



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 09.04.01/07 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)

Т.И. Кадыров

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2024

**Цель работы:** освоить базовые принципы работы со строками и регулярными выражениями на языке Java.

**Вариант: 8.**

**Задание 1:** Из небольшого текста удалить все символы, кроме пробелов, не являющиеся буквами. Между последовательностями подряд идущих букв оставить хотя бы один пробел.

Код решения приведен в листинге 1.

Листинг 1 — реализация решения

```
public class Main {
    public static void main(String[] args) {
        String text = "Из небольшого текста удалить все символы, кроме пробелов, не являющиеся буквами. Между последовательностями подряд идущих букв оставить хотя бы один пробел.";
        System.out.println(text);
        String result = processText(text);
        System.out.println(result);
    }

    public static String processText(String text) {
        StringBuilder result = new StringBuilder();
        boolean lastWasLetter = false;

        for (char c : text.toCharArray()) {
            if (Character.isLetter(c)) {
                result.append(c);
                lastWasLetter = true;
            } else if (c == ' ') {
                if (lastWasLetter) {
                    result.append(' ');
                    lastWasLetter = false;
                }
            }
        }

        return result.toString();
    }
}
```

**Задание 2:** Из текста удалить все слова заданной длины, начинающиеся на согласную букву.

Код решения приведен в листинге 2.

Листинг 2 — реализация решения

```
import java.util.Scanner;

public class Main {
```

```

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    String text = "Из текста удалить все слова заданной длины, начинающиеся на согласную букву.";
    System.out.println(text);
    int wordLength = scanner.nextInt();
    String result = processText(text, wordLength);
    System.out.println(result);
}

public static String processText(String text, int wordLength) {
    StringBuilder result = new StringBuilder();

    String[] words = text.split("\\s+");
    for (String word : words) {
        String cleanWord = cleanWord(word);
        if (cleanWord.length() != wordLength || !startsWithConsonant(cleanWord)) {
            result.append(word).append(" ");
        }
    }

    return result.toString().trim();
}

public static boolean startsWithConsonant(String word) {
    char firstChar = Character.toLowerCase(word.charAt(0));
    return !(firstChar == 'a' || firstChar == 'e' || firstChar == 'i' || firstChar == 'o' || firstChar == 'u');
}

public static String cleanWord(String word) {

    while (!word.isEmpty() && !Character.isLetterOrDigit(word.charAt(word.length() - 1))) {
        word = word.substring(0, word.length() - 1);
    }
    return word;
}
}

```

**Задание 3:** Напечатать квитанцию об оплате телеграммы, если стоимость одного слова задана.

Код решения приведен в листинге 3.

Листинг 3 — реализация решения

```

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String telegramText = "Напечатать квитанцию об оплате телеграммы, если стоимость одного слова задана.";
        double wordCost = scanner.nextDouble();

        double totalCost = calculateTotalCost(telegramText, wordCost);
    }
}

```

```

String receipt = generateReceipt(telegramText, wordCost, totalCost);

System.out.println(receipt);
}

public static double calculateTotalCost(String text, double wordCost) {
    String[] words = text.split("\\s+");
    int wordCount = words.length;
    return wordCount * wordCost;
}

public static String generateReceipt(String text, double wordCost, double totalCost) {
    StringBuilder receipt = new StringBuilder();
    receipt.append("Текст телеграммы: ").append(text).append("\n");
    receipt.append("Стоимость одного слова: ").append(wordCost).append(" руб.\n");
    receipt.append("Количество слов: ").append(text.split("\\s+").length).append("\n");
    receipt.append("Общая стоимость: ").append(totalCost).append(" руб.");
    return receipt.toString();
}
}

```

**Задание 4:** В стихотворении найти одинаковые буквы, которые встречаются во всех словах.

Код решения приведен в листинге 4.

Листинг 4 — реализация решения

```

import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

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

        String poem = "слово корабль олово";

        Set<Character> commonLetters = findCommonLetters(poem);

        if (commonLetters.isEmpty()) {
            System.out.println("Во всех словах стихотворения нет общих букв.");
        } else {
            for (char letter : commonLetters) {
                System.out.print(letter + " ");
            }
        }
    }

    public static Set<Character> findCommonLetters(String poem) {
        String[] words = poem.split("\\s+");
        Set<Character> commonLetters = new HashSet<>();
    }
}

```

```

// Инициализируем множество букв первым словом
for (char letter : words[0].toCharArray()) {
    commonLetters.add(letter);
}

// Пересекаем множество букв каждого последующего слова с общими буквами
for (int i = 1; i < words.length; i++) {
    Set<Character> currentWordLetters = new HashSet<>();
    for (char letter : words[i].toCharArray()) {
        currentWordLetters.add(letter);
    }
    commonLetters.retainAll(currentWordLetters);
}

return commonLetters;
}
}

```

**Задание 5:** На основании правила кодирования, описанного в предыдущем примере, расшифровать заданный набор символов.

Код решения приведен в листинге 5.

