Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 10

тема «Файлы»

по дисциплине «Информатика»

Выполнил: студент группы ИСТ-22-1б Богданов Д.О.

Проверил: Нетбай Г.В.

Пермь, 2023

**Содержание**

[Задание 1 4](#_Toc136801676)

[1.1. Постановка задачи 4](#_Toc136801677)

[1.2. Решение задачи, код программы 4](#_Toc136801678)

[1.3. Тестирование работы программы с проверкой 5](#_Toc136801679)

[Задание 2 6](#_Toc136801680)

[2.1. Постановка задачи 6](#_Toc136801681)

[2.2. Решение задачи, код программы 6](#_Toc136801682)

[2.3. Тестирование работы программы с проверкой 6](#_Toc136801683)

[Задание 3 7](#_Toc136801684)

[3.1. Постановка задачи 7](#_Toc136801685)

[3.2. Решение задачи, код программы 7](#_Toc136801686)

[3.3. Тестирование работы программы с проверкой 7](#_Toc136801687)

[Задание 4 8](#_Toc136801688)

[4.1. Постановка задачи 8](#_Toc136801689)

[4.2. Решение задачи, код программы 8](#_Toc136801690)

[4.3. Тестирование работы программы 8](#_Toc136801691)

[Задание 5 10](#_Toc136801692)

[5.1. Постановка задачи 10](#_Toc136801693)

[5.2. Решение задачи, код программы 10](#_Toc136801694)

[5.3. Тестирование работы программы с проверкой 10](#_Toc136801695)

[Задание 6.2. 12](#_Toc136801696)

[6.1. Постановка задачи 12](#_Toc136801697)

[6.2. Решение задачи, код программы 12](#_Toc136801698)

[6.3. Тестирование работы программы с проверкой 12](#_Toc136801699)

[Задание 7 13](#_Toc136801700)

[7.1. Постановка задачи 13](#_Toc136801701)

[7.2. Решение задачи, код программы 13](#_Toc136801702)

[7.3. Тестирование работы программы с проверкой 13](#_Toc136801703)

[Задание 8 14](#_Toc136801704)

[8.1. Постановка задачи 14](#_Toc136801705)

[8.2. Решение задачи, код программы 14](#_Toc136801706)

[8.3. Тестирование работы программы с проверкой 14](#_Toc136801707)

[Задание 9 15](#_Toc136801708)

[9.1. Постановка задачи 15](#_Toc136801709)

[9.2. Решение задачи, код программы 15](#_Toc136801710)

[9.3. Тестирование работы программы с проверкой 16](#_Toc136801711)

[Задание 10 17](#_Toc136801712)

[10.1. Постановка задачи 17](#_Toc136801713)

[10.2. Решение задачи, код программы 17](#_Toc136801714)

[10.3. Тестирование работы программы с проверкой 17](#_Toc136801715)

[Задание 11 19](#_Toc136801716)

[11.1. Постановка задачи 19](#_Toc136801717)

[11.2. Решение задачи, код программы 19](#_Toc136801718)

[11.3. Тестирование работы программы с проверкой 19](#_Toc136801719)

[Задание 12 21](#_Toc136801720)

[12.1. Постановка задачи 21](#_Toc136801721)

[12.2. Решение задачи, код программы 21](#_Toc136801722)

[12.3. Тестирование работы программы с проверкой 21](#_Toc136801723)

[Задание 13 23](#_Toc136801724)

[13.1. Постановка задачи 23](#_Toc136801725)

[13.2. Решение задачи, код программы 23](#_Toc136801726)

[13.3. Тестирование работы программы с проверкой 24](#_Toc136801727)

[Задание 14 25](#_Toc136801728)

[14.1. Постановка задачи 25](#_Toc136801729)

[14.2. Решение задачи, код программы 25](#_Toc136801730)

[14.3. Тестирование работы программы с проверкой 26](#_Toc136801731)

[Задание 15 27](#_Toc136801732)

[15.1. Постановка задачи 27](#_Toc136801733)

[15.2. Решение задачи, код программы 27](#_Toc136801734)

[15.3. Тестирование работы программы с проверкой 30](#_Toc136801735)

# Задание 1

## 1.1. Постановка задачи

Переделать программу задания 3 лабораторной работы 9. Массив данных записан в файл и считывается в программе для обработки. Результаты обработки массива согласно вопросам задания 3 лабораторной работы 9 записываются в конец файла построчно с заголовками расшифровывающими, что за данные представлены ниже.

