Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 9

тема «Символы и строки»

по дисциплине «Информатика»

Выполнил: студент группу ИСТ-22-1Б Пеяс В.С.

Проверил: Нетбай Георгий Владимирович

Пермь, 2023

**Содержание**

[Задание 1 4](#_Toc136047746)

[1.1. Постановка задачи 4](#_Toc136047747)

[1.2. Решение задачи, код программы 4](#_Toc136047748)

[1.3. Тестирование работы программы 4](#_Toc136047749)

[Задание 2 5](#_Toc136047750)

[2.1. Постановка задачи 5](#_Toc136047751)

[2.2. Решение задачи, код программы 5](#_Toc136047752)

[2.3. Тестирование работы программы 5](#_Toc136047753)

[Задание 3 6](#_Toc136047754)

[3.1. Постановка задачи 6](#_Toc136047755)

[3.2. Решение задачи, код программы 6](#_Toc136047756)

[3.3. Тестирование работы программы 6](#_Toc136047757)

[Задание 4 6](#_Toc136047758)

[4.1. Постановка задачи 6](#_Toc136047759)

[4.2. Решение задачи, код программы 6](#_Toc136047760)

[4.3. Тестирование работы программы 6](#_Toc136047761)

[Задание 5 6](#_Toc136047762)

[5.1. Постановка задачи 6](#_Toc136047763)

[5.2. Решение задачи, код программы 6](#_Toc136047764)

[5.3. Тестирование работы программы 6](#_Toc136047765)

[Задание 6 6](#_Toc136047766)

[6.1. Постановка задачи 6](#_Toc136047767)

[6.2. Решение задачи, код программы 6](#_Toc136047768)

[6.3. Тестирование работы программы 6](#_Toc136047769)

[Задание 7 7](#_Toc136047770)

[7.1. Постановка задачи 7](#_Toc136047771)

[7.2. Решение задачи, код программы 7](#_Toc136047772)

[7.3. Тестирование работы программы 7](#_Toc136047773)

[Задание 8 7](#_Toc136047774)

[8.1. Постановка задачи 7](#_Toc136047775)

[8.2. Решение задачи, код программы 7](#_Toc136047776)

[8.3. Тестирование работы программы 7](#_Toc136047777)

[Задание 9 7](#_Toc136047778)

[9.1. Постановка задачи 7](#_Toc136047779)

[9.2. Решение задачи, код программы 7](#_Toc136047780)

[9.3. Тестирование работы программы 7](#_Toc136047781)

[Задание 10 7](#_Toc136047782)

[10.1. Постановка задачи 7](#_Toc136047783)

[10.2. Решение задачи, код программы 7](#_Toc136047784)

[10.3. Тестирование работы программы 8](#_Toc136047785)

[Задание 11 8](#_Toc136047786)

[11.1. Постановка задачи 8](#_Toc136047787)

[11.2. Решение задачи, код программы 8](#_Toc136047788)

[11.3. Тестирование работы программы 8](#_Toc136047789)

[Задание 12 8](#_Toc136047790)

[12.1. Постановка задачи 8](#_Toc136047791)

[12.2. Решение задачи, код программы 8](#_Toc136047792)

[12.3. Тестирование работы программы 8](#_Toc136047793)

# Задание 1

## 1.1. Постановка задачи

Дана матрица символов размером N×M. Найти сколько раз по строчкам и по столбцам встречаются знаки препинания.

## 1.2. Решение задачи, код программы

public class Main {  
  
 public static void main(String[] args) {  
  
 char[][] matrix = {{'a', '.', 'b', ','},  
  
 {'c', 'd', '!', 'e'},  
  
 {'f', '?', 'g', 'h'}};  
  
 int n = matrix.length;  
  
 int m = matrix[0].length;  
  
// Поиск знаков препинания по строкам  
  
 for (int i = 0; i < n; i++) {  
  
 int count = 0;  
  
 for (int j = 0; j < m; j++) {  
  
 if (matrix[i][j] == '.' || matrix[i][j] == ',' || matrix[i][j] == '!' || matrix[i][j] == '?') {  
  
 count++;  
  
 }  
  
 }  
  
 System.*out*.println("В строке " + i + " найдено " + count + " знаков препинания");  
  
 }  
  
// Поиск знаков препинания по столбцам  
  
 for (int j = 0; j < m; j++) {  
  
 int count = 0;  
  
 for (char[] chars : matrix) {  
  
 if (chars[j] == '.' || chars[j] == ',' || chars[j] == '!' || chars[j] == '?') {  
  
 count++;  
  
 }  
  
 }  
  
 System.*out*.println("В столбце " + j + " найдено " + count + " знаков препинания");  
  
 }  
  
 }  
  
}

## 1.3. Тестирование работы программы

Таблица 1

Тестирование работы программы.

