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 в Java»

по дисциплине «Информатика»

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

Проверил: ассистент каф. ВММБ Нетбай Г.В.

Пермь, 2023

**Содержание**

[Задание 1 4](#_Toc136717036)

[1.1. Постановка задачи 4](#_Toc136717037)

[1.2. Решение задачи, код программы 4](#_Toc136717038)

[1.3. Тестирование работы программы с проверкой 5](#_Toc136717039)

[Задание 2 7](#_Toc136717040)

[2.1. Постановка задачи 7](#_Toc136717041)

[2.2. Решение задачи, код программы 7](#_Toc136717042)

[2.3. Тестирование работы программы с проверкой 7](#_Toc136717043)

[Задание 3 8](#_Toc136717044)

[3.1. Постановка задачи 8](#_Toc136717045)

[3.2. Решение задачи, код программы 8](#_Toc136717046)

[3.3. Тестирование работы программы с проверкой 8](#_Toc136717047)

[Задание 4 9](#_Toc136717048)

[4.1. Постановка задачи 9](#_Toc136717049)

[4.2. Решение задачи, код программы 9](#_Toc136717050)

[4.3. Тестирование работы программы с проверкой 9](#_Toc136717051)

[Задание 5 11](#_Toc136717052)

[5.1. Постановка задачи 11](#_Toc136717053)

[5.2. Решение задачи, код программы 11](#_Toc136717054)

[5.3. Тестирование работы программы с проверкой 11](#_Toc136717055)

[Задание 6 12](#_Toc136717056)

[6.1. Постановка задачи 12](#_Toc136717057)

[6.2. Решение задачи, код программы 12](#_Toc136717058)

[6.3. Тестирование работы программы с проверкой 13](#_Toc136717059)

[Задание 7 14](#_Toc136717060)

[7.1. Постановка задачи 14](#_Toc136717061)

[7.2. Решение задачи, код программы 14](#_Toc136717062)

[7.3. Тестирование работы программы с проверкой 14](#_Toc136717063)

[Задание 8 16](#_Toc136717064)

[8.1. Постановка задачи 16](#_Toc136717065)

[8.2. Решение задачи, код программы 16](#_Toc136717066)

[8.3. Тестирование работы программы с проверкой 18](#_Toc136717067)

[Задание 9 20](#_Toc136717068)

[9.1. Постановка задачи 20](#_Toc136717069)

[9.2. Решение задачи, код программы 20](#_Toc136717070)

[9.3. Тестирование работы программы с проверкой 20](#_Toc136717071)

[Задание 11 22](#_Toc136717072)

[11.1. Постановка задачи 22](#_Toc136717073)

[11.2. Решение задачи, код программы 22](#_Toc136717074)

[11.3. Тестирование работы программы с проверкой 23](#_Toc136717075)

[Задание 12 25](#_Toc136717076)

[12.1. Постановка задачи 25](#_Toc136717077)

[12.2. Решение задачи, код программы 25](#_Toc136717078)

[12.3. Тестирование работы программы с проверкой 26](#_Toc136717079)

[Задание 13 28](#_Toc136717080)

[13.1. Постановка задачи 28](#_Toc136717081)

[13.2. Решение задачи, код программы 28](#_Toc136717082)

[13.3. Тестирование работы программы с проверкой 30](#_Toc136717083)

[Задание 14 31](#_Toc136717084)

[14.1. Постановка задачи 31](#_Toc136717085)

[14.2. Решение задачи, код программы 31](#_Toc136717086)

[14.3. Тестирование работы программы с проверкой 32](#_Toc136717087)

# 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.util.Scanner;  
import java.io.\*;  
public class zad\_1{  
 public static String *line* = "";  
 public static void zad\_1() throws IOException {  
 String[] mas = *FileGetStrMas*("laba\_10\_zad\_1.txt");  
 Scanner scn = new Scanner(System.*in*);  
 System.*out*.printf("Задайте проверяемую латинскую букву: ");  
 String n = scn.next();  
 *Check*(*Stats*(mas,n),n,"laba\_10\_zad\_1.txt");  
 }  
  
  
 public static String[] FileGetStrMas(String FileName) throws IOException {  
 BufferedReader br = new BufferedReader(new FileReader(FileName));  
 *line* = br.readLine();  
 br.close();  
 *line* = *line*.replaceAll("[{}]","");  
 String mas[] = *line*.split(",");  
 return mas;  
 }  
  