## 1.2. Решение задачи, код программы

import java.io.\*;  
public class Lab10Tsk1 {  
  
  
  
 public static void main(String[] args) throws IOException {  
 File file = new File("C:\\Users\\Rirva\\Desktop\\filexsi\\t1.txt");  
 FileWriter fwr = new FileWriter(file, true);  
 BufferedReader br = new BufferedReader(new FileReader(file));  
 fwr.write("\n");  
 String str = br.readLine();  
 char [] arr = str.toCharArray();  
 int cnt1 = 0;  
 for (char c : arr) {  
 if (Character.*isUpperCase*(c) && Character.UnicodeBlock.*of*(c) == Character.UnicodeBlock.*CYRILLIC*) {  
 cnt1++;  
 }  
 }  
 fwr.write("Количество прописных букв русского языка: " + cnt1+"\n");  
 char a = ',';  
 boolean flag = false;  
 for (char c : arr) {  
 if (c == a) {  
 flag = true;  
 break;  
 }  
 }  
 fwr.write("Знак препинания " + a + " присутствует: " + flag+"\n");  
 String word = "ada";  
 int cnt2 = 0;  
 for (char c : arr) {  
 if (word.indexOf(c) != -1) {  
 cnt2++;  
 }  
 if (cnt2 >= 2) {  
 break;  
 }  
 }  
 fwr.write("Среди символов имеются хотя бы две буквы из слова: " + (cnt2 >= 2)+"\n");  
 boolean flag2 = false;  
 for (int i = 0; i < arr.length - 1; i++) {  
 if ((arr[i] == 'т' && arr[i + 1] == 'о') || (arr[i] == 'о' && arr[i + 1] == 'т')) {  
 flag2 = true;  
 break;  
 }  
 }  
 fwr.write("Имеется ли среди символов пара соседствующих букв «то» или «от»: " + flag2+"\n");  
 boolean flag3 = false;  
 for (int i = 0; i < arr.length - 2; i++) {  
 if (arr[i] == arr[i + 1] && arr[i] == arr[i + 2]) {  
 flag3 = true;  
 break;  
 }  
 }  
 fwr.write("Имеется ли среди символов три подряд идущих одинаковых символа: " + flag3+"\n");  
 boolean flagE = false;  
 for (int i = arr.length / 2 + 1; i < arr.length - 1; i++) {  
 for (int j = i + 1; j < arr.length - 1; j++) {  
 if (Character.*isDigit*(arr[i]) && Character.*isDigit*(arr[i + 1]) && Math.*abs*(arr[i] - arr[i + 1]) == 1 &&  
 Character.*isLetter*(arr[j]) && Character.*isLetter*(arr[j + 1]) && arr[j] == arr[j + 1]) {  
 flagE = true;  
 break;  
 }  
 }  
 if (flagE) {  
 break;  
 }  
 }  
 fwr.write("Верно ли условие е: " + flagE+"\n");  
 fwr.flush();  
 fwr.close();  
 }  
 }

## 1.3. Тестирование работы программы с проверкой

Таблица 1

Тестирование работы программы и проверка результатов решения

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

# Задание 2

## 2.1. Постановка задачи

Записать в файл таблицу из задания 7 лабораторной работы 6. Если в таблице присутствуют дробные ответы, то вывести данные в формате 4 знака после запятой.

## 2.2. Решение задачи, код программы

import java.io.File;  
import java.io.FileWriter;  
import java.io.IOException;  
public class Lab10Tsk2 {  
 public static void main(String[] args) throws IOException {  
 File file1 = new File("C:\\Users\\Rirva\\Desktop\\filexsi\\t2.txt");  
 FileWriter fwr = new FileWriter(file1);  
 String[][] array = *table*(9, 9);  
 StringBuilder sb = new StringBuilder();  
 for (int i = 0; i < array.length; i++){  
 for (int j = 0; j < array[i].length; j++){  
 sb.append(i + 1).append(" - ").append(j + 1).append(" = ").append(array[i][j]).append("\t");  
 }  
 fwr.write((sb) + "\n");  
 sb = new StringBuilder();  
 }  
 fwr.flush();  
 fwr.close();  
}  
 public static String[][] table(int lines, int columns){  
 String[][] array = new String[lines][columns];  
 for (int i = 0; i < array.length; i++){  
 for (int j = 0; j < array[i].length; j++){  
 array[i][j] = String.*valueOf*((i + 1)-(j + 1));  
 }  
 }  
 return array;  
 }  
}

## 2.3. Тестирование работы программы с проверкой

Тестирование работы программы и проверка результатов решения

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

# Задание 3

## 3.1. Постановка задачи

3. Напечатать все слова из текстового файла, содержащие хотя бы одну маленькую латинскую букву.

