.println("Результат сохранен в файл " + outputFile);

} catch (IOException e) {
    System.err.println("Ошибка при чтении/записи файла: " +
e.getMessage());
}
}
}

```

Задание №4

Формулировка задания и код программы представлены в листинге 4:

```

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

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

public class Main {

    public static void main(String[] args) {
        String inputFile = "src/input.txt"; // Имя исходного файла
        int StartEndVowelCount = 0;

        try (BufferedReader reader = new BufferedReader(new
FileReader(inputFile))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] words = line.split("\\s+");
                for (String word : words) {
                    if (!word.isEmpty()) {
                        char firstChar =
Character.toLowerCase(word.charAt(0));
                        char lastChar =

```

```

Character.toLowerCase(word.charAt(word.length() - 1));
        if (isRussianVowel(firstChar) &&
isRussianVowel(lastChar)) {
            StartEndVowelCount++;
        }
    }
}

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

} catch (IOException e) {
    System.err.println("Ошибка при чтении файла: " + e.getMessage());
}

}

private static boolean isRussianVowel(char ch) {
    return "aeёиоуыэюя".indexOf(ch) != -1;
}
}

```

Задание №5

Формулировка задания и код программы представлены в листинге 5:

```

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

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

public class Main {

    public static void main(String[] args) {
        String inputFile = "src/input.txt"; // Имя исходного файла
        String outputFile = "src/output.txt"; // Имя файла для записи
результата

        try (BufferedReader br = new BufferedReader(new
FileReader(inputFile));
            BufferedWriter bw = new BufferedWriter(new
FileWriter(outputFile))) {
            StringBuilder sb = new StringBuilder();
            String line;
            while ((line = br.readLine()) != null) {
                String[] words = line.split("\\s+");
                if (words.length > 1) {
                    String firstWord = words[0];
                    words[0] = words[words.length - 1];
                    words[words.length - 1] = firstWord;
                }
                for (String word : words) {
                    sb.append(word).append(" ");
                }
                sb.append("\n");
            }
        }
    }
}

```

```

        bw.write(sb.toString().trim()); // Запись модифицированного предложения
        System.out.println("Результат успешно записан в файл " +
outputFile);
    } catch (IOException e) {
        System.err.println("Ошибка при чтении/записи файла: " +
e.getMessage());
    }
}
}

```

Задание №6

Формулировка задания и код программы представлены в листинге 6:

```

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.HashSet;
import java.util.Set;

/*
    Найти такое слово в первом предложении, которого нет ни в одном из
    остальных предложений.
*/

public class Main {

    public static void main(String[] args) {
        String inputFile = "src/input.txt"; // Имя исходного файла

        try (BufferedReader reader = new BufferedReader(new
FileReader(inputFile))) {
            String firstSentence = reader.readLine();
            Set<String> firstSentenceWords = extractWords(firstSentence);

            String uniqueWord = findUniqueWord(reader, firstSentenceWords);
            System.out.println("Уникальное слово в первом предложении: " +
uniqueWord);

        } catch (IOException e) {
            System.err.println("Ошибка при чтении файла: " + e.getMessage());
        }
    }

    private static Set<String> extractWords(String sentence) {
        Set<String> words = new HashSet<>();
        String[] tokens = sentence.split("[\\s.,!?:]+");
        for (String token : tokens) {
            words.add(token.toLowerCase());
        }
        return words;
    }

    private static String findUniqueWord(BufferedReader reader, Set<String>
firstSentenceWords) throws IOException {
        String line;
        while ((line = reader.readLine()) != null) {
            Set<String> currentWords = extractWords(line);
            for (String word : firstSentenceWords) {
                if (!currentWords.contains(word)) {
                    return word;
                }
            }
        }
    }
}

```

```

    }
}
return "Такого слова не найдено";
}
}

```

Задание №7

Формулировка задания и код программы представлены в листинге 7:

```

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

/*
    Подсчитать, сколько раз заданное слово входит в текст.
*/

public class Main {

    public static void main(String[] args) {
        String inputFile = "src/input.txt"; // Имя исходного файла
        String wordToCount = "явилась"; // Слово, которое нужно подсчитать

        try (BufferedReader reader = new BufferedReader(new
FileReader(inputFile))) {
            String line;
            int count = 0;
            while ((line = reader.readLine()) != null) {
                count += countWordOccurrences(line, wordToCount);
            }

            System.out.println("Слово '" + wordToCount + "' встречается " +
count + " раз(а).");

        } catch (IOException e) {
            System.err.println("Ошибка при чтении файла: " + e.getMessage());
        }
    }

    private static int countWordOccurrences(String line, String word) {
        int count = 0;
        String[] words = line.split("[\\s.,!?]+");
        for (String w : words) {
            if (w.equalsIgnoreCase(word)) {
                count++;
            }
        }
        return count;
    }
}

```

Задание №8

Формулировка задания и код программы представлены в листинге 8:

```

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

```



```

import java.util.Set;
import java.util.TreeSet;

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

public class Main {

    public static void main(String[] args) {
        String inputFile = "src/input.txt"; // Имя исходного файла

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

        try (BufferedReader reader = new BufferedReader(new
FileReader(inputFile))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] words = line.split("[\\s.,!?:]+");
                for (String word : words) {
                    if (!word.isEmpty()) {
                        uniqueWords.add(word.toLowerCase()); // Добавляем
слово в нижнем регистре
                    }
                }
            }

            for (String word : uniqueWords) {
                System.out.println(word);
            }

        } catch (IOException e) {
            System.err.println("Ошибка при чтении файла: " + e.getMessage());
        }
    }
}

```

Вывод: в результате выполнения лабораторной работы были освоены основные принципы строк и регулярных выражений на языке Java.