Листинг 5 — реализация решения

```

public class Main {
    public static void main(String[] args) {
        String originalText = "Hello, world!";
        System.out.println(" Исходный текст: " + originalText);

        String encryptedText = encryptText(originalText);
        System.out.println(" Зашифрованный текст: " + encryptedText);

        String decryptedText = decryptText(encryptedText);
        System.out.println(" Расшифрованный текст: " + decryptedText);
    }

    public static String encryptText(String plaintext) {
        StringBuilder group1 = new StringBuilder();
        StringBuilder group2 = new StringBuilder();
        StringBuilder group3 = new StringBuilder();

        for (int i = 0; i < plaintext.length(); i++) {
            char ch = plaintext.charAt(i);
            if (i % 3 == 0) {
                group1.append(ch);
            } else if (i % 3 == 1) {
                group2.append(ch);
            } else {
                group3.append(ch);
            }
        }
    }
}

```

```

        StringBuilder encryptedText = new StringBuilder();
        encryptedText.append(group1.toString());
        encryptedText.append(group2.toString());
        encryptedText.append(group3.toString());

        return encryptedText.toString();
    }

    public static String decryptText(String encryptedText) {
        int len = encryptedText.length();
        int group1len = len / 3 + (len % 3 >= 1 ? 1 : 0);
        int group2len = len / 3 + (len % 3 > 1 ? 1 : 0);
        int group3len = len / 3;

        char[] group1 = new char[group1len];
        char[] group2 = new char[group2len];
        char[] group3 = new char[group3len];

        for (int i = 0; i < len; i++) {
            if (i < group1len) {
                group1[i] = encryptedText.charAt(i);
            } else if (i < group1len + group2len) {
                group2[i - group1len] = encryptedText.charAt(i);
            } else {
                group3[i - group1len - group2len] = encryptedText.charAt(i);
            }
        }

        StringBuilder res = new StringBuilder();
        for (int i = 0; i < len / 3; i++) {
            res.append(group1[i]);
            res.append(group2[i]);
            res.append(group3[i]);
        }

        if (len % 3 == 2) {
            res.append(group1[group1len - 1]);
            res.append(group2[group2len - 1]);
        } else if (len % 3 == 1) {
            res.append(group1[group1len - 1]);
        }

        return res.toString();
    }
}

```

**Задание 6:** Напечатать слова русского текста в алфавитном порядке по первой букве. Слова, начинающиеся с новой буквы, печатать с красной строки.

Код решения приведен в листинге 6.

## Листинг 6 — реализация решения

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        String russianText = "Напечатать слова русского текста в алфавитном порядке по первой букве. Слова, начинающиеся с новой буквы, печатать с красной строки.";

        Map<Character, List<String>> wordsByFirstLetter = new TreeMap<>();
        String[] words = russianText.split("\\s+");

        for (String word : words) {
            char firstLetter = word.charAt(0);
            List<String> wordList = wordsByFirstLetter.getOrDefault(firstLetter, new ArrayList<>());
            wordList.add(word);
            wordsByFirstLetter.put(firstLetter, wordList);
        }

        for (char letter : wordsByFirstLetter.keySet()) {
            List<String> wordList = wordsByFirstLetter.get(letter);
            boolean firstWordInLine = true;
            for (String word : wordList) {
                if (firstWordInLine) {
                    System.out.println();
                    firstWordInLine = false;
                }
                System.out.print(word + " ");
            }
        }
    }
}
```

**Задание 7:** Подсчитать, сколько раз заданное слово входит в текст.

Код решения приведен в листинге 7.

## Листинг 7 — реализация решения

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String text = "Подсчитать, сколько раз заданное слово входит в текст";
        String wordToSearch = scanner.next();
        int count = countOccurrences(text, wordToSearch);
        System.out.println("Слово '" + wordToSearch + "' встречается " + count + " раз.");
    }

    public static int countOccurrences(String text, String word) {
        int count = 0;
        int index = 0;
        while ((index = text.indexOf(word, index)) != -1) {

```

```
        count++;
        index += word.length();
    }
    return count;
}
}
```

**Задание 8:** Преобразовать каждое слово в тексте, удалив из него все последующие (предыдущие) вхождения первой (последней) буквы этого слова

Код решения приведен в листинге 8.

Листинг 8 — реализация решения

```
import java.util.Scanner;

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

        String transformedText = transformWords(text);
        System.out.println(transformedText);
    }

    public static String transformWords(String text) {
        StringBuilder result = new StringBuilder();

        String[] words = text.split("\\s+");
        for (String word : words) {
            char firstChar = word.charAt(0);

            boolean isFirstCharMet = false;

            for (char c : word.toCharArray()) {
                if (c == firstChar && !isFirstCharMet) {
                    isFirstCharMet = true;
                    result.append(c);
                } else if (c != firstChar) {
                    result.append(c);
                }
            }

            result.append(" ");
        }

        return result.toString().trim();
    }
}
```

**Вывод:** в ходе лабораторной работы были освоены базовые принципы работы со строками и регулярными выражениями на языке Java.