## 3.2. Решение задачи, код программы

import java.io.\*;  
public class Lab10Tsk3 {  
 public static void main(String[] args) throws IOException{  
 BufferedReader br = new BufferedReader(new FileReader("C:\\Users\\Rirva\\Desktop\\filexsi\\t3.txt"));  
 FileWriter fwr = new FileWriter("C:\\Users\\Rirva\\Desktop\\filexsi\\t3.txt", true);  
 String line; fwr.write("\nOUT\n");  
 while ((line = br.readLine()) != null) {  
 if (line.matches(".\*[a-z].\*")) {  
 String[] words = line.split("\\s+");  
 for (String word : words) {  
 if (word.matches(".\*[a-z].\*")) {  
 fwr.write(word + "\n");  
 }  
 }  
 }  
 }  
 fwr.flush();  
 fwr.close();  
 }  
}

## 3.3. Тестирование работы программы с проверкой

Таблица 3

Тестирование работы программы и проверка результатов решения

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

# Задание 4

## 4.1. Постановка задачи

Выделить в текстовом файле все слова, разделенные символами-разделителями «\_.,;:\n\t!?», и все слова, в которых количество гласных русских букв равно количеству согласных русских букв, построчно записать в другой текстовый файл.

## 4.2. Решение задачи, код программы

import java.io.\*;  
public class Lab10Tsk4 {  
 public static void main(String[] args) throws IOException{  
 File fileIN = new File("C:\\Users\\Rirva\\Desktop\\filexsi\\t4.1.txt");  
 File fileOUT = new File("C:\\Users\\Rirva\\Desktop\\filexsi\\t4.2.txt");  
 FileReader fread = new FileReader(fileIN);  
 BufferedReader bufferedReader = new BufferedReader(fread);  
 FileWriter fwr = new FileWriter(fileOUT);  
 BufferedWriter bufferedWriter = new BufferedWriter(fwr);  
 String line;  
 while ((line = bufferedReader.readLine()) != null) {  
 String[] words = line.split("[\_.,;:\\n\\t!?\\s]+");  
 for (String word : words) {  
 if (*isBalanced*(word)) {  
 bufferedWriter.write(word);  
 bufferedWriter.newLine();  
 }  
 }  
 }  
 fwr.flush();  
 bufferedReader.close();  
 bufferedWriter.close();  
 fwr.close();  
 }  
  
 private static boolean isBalanced(String word) {  
 int cnt1 = 0;  
 int cnt2 = 0;  
 for (int i = 0; i < word.length(); i++) {  
 char c = Character.*toLowerCase*(word.charAt(i));  
 if (c == 'а' || c == 'е' || c == 'ё' || c == 'и' || c == 'о' || c == 'у' || c == 'ы' || c == 'э' || c == 'ю' || c == 'я') {  
 cnt1++;  
 } else if (c >= 'а' && c <= 'я') {  
 cnt2++;  
 }  
 }  
 return cnt1 == cnt2;  
 }  
}

## 4.3. Тестирование работы программы

Таблица 4

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 5

## 5.1. Постановка задачи

Дано два файла с предложениями. Дописать в конец второго файла предложения из первого файла, в которых есть палиндромы.

## 5.2. Решение задачи, код программы

import java.io.\*;  
public class Lab10Tsk5 {  
 public static void main(String[] args) throws IOException {  
 File fileIN = new File("C:\\Users\\Rirva\\Desktop\\filexsi\\t5.1.txt");  
 File fileOUT = new File("C:\\Users\\Rirva\\Desktop\\filexsi\\t5.2.txt");  
 BufferedReader fread = new BufferedReader(new FileReader(fileIN));  
 BufferedWriter fwr = new BufferedWriter(new FileWriter(fileOUT, true));  
 String line1;  
 fwr.write("\nOUT\n");  
 while ((line1 = fread.readLine()) != null) {  
 String [] words = line1.split("\\s");  
 for (String word : words){  
 if (*isPalindrome*(word)) {  
 fwr.write(word + "\n");  
 }  
 }  
 }  
 fwr.flush();  
 fwr.close();  
 }  
  
 public static boolean isPalindrome(String str) {  
 int i = 0, j = str.length() - 1;  
 while (i < j) {  
 if (str.charAt(i) != str.charAt(j)) {  
 return false;  
 }  
 i++;  
 j--;  
 }  
 return true;  
 }  
}

## 5.3. Тестирование работы программы с проверкой

Таблица 5

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 6.2.

## 6.1. Постановка задачи

Входной файл содержит сведения о кофе: страна изготовления, масса пачки, тип, цена. Найти пачки с зерном и массой от 150 до 380 гр., подсчитать их количество и дописать ответ в конец файла.