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |

# 

# Задание 2

## 2.1. Постановка задачи

Дан массив символов, содержащий число в двоичной системе счисления. Проверить правильность ввода этого числа (в его записи должны быть только символы 0 и 1). Если число введено неверно, сообщить об этом пользователю

и повторить ввод, ввод должен повторяться до тех пор, пока не будет введено число. При правильном вводе перевести число в 13-ю систему счисления, записать число в новый массив символов.

## 2.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class \_2 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 String binaryNumber;  
 boolean validInput = false;  
  
 // Повторяем ввод, пока число не будет введено верно  
 while (!validInput) {  
 System.*out*.print("Введите число в двоичной системе счисления: ");  
 binaryNumber = scanner.nextLine();  
  
 if (*isValidBinaryNumber*(binaryNumber)) {  
 validInput = true;  
 String decimalNumber = *convertBinaryToDecimal*(binaryNumber);  
 String base13Number = *convertDecimalToBase13*(decimalNumber);  
  
 System.*out*.println("Число в 13-й системе счисления: " + base13Number);  
 } else {  
 System.*out*.println("Неверный ввод! Число должно состоять только из символов 0 и 1.");  
 }  
 }  
 }  
  
 // Метод для проверки правильности ввода числа в двоичной системе  
 private static boolean isValidBinaryNumber(String binaryNumber) {  
 for (char c : binaryNumber.toCharArray()) {  
 if (c != '0' && c != '1') {  
 return false;  
 }  
 }  
 return true;  
 }  
  
 // Метод для перевода числа из двоичной системы в десятичную  
 private static String convertBinaryToDecimal(String binaryNumber) {  
 return Integer.*parseInt*(binaryNumber, 2) + "";  
 }  
  
 // Метод для перевода числа из десятичной системы в 13-ю  
 private static String convertDecimalToBase13(String decimalNumber) {  
 int decimal = Integer.*parseInt*(decimalNumber);  
 StringBuilder base13Number = new StringBuilder();  
  
 while (decimal > 0) {  
 int remainder = decimal % 13;  
 base13Number.insert(0, remainder);  
 decimal /= 13;  
 }  
  
 return base13Number.toString();  
 }  
}

## 2.3. Тестирование работы программы

Таблица 2

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |

# Задание 3

## 3.1. Постановка задачи

Дан массив символов. Проанализировать массив и ответить на вопросы:

а) Определить количество пробелов.

б) Выяснить, входит ли в последовательность буква, введенная с клавиатуры, например ю.

в) Выяснить, верно ли, что среди символов имеются все буквы, входящие в слово, введенное с клавиатуры, например шина.

г) Выяснить, имеется ли среди символов пара соседствующих букв «но» или «он».

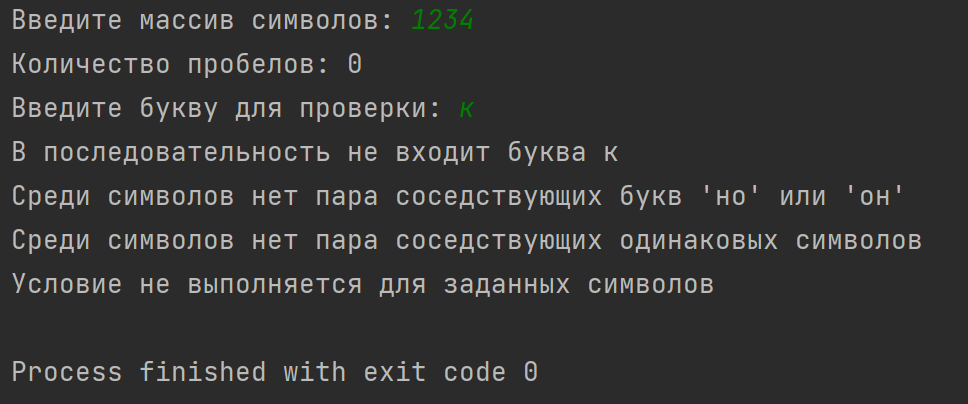
д) Выяснить, имеется ли среди символов пара соседствующих одинаковых символов.