 public static void Check(int[] mas, String n, String FileName) throws IOException {  
 PrintWriter bw = new PrintWriter(new FileWriter(FileName));  
 bw.append("{"+*line*+"}"+"\n");  
 bw.append("\nРезультаты:\n");  
 bw.append("Прописных латинских букв "+ mas[0] +" штук \n");  
 if (mas[1] == 1) { bw.append("Буква <<"+ n +">> встречается в массиве\n");  
 } else { bw.append("Буква <<"+ n +">> не встречается в массиве\n");}  
 if (mas[2] >= 2){ bw.append("Запятая встречается в массиве 2-а или более раз\n");  
 } else { bw.append("Запятая встречается в массиве менее 2-х раз\n");}  
 if (mas[3] == 1) { bw.append("Среди элементов встречается <<во>> или <<ов>>\n");  
 } else { bw.append("Среди букв не встречается <<во>> или <<ов>>\n");}  
 if (mas[4] == 1) { bw.append("Среди элементов встречаются одинаковый цифры стоящие подряд\n");  
 } else { bw.append("Среди элементов не встречаются одинаковый цифры стоящие подряд\n");}  
 if (mas[5] == 1) { bw.append("Среди элементов встречаются одинаковые буквы " +  
 "разного регистра и нули стоящие подряд\n");  
 } else { bw.append("Среди элементов не встречаются одинаковые буквы " +  
 "разного регистра и нули стоящие подряд\n");}  
 bw.close();  
 }  
  
 public static int[] Stats(String[] mas, String n) {  
 int arr[] = new int[6];  
 for(int i = 0; i < mas.length; i++){  
 if (mas[i].matches("[А**-**яA**-**z]+")){  
 if (mas[i].matches("[А**-**Я]+")) {  
 arr[0]++;  
 }  
 if((mas[i].toUpperCase()).equals(n.toUpperCase())) {  
 arr[1] = 1;  
 }  
 if((i != mas.length - 1) && ((mas[i].equals("в") && mas[i+1].equals("о"))  
 || (mas[i].equals("о") && mas[i+1].equals("в")))){  
 arr[3] = 1;  
 }  
 if(i > 1 && i != mas.length-1 && !mas[i].equals(mas[i+1]) && (mas[i].toUpperCase().equals(mas[i+1])  
 || mas[i].toLowerCase().equals(mas[i+1]))) {  
 for (int j = i + 1; j < mas.length - 1; j++) {  
 if (mas[j].equals("0") && mas[j + 1].equals("0")) {  
 arr[5] = 1;  
 }  
 }  
 }  
 }else if (mas[i].equals(",")){  
 arr[2]++;  
 }else if (i != mas.length - 1 && mas[i].matches("[0**-**9]+")  
 && mas[i+1].matches("[0**-**9]+") && mas[i].equals(mas[i+1])){  
 arr[4] = 1;  
 }  
 }  
 return arr;  
 }  
}

## 1.3. Тестирование работы программы с проверкой

Таблица 1.1

Тестирование работы программы и проверка результатов решения

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

# 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.\*;  
  
public class zad\_2 extends MyFiles{  
 public static void zad\_2() throws IOException {  
 *CreateNewFile*("D:\\Java\\laba-10","laba\_10\_zad\_2.txt");  
 FileWriter fw = new FileWriter("laba\_10\_zad\_2.txt");  
 for (int i = 1; i <= 9; i++){  
 for (int j = 1; j <= 9; j++){  
 fw.write(j+" / "+i+" = "+(j/i)+"\t");  
 }  
 fw.write("\n");  
 }  
 fw.close();  
 }  
}

## 2.3. Тестирование работы программы с проверкой

Таблица 2.1

Тестирование работы программы и проверка результатов решения

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

# Задание 3

## 3.1. Постановка задачи

Напечатать все слова из текстового файла, содержащие аббревиатуру ABBA или АББА.

.

## 3.2. Решение задачи, код программы

import java.io.\*;  
  
public class zad\_3 {  
 public static void zad\_3() throws IOException {  
 BufferedReader br = new BufferedReader(new FileReader("laba\_10\_zad\_3.txt"));  
 System.*out*.println("Слова содержащие аббревиатуру ABBA или АББА:");  
 String line;  
 while((line = br.readLine()) != null){  
 if (line.matches(".\*(ABBA|АББА).\*")){  
 String[] words = line.split("\\s+");  
 for (String word : words) {  
 if (word.matches(".\*(ABBA|АББА).\*")) {  
 System.*out*.println(word);  
 }  
 }  
 }  
 }  
 br.close();  
 }  
}

## 3.3. Тестирование работы программы с проверкой

Таблица 3.1

Тестирование работы программы и проверка результатов решения

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

# Задание 4

## 4.1. Постановка задачи

Выделить в текстовом файле все слова, разделенные символами-разделителями «\_.,;:\n\t!?», и все слова, которые состоят из прописных гласных латинских букв и цифр, построчно записать в другой текстовый файл.