## 6.2. Решение задачи, код программы

import java.io.BufferedReader;  
import java.io.BufferedWriter;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
public class Lab10Tsk6 {  
 public static void main(String[] args) throws IOException{  
 int cnt = 0;  
 BufferedReader fread = new BufferedReader(new FileReader("C:\\Users\\Rirva\\Desktop\\filexsi\\t6.txt"));  
 BufferedWriter fwr = new BufferedWriter(new FileWriter("C:\\Users\\Rirva\\Desktop\\filexsi\\t6.txt",true));  
 String line; fwr.write("\nOUT\n");  
 while ((line = fread.readLine()) != null) {  
 String[] coffeeData = line.split(",");  
 if (coffeeData.length >= 3) {  
 String type = coffeeData[2].trim();  
 int weight = Integer.*parseInt*(coffeeData[1].trim());  
 if ("зерно".equals(type) && weight >= 150 && weight <= 380) {  
 cnt++;  
 fwr.write(line);  
 fwr.newLine();  
 }  
 }  
 }  
 fwr.write("Количество пачек с зерном и массой от 150 до 380 гр.: " + cnt);  
 fwr.flush();  
 fwr.close();  
 }  
}

## 6.3. Тестирование работы программы с проверкой

Таблица 6

Тестирование работы программы и проверка результатов решения

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

# Задание 7

## 7.1. Постановка задачи

В заданной строке посчитать количество разных символов, входящих в эту строку. Создать новую строку, состоящую из букв первой строки и количества их вхождений в нее.

## 7.2. Решение задачи, код программы

import static java.lang.Math.\*;  
import java.io.File;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.text.DecimalFormat;  
public class Lab10Tsk7 {  
 public static void main(String[] args) throws IOException {  
 File file1 = new File("C:\\Users\\Rirva\\Desktop\\filexsi\\t7.txt");  
 FileWriter fwr = new FileWriter(file1);  
 fwr.write("Условия:\n\n(x >= -3 && x <= 3); x += 0.5\n\nПри x >= 0, x^2-5\n\nВ противном -x^2+5\n\nИтог\n\n"  
 );  
 DecimalFormat d = new DecimalFormat("#.#");  
 double x = -3;  
 while (x <= 3){  
 if (x >= 0) fwr.write( "f(" + x + ")\t=\t" + d.format(*pow*(x,2)-5) + "\n");  
 else fwr.write("f(" + x + ")\t=\t" + d.format(-*pow*(x,2)+5) + "\n");  
 x += 0.5;  
 }  
 fwr.flush();  
 fwr.close();  
 }  
}

Таблица 7

## 7.3. Тестирование работы программы с проверкой

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

# Задание 8

## 8.1. Постановка задачи

Составить файл постановки и решения задачи для попадания точки в область лабораторная работа 5 задание 5.

Структура файла: постановка задачи; массив точек для проверки; стилизованный ответ.

Постановка задачи (для каждой фигуры, с заголовком о номере фигуры): уравнение кривой с порядковым номером, отрезок на котором она определена.

Массив точек для проверки: номер точки, координата x; координата y.

Стилизованный ответ: номер точки, ответ о попадании точки в область, на границу или о том, что точка не попала.

## 8.2. Решение задачи, код программы

Таблица 8

## 8.3. Тестирование работы программы с проверкой

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

# Задание 9

## 9.1. Постановка задачи

 Создать файл с данными по структурированному вводу данных согласно постановке задачи 6 лабораторной работы 9. Изменить программу задания 6 лабораторной работы 9 с анализом данных по тем же критериям из файла. Ответы построчно дописать в исходный файл.

## 9.2. Решение задачи, код программы

import java.io.\*;  
import java.util.\*;  
public class tsk9 {  
 public static void main(String[] args) throws IOException {  
 File file = new File("C:\\Users\\Rirva\\Desktop\\filexsi\\t9.txt");  
 FileWriter fwr = new FileWriter(file, true);  
  
 BufferedReader fread = new BufferedReader(new FileReader(file));  
  
 String[] array = new String[3];  
  
 for (int i = 0; i < array.length ; i++) {  
 array[i] = fread.readLine();  
 }  
  
 int cntOver1Mln = 0;  
 int cntWithGP = 0;  
  
 for (String row : array) {  
  
 String[] data = row.split(" ");  
  
 int funding = Integer.*parseInt*(data[1]);  
 int groupSize= Integer.*parseInt*(data[2]);  
  
 if (funding > 1000000 && groupSize < 5)  
 cntOver1Mln++;  
  
 int students = Integer.*parseInt*(data[3]);  
 int aspirants = Integer.*parseInt*(data[4]);  
  
 if (students >= 2 && aspirants > 0)  
 cntWithGP++;  
 }  
  
 fwr.write("Количество проектов с финансированием более 1 млн: " + cntOver1Mln + "\n");  
  
 fwr.write("Количество проектов с аспирантами и студентами: " + cntWithGP + "\n");  
  
 Map<String, Integer> facultyCnt = new HashMap<>();  
  
 for (String row : array) {  
 String[] data = row.split(" ");  
 String faculty = data[0];  
 facultyCnt.put(faculty, facultyCnt.getOrDefault(faculty, 0) + 1);  
 }  
  
 String mapAsString = facultyCnt.toString();  
 fwr.write(mapAsString);  
 fwr.flush();  
 fwr.close();  
 }  
  
}

}

## 9.3. Тестирование работы программы с проверкой

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

# Задание 10

## 10.1. Постановка задачи

На основе программы задания 5 лабораторной работы 6 организовать запись в файл данных о результатах нахождения функции по ряду Маклорена.