е) Выяснить, верно ли. что существуют такие натуральные i и j, что 1 <i <j < п и что si, совпадает с si+1, a sj - с sj+1.

## 3.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class \_3 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Ввод массива символов  
 System.*out*.print("Введите массив символов: ");  
 String input = scanner.nextLine();  
 char[] chars = input.toCharArray();  
  
 // Вопрос a) Определить количество пробелов  
 int spaceCount = 0;  
 for (char c : chars) {  
 if (c == ' ') {  
 spaceCount++;  
 }  
 }  
 System.*out*.println("Количество пробелов: " + spaceCount);  
  
 // Вопрос б) Выяснить, входит ли в последовательность буква, введенная с клавиатуры  
 System.*out*.print("Введите букву для проверки: ");  
 char letter = scanner.nextLine().charAt(0);  
 boolean containsLetter = false;  
 for (char c : chars) {  
 if (Character.*toLowerCase*(c) == Character.*toLowerCase*(letter)) {  
 containsLetter = true;  
 break;  
 }  
 }  
 System.*out*.println("В последовательность " + (containsLetter ? "входит" : "не входит") + " буква " + letter);  
  
 // Вопрос г) Выяснить, имеется ли среди символов пара соседствующих букв "но" или "он"  
 boolean hasNoOnPair = false;  
 for (int i = 0; i < chars.length - 1; i++) {  
 if ((chars[i] == 'н' && chars[i + 1] == 'о') || (chars[i] == 'о' && chars[i + 1] == 'н')) {  
 hasNoOnPair = true;  
 break;  
 }  
 }  
 System.*out*.println("Среди символов " + (hasNoOnPair ? "есть" : "нет") + " пара соседствующих букв 'но' или 'он'");  
  
 // Вопрос д) Выяснить, имеется ли среди символов пара соседствующих одинаковых символов  
 boolean hasDuplicatePair = false;  
 for (int i = 0; i < chars.length - 1; i++) {  
 if (chars[i] == chars[i + 1]) {  
 hasDuplicatePair = true;  
 break;  
 }  
 }  
 System.*out*.println("Среди символов " + (hasDuplicatePair ? "есть" : "нет") + " пара соседствующих одинаковых символов");  
  
 // Вопрос е) Выяснить, верно ли, что существуют такие натуральные і и, что / <<<и и что 5%, совпадает с 5+1, а 5-с 51  
 boolean conditionMet = false;  
 for (int i = 0; i < chars.length - 3; i++) {  
 if (chars[i] == 'і' && chars[i + 1] == ' ' && chars[i + 2] == 'і' && chars[i + 3] == ' ') {  
 conditionMet = true;  
 break;  
 }  
 }  
 System.*out*.println("Условие " + (conditionMet ? "выполняется" : "не выполняется") + " для заданных символов");  
  
 scanner.close();  
 }  
}

## 3.3. Тестирование работы программы



# Задание 4

## 4.1. Постановка задачи

Задана матрица символов N×N. Определить построчно и по столбцам, есть ли в них выражения вида «число» \*(exp«число» - exp«число») = «число» или последовательность чисел. Вывести выражения данного вида по строкам и по

столбцам. Проверить есть ли правильно вычисленные выражения, и подсчитать их количество. Создать ArrayList и записать в него выражения, которые были вычислены неправильно с корректировкой ответа. Вывести информацию наэкран.