## 4.2. Решение задачи, код программы

import java.io.\*;  
  
public class zad\_4 {  
 public static void zad\_4() throws IOException {  
 BufferedReader bf = new BufferedReader(new FileReader("laba\_10\_zad\_4\_1.txt"));  
 PrintWriter pw = new PrintWriter("laba\_10\_zad\_4\_2.txt");  
 String line;  
 String last = "";  
 int cnt1 = 0;  
 while((line = bf.readLine()) != null){  
 String[] words = line.split("\\s+");  
 int cnt2 = 0;  
 for(int i = 0;i < words.length;i++){  
 if (i != words.length-1 && words[i].matches(".+[\_.,;:\\t!?]")){  
 pw.println(words[i]+" "+words[i+1]);  
 }  
 if (!last.equals("") && cnt2++ < 1 && cnt1++ > 0) {  
 pw.println(last + " " + words[0]);  
 }  
 last = words[words.length-1];  
 int cnt3 = 0;  
 int cnt4 = 0;  
 for (int j = 0;j < words[i].length();j++){  
 if ((words[i].charAt(j) >= 'A' && words[i].charAt(j) <= 'Z')  
 || (words[i].charAt(j) >= 'a' && words[i].charAt(j) <= 'z')){  
 cnt3++;  
 }else if(Character.*isDigit*(words[i].charAt(j))){  
 cnt4++;  
 }  
 }  
 if (cnt3 + cnt4 == words[i].length() && cnt3 > 0 && cnt4 > 0) {  
 pw.println(words[i]);  
 }  
 }  
 }  
 pw.close();  
 bf.close();  
 }  
}

## 4.3. Тестирование работы программы с проверкой

Таблица 4.1

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 цифры I, II, III, IV, V.

## 5.2. Решение задачи, код программы

import java.io.\*;  
import java.util.ArrayList;  
  
public class zad\_5 extends MyFiles {  
 public static void zad\_5() throws IOException {  
 ArrayList<String> arr1 = *FileReadInArr*("laba\_10\_zad\_5\_2.txt");  
 ArrayList<String> arr2 = *FileReadInArr*("laba\_10\_zad\_5\_1.txt");  
 *RefreshFile\_2*(arr1,arr2,"laba\_10\_zad\_5\_2.txt");  
 }  
  
 public static void RefreshFile\_2(ArrayList<String> arr1, ArrayList<String> arr2,String FileName) throws FileNotFoundException {  
 PrintWriter pw = new PrintWriter(FileName);  
 for(String unit: arr1){  
 pw.println(unit);  
 }  
 for(String unit: arr2){  
 if (unit.matches(".\*[IV].\*")) {  
 pw.println(unit);  
 }  
 }  
 pw.close();  
 }  
}

## 5.3. Тестирование работы программы с проверкой

Таблица 5.1

Тестирование работы программы и проверка результатов решения

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

# Задание 6

## 6.1. Постановка задачи

Входной файл содержит сведения о волейболистах пермского края: ФИО, клуб за который выступает, с какого года, результативность в очках, цена контракта. Подсчитать количество волейболистов по клубам с ценой контракта не менее 100 000, участвующих в играх клуба не менее 5 лет, дописать ответы построчно в конец файла.

## 6***.2. Решение задачи, код программы***

import java.util.Scanner;  
  
public import java.io.\*;  
import java.util.ArrayList;  
  
public class zad\_6 extends MyFiles {  
 public static void zad\_6() throws IOException {  
 *RefreshFile*("laba\_10\_zad\_6.txt");  
 }  
  
 public static void RefreshFile(String FileName) throws IOException {  
 ArrayList<String> persons = *FileReadInArr*(FileName);  
 int[] counts = *Check*(persons);  
 PrintWriter pw = new PrintWriter(FileName);  
 for(String person: persons){  
 pw.println(person);  
 }  
 pw.println("Число волейболистов по клубам с ценой контракта не менее 100 000: "+counts[0]);  
 pw.println("Число волейболистов по клубам участвующих в играх клуба не менее 5 лет: "+counts[1]);  
 pw.close();  
 }  
  