Примерный вид файла ответов:

Функция f(x) = … для x = … равняется …

Результаты определения значений функции f(x) = … с помощью ряда Маклорена

Погрешность итерационной процедуры …

Значение функции по Маклорену Погрешность, % Число итераций

           10                           4                 1

## 10.2. Решение задачи, код программы

import java.io.\*;  
import java.util.Scanner;  
  
import static java.lang.Math.\*;  
  
public class Lab10Tsk10 {  
 public static void main(String[] args) throws IOException {  
 File file1 = new File("C:\\Users\\Rirva\\Desktop\\filexsi\\t10.txt");  
 FileWriter fwr = new FileWriter(file1, true);  
 Scanner in = new Scanner(System.*in*);  
 double x;  
 System.*out*.println("Введите значение x");  
 while (Math.*abs*(x = in.nextDouble()) <= 1){  
 System.*out*.println("Ошибка! Введите другое значение x");  
 }  
  
 for (double ep = 0.0001; ep <= 0.01; ep\*=10){  
 int n=0;  
 double sum=0;  
 while ((*abs*(*atan*(x)-((*PI*/2)+sum)))>ep) {  
 sum += (*pow*(-1,n+1))\*(1d/(((2\*n)+1)\*(*pow*(x,((2\*n)+1)))));  
 n++;  
 }  
 sum=*PI*/2+sum;  
 fwr.write("Функция f(x) = arctg(x) для x = " + x + " равняется " + sum + "\n"  
 +"Результаты определения значений функции f(x) = arctg(x) с помощью ряда Маклорена\n"  
 +"Погрешность итерационной процедуры "+ep+"\n"  
 +"Значение функции по Маклорену\t\tПогрешность, %\tЧисло итераций\n"  
 +"\t"+sum+"\t\t\t\t"+ep\*100+"\t\t\t"+n+"\n\n");  
 }  
 fwr.flush();  
 fwr.close();  
 }  
}

Таблица 10

## 10.3. Тестирование работы программы с проверкой

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

# 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.io.\*;  
import java.text.DecimalFormat;  
import static java.lang.Math.\*;  
public class Lab10Tsk11 {  
 public static void main(String[] args) throws IOException {  
 File file = new File("C:\\Users\\Rirva\\Desktop\\filexsi\\t11.txt");  
 FileWriter fwr = new FileWriter(file);  
 DecimalFormat dec = new DecimalFormat("#.#");  
 fwr.write("z = sin(x)+cos^2(x),\n" +  
 "x:[0;4]; шаг 0.2\ny:[0;4]; шаг 0.2\n" +  
 "\t\t\t\t\t\t\t\nx/y\t");  
 for (double y = 0; y <= 4; y+=0.2){  
 fwr.write(dec.format(y) + "\t");  
 }  
 fwr.write("\n");  
 for (double x = 0; x <= 4; x+=0.2){  
 fwr.write(dec.format(x) + "\t|");  
 for (double y = 0; y <= 4; y+=0.2){  
 Double z = *sin*(x)+*cos*(x)\**cos*(x);  
 fwr.write(dec.format(z) + "|\t|");  
 }  
 fwr.write("\n");  
 }  
 fwr.flush();  
 fwr.close();  
 }  
  
}

Таблица 11

## 11.3. Тестирование работы программы с проверкой

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

# Задание 12

## 12.1. Постановка задачи

12. Переделать программу задания 12 лабораторной работы 9 по работе с регулярными выражениями. Дан файл с текстом, применить анализ и редактирование текст файла с использованием ранее написанного алгоритма форматирования текста. Записать отредактированный текст в файл после исходного текста через пустую строку и заголовок.

## 12.2. Решение задачи, код программы

import java.io.\*;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
public class Lab10Tsk12 {  
 public static void main(String[] args) throws IOException {  
 File file = new File("C:\\Users\\Rirva\\Desktop\\filexsi\\t12.txt");  
 BufferedReader fread = new BufferedReader(new FileReader(file));  
 BufferedWriter fwr = new BufferedWriter(new FileWriter(file,true));  
 String line = fread.readLine();  
 fwr.write("\nOUT\n");  
 String [] array = line.split(";");  
 Pattern pattern = Pattern.*compile*("(\\d+)\\s+учащихся,\\s+(\\d+)\\s+отличник");  
 for (String month : array) {  
 fwr.write("Исходная строка: " + month+"\n");  
 Matcher matcher = pattern.matcher(month);  
 if (matcher.find()) {  
 int students = Integer.*parseInt*(matcher.group(1));  
 int excellent = Integer.*parseInt*(matcher.group(2));  
 String newMonth = month.replaceAll("(\\d+)\\s+учащихся", *getDeclension*(students, "учащийся", "учащихся", "учащих"))  
 .replaceAll("(\\d+)\\s+отличник", *getDeclension*(excellent, "отличник", "отличника", "отличников"));  
  