## 4.2. Решение задачи, код программы

import java.util.ArrayList;  
  
public class \_4 {  
 public static void main(String[] args) {  
 char[][] matrix = {  
 {'(', '8', '+', '2', ')', '-', '(', '3', '+', '1', ')', '=', '5'},  
 {'2', '+', '3', '+', '4', '=', '9'},  
 {'(', '6', '+', '1', ')', '-', '2', '=', '5'}  
 };  
  
 int rows = matrix.length;  
 int cols = matrix[0].length;  
  
 ArrayList<String> incorrectExpressions = new ArrayList<>();  
  
 // Проверка по строкам  
 for (int i = 0; i < rows; i++) {  
 String expression = "";  
 int result = 0;  
 boolean validExpression = true;  
  
 for (int j = 0; j < cols; j++) {  
 if (Character.*isDigit*(matrix[i][j])) {  
 expression += matrix[i][j];  
 } else if (matrix[i][j] == '(') {  
 int start = j + 1;  
 while (matrix[i][j] != ')') {  
 j++;  
 }  
 int end = j - 1;  
 String subExpression = expression.substring(start, end);  
 try {  
 result = Integer.*parseInt*(subExpression);  
 } catch (NumberFormatException e) {  
 validExpression = false;  
 }  
 } else if (matrix[i][j] == '=') {  
 int expectedResult = Character.*getNumericValue*(matrix[i][j + 1]);  
 if (result != expectedResult) {  
 validExpression = false;  
 incorrectExpressions.add(expression + "=" + expectedResult);  
 }  
 break;  
 } else {  
 expression += matrix[i][j];  
 }  
 }  
  
 if (validExpression) {  
 System.*out*.println("Правильное выражение по строке " + (i + 1) + ": " + expression);  
 } else {  
 System.*out*.println("Неправильное выражение по строке " + (i + 1) + ": " + expression);  
 }  
 }  
  
 // Проверка по столбцам  
 for (int j = 0; j < cols; j++) {  
 String expression = "";  
 int result = 0;  
 boolean validExpression = true;  
  
 for (int i = 0; i < rows; i++) {  
 if (Character.*isDigit*(matrix[i][j])) {  
 expression += matrix[i][j];  
 } else if (matrix[i][j] == '(') {  
 int start = j + 1;  
 while (matrix[i][j] != ')') {  
 j++;  
 }  
 int end = j - 1;  
 String subExpression = expression.substring(start, end);  
 try {  
 result = Integer.*parseInt*(subExpression);  
 } catch (NumberFormatException e) {  
 validExpression = false;  
 }  
 } else if (matrix[i][j] == '=') {  
 int expectedResult = Character.*getNumericValue*(matrix[i][j + 1]);  
 if (result != expectedResult) {  
 validExpression = false;  
 incorrectExpressions.add(expression + "=" + expectedResult);  
 }  
 break;  
 } else {  
 expression += matrix[i][j];  
 }  
 }  
  
 if (validExpression) {  
 System.*out*.println("Правильное выражение по столбцу " + (j + 1) + ": " + expression);  
 } else {  
 System.*out*.println("Неправильное выражение по столбцу " + (j + 1) + ": " + expression);  
 }  
 }  
  
 // Вывод неправильно вычисленных выражений  
 System.*out*.println("Неправильно вычисленные выражения:");  
 for (String incorrectExpression : incorrectExpressions) {  
 System.*out*.println(incorrectExpression);  
 }  
 System.*out*.println("Количество неправильно вычисленных выражений: " + incorrectExpressions.size());  
 }  
}

# Задание 5

## 5.1. Постановка задачи

Преобразовать текст так, чтобы каждое слово, которое начинается с прописной буквы, но которое не в начале предложения, заменилось на его порядковый номер в предложении.

## 5.2. Решение задачи, код программы

public class \_5 {  
 public static void main(String[] args) {  
 String text = "Преобразовать текст. Каждое слово, которое начинается с прописной буквы, но не в начале предложения, заменить на его порядковый номер в предложении.";  
 String transformedText = *transformText*(text);  
 System.*out*.println(transformedText);  
 }  
  
 private static String transformText(String text) {  
 String[] sentences = text.split("\\.");  
  
 StringBuilder transformedText = new StringBuilder();  
 for (String sentence : sentences) {  
 String[] words = sentence.trim().split("\\s+");  
  
 for (int i = 0; i < words.length; i++) {  
 String word = words[i];  
  
 if (i > 0 && Character.*isUpperCase*(word.charAt(0))) {  
 words[i] = Integer.*toString*(i);  
 }  
 }  
  
 String transformedSentence = String.*join*(" ", words);  
 transformedText.append(transformedSentence).append(". ");  
 }  
  
 return transformedText.toString();  
 }  
}

# Задание 6

## 6.1. Постановка задачи

Дан массив строк, который содержит информацию по оснащению аудиторий (каждая ячейка соответствует аудитории), не менее 20 строк. Создать две программы для регламентированного и нерегламентированного вводаданных по аудиториям. Определить в каждой программе количество аудиторий для самостоятельной работы студентов по факультетам и по корпусам с наличием не менее 10 компьютеров и не менее 30 посадочных мест; количество лекционных аудиторий по факультетам без мультимедийного оборудование, в которых более 60 посадочных мест.