  
 public static int[] Check(ArrayList<String> persons) {  
 int[] counts = new int[2];  
 String teams = "";  
 for (String person : persons) {  
 String[] facts = person.split(",");  
 if (!teams.matches(".\*" + facts[1] + ".\*")) {  
 teams += facts[1] + ",";  
 }  
 }  
 for (String team : teams.split(",")) {  
 for (String person : persons) {  
 person = person.replace('.', ',');  
 String[] facts = person.split(",");  
 if (person.matches(".\*" + team + ".\*")) {  
 if (Integer.*parseInt*(facts[3]) >= 100000) {  
 counts[0]++;  
 }  
 if (2023 - Integer.*parseInt*(facts[2]) >= 5) {  
 counts[1]++;  
 }  
 }  
 }  
 }  
 return counts;  
 }  
}

## 6.3. Тестирование работы программы с проверкой

Таблица 6.1

Тестирование работы программы и проверка результатов решения

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

# Задание 7

## 7.1. Постановка задачи

Записать в файл значения функции f(x) из задания 3 лабораторной работы 5 для всего диапазона переменной в виде: значение x; значение f(x). В начале файла должна быть указана постановка задачи, т.е. диапазон аргумента и вид функции (запись на языке java). Перед выводом цифровых значений должен быть сделан заголовок о том, в каком порядке идет вывод.

## 7.2. Решение задачи, код программы

import java.io.IOException;  
import java.io.PrintWriter;  
  
import static java.lang.Math.*sin*;  
import static java.lang.Math.*tan*;  
  
public class zad\_7 extends MyFiles{  
 public static void zad\_7() throws IOException {  
 *CreateNewFile*("D:\\Java\\laba-10","laba\_10\_zad\_7.txt");  
 *Calc*("laba\_10\_zad\_7.txt");  
  
 }  
  
 public static void Calc(String FileName) throws IOException {  
 PrintWriter pw = new PrintWriter(FileName);  
 pw.println("Постановка: f(x) = tan(x)+sin(x), где -3 <= x <= 3\n");  
 pw.println("Вычисления начинаются с шагом x += 0.5:");  
 for (double x = -3; x <= 3; x += 0.5){  
 if (x >= 1){  
 pw.println(x + "\t\t\t" + (*tan*(x)+*sin*(x)));  
 } else if(x >= 0 && x < 1) {  
 pw.println(x + "\t\t\t" + (*tan*(x) + *sin*(x)));  
 }else{  
 pw.println(x + "\t\t" + (5\*x\*x - 3));  
 }  
 }  
 pw.close();  
 }  
}

## 7.3. Тестирование работы программы с проверкой

Таблица 5.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № 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

import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.util.Scanner;  
  