 fwr.write("Измененная строка: " + newMonth+"\n");  
 } else {  
 fwr.write("Ошибка: не найдены данные о численности учащихся и отличников"+"\n");  
 }  
 }  
 fwr.flush();  
 fwr.close();  
 }  
 private static String getDeclension(int number, String one, String two, String five) {  
 int mod10 = number % 10;  
 int mod100 = number % 100;  
 if (mod10 == 1 && mod100 != 11) {  
 return number + " " + one;  
 } else if (mod10 >= 2 && mod10 <= 4 && (mod100 < 10 || mod100 >= 20)) {  
 return number + " " + two;  
 } else {  
 return number + " " + five;  
 }  
 }  
}

Таблица 12

## 12.3. Тестирование работы программы с проверкой

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |

# Задание 13

## 13.1. Постановка задачи

12. Переделать программу задания 12 лабораторной работы 9 по работе с регулярными выражениями. Дан файл с текстом, применить анализ и редактирование текст файла с использованием ранее написанного алгоритма форматирования текста. Записать отредактированный текст в файл после исходного текста через пустую строку и заголовок.

## 13.2. Решение задачи, код программы

import java.io.\*;  
import java.util.\*;  
public class Lab10Tsk13 {  
 public static void main(String[] args) throws IOException {  
 int k = 2; int p = 2; int n = 3;  
 int[][][] array1 = new int[k][n][n];  
 Scanner in = new Scanner(new File("C:\\Users\\Rirva\\Desktop\\filexsi\\t13.1.txt"));  
 for (int i = 0; i < k; i++) {  
 for (int j = 0; j < n; j++) {  
 for (int l = 0; l < n; l++) {  
 array1[i][j][l] = in.nextInt();  
 }  
 }  
 }  
 System.*out*.println("Содержимое первого файла:");  
 for (int[][] matrix : array1) {  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 System.*out*.print(matrix[i][j] + " ");  
 }  
 System.*out*.println();  
 }  
 System.*out*.println();  
 }  
 int[][][] array2 = new int[p][n][n];  
 Scanner in1 = new Scanner(new File("C:\\Users\\Rirva\\Desktop\\filexsi\\t13.2.txt"));  
 for (int i = 0; i < p; i++) {  
 for (int j = 0; j < n; j++) {  
 for (int m = 0; m < n; m++) {  
 array2[i][j][m] = in1.nextInt();  
 }  
 }  
 }  
 System.*out*.println("Содержимое второго файла:");  
 for (int[][] matrix : array2) {  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 System.*out*.print(matrix[i][j] + " ");  
 }  
 System.*out*.println();  
 }  
 System.*out*.println();  
 }  
 int minSum = Integer.*MAX\_VALUE*;  
 int[][] minMatrix = null;  
 for (int[][] matrix : array1) {  
 int sum = 0;  
 for (int i = 0; i < n; i++) {  
 sum += matrix[i][i];  
 }  
 if (sum < minSum) {  
 minSum = sum;  
 minMatrix = matrix;  
 }  
 }  
 boolean flag = true;  
 for (int[][] matrix : array2) {  
 int sum = 0;  
 for (int i = 0; i < n; i++) {  
 sum += matrix[i][i];  
 }  
 if (sum <= minSum) {  
 flag = false;  
 break;  
 }  
 }  
 if (flag) {  
 PrintWriter fwr = new PrintWriter(new FileWriter("C:\\Users\\Rirva\\Desktop\\filexsi\\t13.2.txt", true));  
 fwr.write("\nВывод\n");  
 fwr.println();  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < n; j++) {  
 fwr.print(minMatrix[i][j] + " ");  
 }  
 fwr.println();  
 }  
 fwr.flush();  
 fwr.close();  
 }  
 }  
}

Таблица 13

## 13.3. Тестирование работы программы с проверкой

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

# Задание 14

## 14.1. Постановка задачи

Дан файл результатов исследования (листинг натурного эксперимента, результаты численного решения задачи и т.п.). Выполнить анализ данных: 1. Найти максимальное, минимальное и среднее значение параметров по столбцу; 2. Отклонение каждого значения параметра от среднего.

Записать данные в конец файла построчно. Сначала ответ на первый вопрос, потом ответ на 2 вопрос.