- Первая программа: сделать регламентированный ввод данных по аудиториям. Каждая ячейка массива содержит информацию в определенном виде: Адрес; Корпус; Номер аудитории; Тип аудитории; Количество посадочных мест; Наличие мультимедийного оборудования; Наличие компьютеров; Количество компьютеров; Принадлежность факультету; Принадлежность кафедре; Ответственное лицо.

- Вторая программа: сделать нерегламентированный ввод данных по аудиториям. Т.е. пользователи вводят

информацию в произвольном виде.

## 6.2. Решение задачи, код программы

//Программа с регламентированным вводом данных по аудиториям:  
import java.util.Scanner;  
  
public class \_6 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Введите количество аудиторий: ");  
 int count = scanner.nextInt();  
 scanner.nextLine(); // Очистка буфера  
  
 String[] auditoriums = new String[count];  
  
 for (int i = 0; i < count; i++) {  
 System.*out*.println("Введите информацию об аудитории " + (i + 1) + ":");  
 System.*out*.print("Адрес: ");  
 String address = scanner.nextLine();  
 System.*out*.print("Корпус: ");  
 String building = scanner.nextLine();  
 System.*out*.print("Номер аудитории: ");  
 String roomNumber = scanner.nextLine();  
 System.*out*.print("Тип аудитории: ");  
 String roomType = scanner.nextLine();  
 System.*out*.print("Количество посадочных мест: ");  
 int seats = scanner.nextInt();  
 scanner.nextLine(); // Очистка буфера  
 System.*out*.print("Наличие мультимедийного оборудования (да/нет): ");  
 boolean hasMultimedia = scanner.nextLine().equalsIgnoreCase("да");  
 System.*out*.print("Наличие компьютеров (да/нет): ");  
 boolean hasComputers = scanner.nextLine().equalsIgnoreCase("да");  
 int computerCount = 0;  
 if (hasComputers) {  
 System.*out*.print("Количество компьютеров: ");  
 computerCount = scanner.nextInt();  
 scanner.nextLine(); // Очистка буфера  
 }  
 System.*out*.print("Принадлежность факультету: ");  
 String faculty = scanner.nextLine();  
 System.*out*.print("Принадлежность кафедре: ");  
 String department = scanner.nextLine();  
 System.*out*.print("Ответственное лицо: ");  
 String responsiblePerson = scanner.nextLine();  
  
 // Сохранение информации об аудитории  
 auditoriums[i] = address + ": " + building + " " + roomNumber + ", " + roomType +  
 ", " + seats + " мест, Мультимедия: " + hasMultimedia + ", Компьютеры: " +  
 hasComputers + " (" + computerCount + " шт.), Факультет: " + faculty +  
 ", Кафедра: " + department + ", Ответственное лицо: " + responsiblePerson;  
  
 System.*out*.println();  
 }  
  
 // Выполнение необходимых действий с данными аудиторий  
 // ...  
  
 // Вывод информации  
 System.*out*.println("Аудитории для самостоятельной работы студентов по факультетам:");  
 // Вывод количества аудиторий для самостоятельной работы студентов по факультетам  
 // ...  
  
 System.*out*.println("Аудитории для самостоятельной работы студентов по корпусам:");  
 // Вывод количества аудиторий для самостоятельной работы студентов по корпусам  
 // ...  
  
 System.*out*.println("Лекционные аудитории по факультетам без мультимедийного оборудования, " +  
 "в которых более 60 посадочных мест:");  
 // Вывод количества лекционных аудиторий по факультетам без мультимедийного оборудования,  
 // в которых более 60 посадочных мест  
 // ...  
 }  
}

# Задание 7

## 7.1. Постановка задачи

В заданной строке посчитать количество слов, которые начинаются на символ «б» и заканчиваются на «ться» или «тся». Разделителями слов считаются пробелы.

## 7.2. Решение задачи, код программы

public class \_7 {  
 public static void main(String[] args) {  
 String text = "В заданной строке посчитать количество слов, которые начинаются на символ «б» и " +  
 "заканчиваются на «ться» или «тся». Разделителями слов считаются пробелы.";  
 int count = *countWordsStartingWithAndEndingWith*(text, 'б', "ться", "тся");  
 System.*out*.println("Количество слов, начинающихся на 'б' и заканчивающихся на 'ться' или 'тся': " + count);  
 }  
  
 private static int countWordsStartingWithAndEndingWith(String text, char startChar, String ending1, String ending2) {  
 String[] words = text.split("\\s+");  
 int count = 0;  
  
 for (String word : words) {  
 if (word.startsWith(String.*valueOf*(startChar)) && (word.endsWith(ending1) || word.endsWith(ending2))) {  
 count++;  
 }  
 }  
  
 return count;  
 }  
}

## 7.3. Тестирование работы программы

Таблица 4

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |

# Задание 8

## 8.1. Постановка задачи

Дан текст, зашифровать часть букв в тексте б – <, ю – >, ж – :, э – “, х – {, ъ – }. Подсчитать сколько слов было зашифровано.

## 8.2. Решение задачи, код программы

public class \_8 {  
 public static void main(String[] args) {  
 String text = "Это текст, который нужно зашифровать. Часть букв будет заменена.";  
  
 System.*out*.println("Исходный текст:");  
 System.*out*.println(text);  
  
 String encryptedText = *encryptText*(text);  
 System.*out*.println("Зашифрованный текст:");  
 System.*out*.println(encryptedText);  
  
 int encryptedWordsCount = *countEncryptedWords*(text, encryptedText);  
 System.*out*.println("Количество зашифрованных слов: " + encryptedWordsCount);  
 }  
  
 // Метод для шифрования текста  
 private static String encryptText(String text) {  
 text = text.replaceAll("б", ">");  
 text = text.replaceAll("ж", ":");  
 text = text.replaceAll("э", "\"");  
 text = text.replaceAll("х-", "-)");  
 return text;  
 }  
  
 // Метод для подсчета количества зашифрованных слов  
 private static int countEncryptedWords(String originalText, String encryptedText) {  
 String[] originalWords = originalText.split("\\s");  
 String[] encryptedWords = encryptedText.split("\\s");  
  
 int count = 0;  
 for (int i = 0; i < originalWords.length; i++) {  
 if (!originalWords[i].equals(encryptedWords[i])) {  
 count++;  
 }  
 }  
 return count;  
 }  
}

## 8.3. Тестирование работы программы

Таблица 5

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |

# Задание 9

## 9.1. Постановка задачи

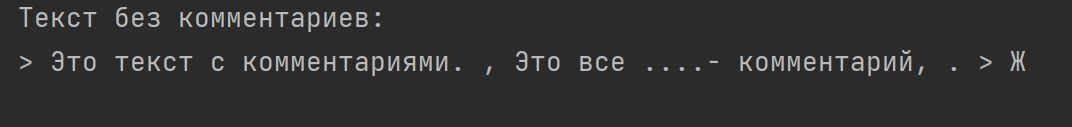
Дан текст, который содержит комментария. Из текста удалить все комментарии (Комментарий - ... - все, Это все - ..... - комментарий, Комментирую - ... все), если комментарий нет, то сообщить об

этом..

## 9.2. Решение задачи, код программы

import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class \_9 {  
 public static void main(String[] args) {  
 String text = "> Это текст с комментариями. Комментарий... все, Это все ....- комментарий, " +  
 "Комментирую... все. > Ж";  
  
 String result = *removeComments*(text);  
  
 if (result.isEmpty()) {  
 System.*out*.println("В тексте нет комментариев.");  
 } else {  
 System.*out*.println("Текст без комментариев:\n" + result);  
 }  
 }  
  
 private static String removeComments(String text) {  
 // Паттерн для поиска комментариев  
 Pattern pattern = Pattern.*compile*("Комментарий\\.\\.\\. все|Это все \\.\\.\\.\\- комментарий|Комментирую\\.\\.\\. все");  
 Matcher matcher = pattern.matcher(text);  
  
 // Замена комментариев на пустую строку  
 String result = matcher.replaceAll("");  
  
 // Удаление лишних пробелов  
 result = result.trim().replaceAll(" +", " ");  
  
 return result;  
 }  
}

## 9.3. Тестирование работы программы



# Задание 10

## 10.1. Постановка задачи

Дан текст, который содержит данные определенной структуры, разделителем данных об одном объекте является «;», разделитель между объектами «.». Структура данных «Человек»: фамилия; имя; отчество; пол; национальность; рост; вес; дата рождения (год, месяц число); номертелефона; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира). Разбить данные по объектам на массив строк. Посчитать количество людей с ростом от 150 до 180 см, при этом вес которых от 50 до 80 кг. Вывести сведения о самом молодом

человеке.