## 14.2. Решение задачи, код программы

import java.io.\*;  
import java.util.\*;  
  
public class Lab10Tsk14 {  
 public static void main(String[] args) throws IOException {  
  
 File file = new File("C:\\Users\\Rirva\\Desktop\\filexsi\\tsk141.txt");  
 BufferedReader reader = new BufferedReader(new FileReader(file));   
  
 List<List<Double>> columns = new ArrayList<>();   
  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] tokens = line.trim().split("\\s+");   
 for (int i = 0; i < tokens.length; i++) {  
 if (columns.size() <= i) {  
 columns.add(new ArrayList<>());   
 }  
 double value = Double.*parseDouble*(tokens[i]);   
 columns.get(i).add(value);   
 }  
 }  
  
 reader.close();   
  
 FileWriter fw = new FileWriter(file, true);   
  
   
 for (int i = 0; i < columns.size(); i++) {  
 List<Double> column = columns.get(i);  
  
   
 double max = Collections.*max*(column);  
 double min = Collections.*min*(column);  
 double avg = column.stream().mapToDouble(a -> a).sum() / column.size();  
  
   
 String text = "\nColumn " + (i + 1) + ": Max = " + max + ", Min = " + min + ", Average = " + avg;  
 fw.write(text);  
 fw.write(System.*lineSeparator*());  
  
   
 for (double value : column) {  
 double deviation = value - avg;  
  
 String deviationStr = String.*valueOf*(deviation);  
  
 fw.write(deviationStr);  
 fw.write(" ");  
 }  
 fw.write(System.*lineSeparator*());  
 }  
  
 fw.close();   
 }  
}

Таблица 14

## 14.3. Тестирование работы программы с проверкой

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

# Задание 15

## 15.1. Постановка задачи

 Создать файл с координатами 10 участков произвольной четырехугольной формы. Создать файл с областями обитания животных на территории: волк, заяц. Данные заполнены с координатами обитания животных: волк эллипс, заяц окружность. Дан файл с природными объектами на территории: озера, ключи. Данные заполнены с координатами объектов: координаты центра и радиус.

Для каждого участка определить количество животных, которые на нем обитают и наличие водоема. Сформировать информацию об участках в файл ответа по типу:

————————

N участка Кол-во зон с животными % S, занимаемой животными кол-во водоемов

    1                4                        3                        2

...

## 15.2. Решение задачи, код программы

import java.io.File;  
import java.io.FileNotFoundException;  
import java.util.ArrayList;  
import java.util.Scanner;  
public class Lab10Tsk15 {  
 public static void main(String[] args) {  
 // Чтение данных из файлов  
 ArrayList<Polygon> plots = *readPlotsFromFile*("C:\\Users\\Rirva\\Desktop\\filexsi\\plots.txt");  
 ArrayList<AnimalArea> animalAreas = *readAnimalAreasFromFile*("C:\\Users\\Rirva\\Desktop\\filexsi\\animal\_areas.txt");  
 ArrayList<NatureObject> natureObjects = *readNatureObjectsFromFile*("C:\\Users\\Rirva\\Desktop\\filexsi\\nature\_objects.txt");  
  
 // Определение для каждого участка количества животных и наличия водоема  
 ArrayList<PlotInfo> plotInfos = new ArrayList<>();  
 for (int i = 0; i < plots.size(); i++) {  
 Polygon plot = plots.get(i);  
 int animalAreaCount = 0;  
 double animalAreaFraction = 0.0;  
 int waterBodyCount = 0;  
 for (AnimalArea animalArea : animalAreas) {  
 if (plot.contains(animalArea.getCenter())) {  
 animalAreaCount++;  
 animalAreaFraction += animalArea.getArea() / plot.getArea();  
 }  
 }  
 for (NatureObject natureObject : natureObjects) {  
 if (plot.contains(natureObject.getCenter())) {  
 waterBodyCount++;  
 }  
 }  
 plotInfos.add(new PlotInfo(i + 1, animalAreaCount, animalAreaFraction, waterBodyCount));  
 }  
  
 // Вывод результатов в файл  
 *writePlotInfosToFile*("C:\\Users\\Rirva\\Desktop\\filexsi\\plot\_infos.txt", plotInfos);  
 }  
  
 // Класс для хранения информации об участке  
 public static class PlotInfo {  
 public int plotNumber;  
 public int animalAreaCount;  
 public double animalAreaFraction;  
 public int waterBodyCount;  
  
 public PlotInfo(int plotNumber, int animalAreaCount, double animalAreaFraction, int waterBodyCount) {  
 this.plotNumber = plotNumber;  
 this.animalAreaCount = animalAreaCount;  
 this.animalAreaFraction = animalAreaFraction;  
 this.waterBodyCount = waterBodyCount;  
 }  
 }  
  
 // Чтение участков из файла  
 public static ArrayList<Polygon> readPlotsFromFile(String filename) {  
 ArrayList<Polygon> plots = new ArrayList<>();  
 try {  
 Scanner scanner = new Scanner(new File(filename));  
 while (scanner.hasNextLine()) {  
 String line = scanner.nextLine();  
 String[] parts = line.split(" ");  
 Polygon plot = new Polygon();  
 for (int i = 0; i < parts.length; i += 2) {  
 double x = Double.*parseDouble*(parts[i]);  
 double y = Double.*parseDouble*(parts[i + 1]);  
 plot.addPoint(x, y);  
 }  
 plots.add(plot);  
 }  
 scanner.close();  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 }  
 return plots;  
 }  
  