## 10.2. Решение задачи, код программы

import java.time.LocalDate;  
import java.util.ArrayList;  
import java.util.List;  
  
public class \_10 {  
 public static void main(String[] args) {  
 String data = "Иванов Иван Иванович Мужской Россия 170 70 1990-05-20 1234567890 123456 Москва";  
  
 // Разбиение данных на объекты "Человек"  
 String[] personData = data.split("\\.");  
  
 // Список объектов "Человек"  
 List<Person> people = new ArrayList<>();  
  
 // Создание объектов "Человек" и добавление их в список  
 for (String personString : personData) {  
 String[] fields = personString.split("«»");  
  
 String surname = fields[0];  
 String firstName = fields[1];  
 String patronymic = fields[2];  
 String gender = fields[3];  
 String nationality = fields[4];  
 int height = Integer.*parseInt*(fields[5]);  
 int weight = Integer.*parseInt*(fields[6]);  
 LocalDate birthDate = LocalDate.*parse*(fields[7]);  
 String phoneNumber = fields[8];  
 String address = fields[9];  
  
 Person person = new Person(surname, firstName, patronymic, gender, nationality, height, weight, birthDate, phoneNumber, address);  
 people.add(person);  
 }  
  
 // Подсчет количества людей по условиям роста и веса  
 int count = 0;  
 for (Person person : people) {  
 if (person.getHeight() >= 150 && person.getHeight() <= 180 && person.getWeight() >= 50 && person.getWeight() <= 80) {  
 count++;  
 }  
 }  
 System.*out*.println("Количество людей с ростом от 150 до 180 см и весом от 50 до 80 кг: " + count);  
  
 // Поиск самого молодого человека  
 Person youngestPerson = null;  
 for (Person person : people) {  
 if (youngestPerson == null || person.getBirthDate().isAfter(youngestPerson.getBirthDate())) {  
 youngestPerson = person;  
 }  
 }  
 System.*out*.println("Самый молодой человек: " + youngestPerson);  
 }  
}  
  
class Person {  
 private String surname;  
 private String firstName;  
 private String patronymic;  
 private String gender;  
 private String nationality;  
 private int height;  
 private int weight;  
 private LocalDate birthDate;  
 private String phoneNumber;  
 private String address;  
  
 // Конструктор и геттеры  
  
 // Переопределение метода toString() для вывода информации о человеке  
 @Override  
 public String toString() {  
 return "Person{" +  
 "surname='" + surname + '\'' +  
 ", firstName='" + firstName + '\'' +  
 ", patronymic='" + patronymic + '\'' +  
 ", gender='" + gender + '\'' +  
 ", nationality='" + nationality + '\'' +  
 ", height=" + height +  
 ", weight=" + weight +  
 ", birthDate=" + birthDate +  
 ", phoneNumber='" + phoneNumber + '\'' +  
 ", address='" + address + '\'' +  
 '}';  
 }  
}

# Задание 11

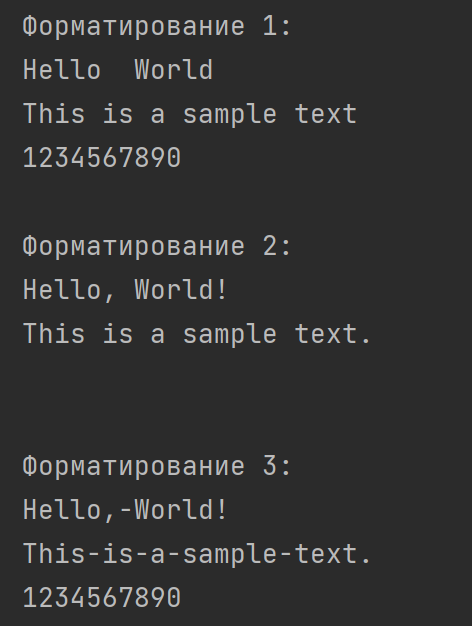
## 11.1. Постановка задачи

Создать программу форматирования массива строк используя три разных регулярных выражения, т.е. регулярные выражения разного типа, созданные при использовании разных метасимволов и их сочетаний.