 // Чтение областей обитания животных из файла  
 public static ArrayList<AnimalArea> readAnimalAreasFromFile(String filename) {  
 ArrayList<AnimalArea> animalAreas = new ArrayList<>();  
 try {  
 Scanner scanner = new Scanner(new File(filename));  
 while (scanner.hasNextLine()) {  
 String line = scanner.nextLine();  
 String[] parts = line.split(" ");  
 double x = Double.*parseDouble*(parts[0]);  
 double y = Double.*parseDouble*(parts[1]);  
 double radius1 = Double.*parseDouble*(parts[2]);  
 double radius2 = Double.*parseDouble*(parts[3]);  
 animalAreas.add(new AnimalArea(x, y, radius1, radius2));  
 }  
 scanner.close();  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 }  
 return animalAreas;  
 }  
  
 // Чтение природных объектов из файла  
 public static ArrayList<NatureObject> readNatureObjectsFromFile(String filename) {  
 ArrayList<NatureObject> natureObjects = new ArrayList<>();  
 try {  
 Scanner scanner = new Scanner(new File(filename));  
 while (scanner.hasNextLine()) {  
 String line = scanner.nextLine();  
 String[] parts = line.split(" ");  
 double x = Double.*parseDouble*(parts[0]);  
 double y = Double.*parseDouble*(parts[1]);  
 double radius = Double.*parseDouble*(parts[2]);  
 natureObjects.add(new NatureObject(x, y, radius));  
 }  
 scanner.close();  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 }  
 return natureObjects;  
 }  
  
 // Запись информации об участках в файл  
 public static void writePlotInfosToFile(String filename, ArrayList<PlotInfo> plotInfos) {  
 try {  
 java.io.FileWriter writer = new java.io.FileWriter(filename);  
 writer.write("N участка\tКол-во зон с животными\t\t\t% S, занимаемой животными\tкол-во водоемов\n");  
 for (PlotInfo plotInfo : plotInfos) {  
 writer.write(String.*format*("%d\t\t\t\t%d\t\t\t\t%.2f\t\t\t\t%d\n",  
 plotInfo.plotNumber,  
 plotInfo.animalAreaCount,  
 plotInfo.animalAreaFraction \* 100,  
 plotInfo.waterBodyCount));  
 }  
 writer.close();  
 } catch (java.io.IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 // Класс для хранения точки на плоскости  
 public static class Point {  
 public double x;  
 public double y;  
  
 public Point(double x, double y) {  
 this.x = x;  
 this.y = y;  
 }  
 }  
  
 // Класс для хранения многоугольника  
 public static class Polygon {  
 private ArrayList<Point> points = new ArrayList<>();  
  
 public void addPoint(double x, double y) {  
 points.add(new Point(x, y));  
 }  
  
 public double getArea() {  
 double area = 0.0;  
 int j = points.size() - 1;  
 for (int i = 0; i < points.size(); i++) {  
 Point p1 = points.get(i);  
 Point p2 = points.get(j);  
 area += (p2.x + p1.x) \* (p2.y - p1.y);  
 j = i;  
 }  
 return Math.*abs*(area / 2.0);  
 }  
  
 public boolean contains(Point p) {  
 int i, j;  
 boolean c = false;  
 for (i = 0, j = points.size() - 1; i < points.size(); j = i++) {  
 if (((points.get(i).y > p.y) != (points.get(j).y > p.y)) &&  
 (p.x < (points.get(j).x - points.get(i).x) \* (p.y - points.get(i).y) /  
 (points.get(j).y - points.get(i).y) + points.get(i).x)) {  
 c = !c;  
 }  
 }  
 return c;  
 }  
 }  
  
 // Класс для хранения области обитания животного  
 public static class AnimalArea {  
 public Point center;  
 public double radius1;  
 public double radius2;  
  
 public AnimalArea(double x, double y, double radius1, double radius2) {  
 this.center = new Point(x, y);  
 this.radius1 = radius1;  
 this.radius2 = radius2;  
 }  
  
 public Point getCenter() {  
 return center;  
 }  
  
 public double getArea() {  
 return Math.*PI* \* radius1 \* radius2;  
 }  
 }  
  
 // Класс для хранения природного объекта  
 public static class NatureObject {  
 public Point center;  
 public double radius;  
  
 public NatureObject(double x, double y, double radius) {  
 this.center = new Point(x, y);  
 this.radius = radius;  
 }  
  
 public Point getCenter() {  
 return center;  
 }  
   
 }  
}

}

Таблица 14

## 15.3. Тестирование работы программы с проверкой

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