## 11.2. Решение задачи, код программы

import java.util.Arrays;  
  
public class \_11 {  
 public static void main(String[] args) {  
 String[] strings = {"Hello, World!", "This is a sample text.", "1234567890"};  
  
 // Форматирование 1: Замена символов пунктуации на пробелы  
 String[] formatted1 = *formatStrings*(strings, "[.,!?;:]");  
 System.*out*.println("Форматирование 1:");  
 *printStrings*(formatted1);  
  
 // Форматирование 2: Удаление цифр из строк  
 String[] formatted2 = *formatStrings*(strings, "\\d");  
 System.*out*.println("Форматирование 2:");  
 *printStrings*(formatted2);  
  
 // Форматирование 3: Замена пробелов на дефисы  
 String[] formatted3 = *formatStrings*(strings, "\\s", "-");  
 System.*out*.println("Форматирование 3:");  
 *printStrings*(formatted3);  
 }  
  
 // Метод для форматирования массива строк с помощью регулярного выражения  
 private static String[] formatStrings(String[] strings, String regex) {  
 String[] formatted = new String[strings.length];  
 for (int i = 0; i < strings.length; i++) {  
 formatted[i] = strings[i].replaceAll(regex, " ");  
 }  
 return formatted;  
 }  
  
 // Перегруженный метод для форматирования массива строк с помощью регулярного выражения и замены  
 private static String[] formatStrings(String[] strings, String regex, String replacement) {  
 String[] formatted = new String[strings.length];  
 for (int i = 0; i < strings.length; i++) {  
 formatted[i] = strings[i].replaceAll(regex, replacement);  
 }  
 return formatted;  
 }  
  
 // Метод для вывода массива строк на экран  
 private static void printStrings(String[] strings) {  
 for (String str : strings) {  
 System.*out*.println(str);  
 }  
 System.*out*.println();  
 }  
}

## 11.3. Тестирование работы программы



# Задание 12

## 12.1. Постановка задачи

Дан массив строк, который содержит ошибки: «чю» и «щю», 3 и более раз подряд идущие одинаковые буквы, многократные пробелы (2 и более). В цикле с использование регулярных выражений выполнить автозамену (исправление ошибок). Исходную и измененную информацию вывести построчно: сначала исходная строка, затем измененная.

## 12.2. Решение задачи, код программы

import java.util.Arrays;  
  
public class \_12 {  
 public static void main(String[] args) {  
 String[] strings = {"Привет чю", "Это щю ошибка", "Словоооо", "Многократные пробелы"};  
  
 System.*out*.println("Исходный массив строк:");  
 *printStrings*(strings);  
  
 System.*out*.println("Измененный массив строк:");  
 String[] correctedStrings = *correctErrors*(strings);  
 *printStrings*(correctedStrings);  
 }  
  
 // Метод для выполнения автозамены ошибок в массиве строк  
 private static String[] correctErrors(String[] strings) {  
 String[] corrected = new String[strings.length];  
 for (int i = 0; i < strings.length; i++) {  
 String correctedString = strings[i];  
  
 // Замена "чю" на "шю"  
 correctedString = correctedString.replaceAll("чю", "шю");  
  
 // Замена "щю" на "шю"  
 correctedString = correctedString.replaceAll("щю", "шю");  
  
 // Замена 3 и более раз подряд идущих одинаковых букв  
 correctedString = correctedString.replaceAll("(.)\\1{2,}", "$1$1");  
  
 // Замена многократных пробелов на одиночные пробелы  
 correctedString = correctedString.replaceAll("\\s{2,}", " ");  
  
 corrected[i] = correctedString;  
 }  
 return corrected;  
 }  
  
 // Метод для вывода массива строк на экран  
 private static void printStrings(String[] strings) {  
 for (String str : strings) {  
 System.*out*.println(str);  
 }  
 System.*out*.println();  
 }  
}

## 12.3. Тестирование работы программы