public class zad\_8 extends MyFiles{  
 public static void zad\_8() throws IOException {  
 *CreateNewFile*("D:\\Java\\laba-10","laba\_10\_zad\_8.txt");  
 *File*("laba\_10\_zad\_8.txt");  
 }  
 public static void File(String FileName) throws FileNotFoundException {  
 PrintWriter pw = new PrintWriter(FileName);  
 pw.println("Постановка:");  
 pw.println("Коэфициенты находятся следующим образом:\n");  
 pw.println("k(x1,y1,x2,y2) = (y1 - y2)/(x1 - x2); b(x1,y1,x2,y2) = y1 - x1\*k(x1,y1,x2,y2)\n");  
 pw.println("Левая фигура!\nФигура 1:");  
 pw.println("№1 y = x \* k(-4, 0, 1, 4) + b(-4, 0, 1, 4) D: -4 <= x <= 1");  
 pw.println("№2 y = x \* k(1, 4, -7, 5) + b(1, 4, -7, 5) D: -7 <= x <= 1");  
 pw.println("№3 y = x \* k(-4, 0, -7, 5) + b(-4, 0, -7, 5) D: -7 <= x <= -4");  
 pw.println("Фигура 2:");  
 pw.println("№1 y = x \* k(-4, 0, -7, 5) + b(-4, 0, -7, 5) D: -7 <= x <= -4");  
 pw.println("№2 y = x \* k(-6, -4, -7, 5) + b(-6, -4, -7, 5) D: -7 <= x <= -6");  
 pw.println("№3 y = x \* k(-6, -4, -4, 0) + b(-6, -4, -4, 0) D: -6 <= x <= -4");  
 pw.println("Фигура 2:");  
 pw.println("№1 y = x \* k(-6, -4, -4, 0) + b(-6, -4, -4, 0) D: -6 <= x <= -4");  
 pw.println("№2 y = x \* k(-2, -6, -6, -4) + b(-2, -6, -6, -4) D: -6 <= x <= -2");  
 pw.println("№3 y = x \* k(-2, -6, -4, 0) + b(-2, -6, -4, 0) D: -4 <= x <= -2\n");  
 pw.println("Правая фигура!\nФигура 4:");  
 pw.println("№1 y = x \* k(1, -7, 3, -4) + b(1, -7, 3, -4) D: 1 <= x <= 3");  
 pw.println("№2 y = x \* k(1, -2, 3, -1) + b(1, -2, 3, -1) D: 1 <= x <= 3");  
 pw.println("№3 y = x \* k(-6, -4, -4, 0) + b(-6, -4, -4, 0) D: -6 <= x <= -4");  
 pw.println("№4 x = 1 D: x > 1");  
 pw.println("№5 x = 3 D: x < 3");  
 pw.println("Фигура 5:");  
 pw.println("№1 y = x \* k(3, -1, 5,2) + b(3, -1, 5,2) D: 3 <= x <= 5");  
 pw.println("№2 y = x \* k(3, -4, 6, -3) + b(3, -4, 6, -3) D: 3 <= x <= 6");  
 pw.println("№3 y = x \* k(6, -3, 5,2) + b(6, -3, 5,2) D: 5 <= x <= 6");  
 pw.println("№4 x = 3 D: x > 3\n");  
 Scanner scn = new Scanner(System.*in*);  
 pw.println("---------------------------------------------------------------------\n");  
 pw.println("Матрица данных:");  
 System.*out*.printf("Введите размерность матрици NxN значений x и y:\nN = ");  
 int N = scn.nextInt();  
 int[][] X = new int[N][N];  
 int[][] Y = new int[N][N];  
 int count = 0;  
 for(int i = 0; i < N;i++){  
 for(int j = 0; j < N;j++){  
 System.*out*.println("Точка №" + ++count);  
 System.*out*.printf("Задайте Х: ");  
 int x = scn.nextInt();  
 X[i][j] = x;  
 System.*out*.printf("Задайте Y: ");  
 int y = scn.nextInt();  
 Y[i][j] = y;  
 pw.write("№" + count + " ("+x+","+y+")\t");  
 }  
 pw.write("\n");  
 }  
 pw.write("\n");  
 pw.println("---------------------------------------------------------------------\n");  
 pw.println("Стилизованный вывод:");  
 count = 0;  
 for(int i = 0; i < N;i++){  
 for(int j = 0; j < N;j++){  
 int x,y;  
 x = X[i][j];  
 y = Y[i][j];  
 pw.printf("Точка №" + ++count + " ("+x+","+y+") ");  
 if (*figOne*(x,y) || *figTwo*(x,y) || *figThree*(x,y)) {  
 pw.printf("попала в левую фигуру");  
 } else if(*figFour*(x,y) || *figFive*(x,y)){  
 pw.printf("попала во правую фигуру");  
 } else {  
 pw.println("не попала");  
 }  
 }  
 }  
 pw.close();  
 }  
 public static double k(double x1,double y1,double x2,double y2){  
 return (y1 - y2)/(x1 - x2);  
 }  
 public static double b(double x1,double y1,double x2,double y2){  
 return y1 - x1\**k*(x1,y1,x2,y2);  
 }  
 public static boolean figOne(double x, double y) {  
 if (y >= x \* *k*(-4, 0, 1, 4) + *b*(-4, 0, 1, 4) &&  
 y <= x \* *k*(1, 4, -7, 5) + *b*(1, 4, -7, 5) &&  
 y >= x \* *k*(-4, 0, -7, 5) + *b*(-4, 0, -7, 5)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figTwo(double x, double y) {  
 if (y <= x \* *k*(-4, 0, -7, 5) + *b*(-4, 0, -7, 5) &&  
 y >= x \* *k*(-6, -4, -7, 5) + *b*(-6, -4, -7, 5) &&  
 y >= x \* *k*(-6, -4, -4, 0) + *b*(-6, -4, -4, 0)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figThree(double x, double y) {  
 if (y <= x \* *k*(-6, -4, -4, 0) + *b*(-6, -4, -4, 0) &&  
 y >= x \* *k*(-2, -6, -6, -4) + *b*(-2, -6, -6, -4) &&  
 y <= x \* *k*(-2, -6, -4, 0) + *b*(-2, -6, -4, 0)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figFour(double x, double y) {  
 if (y >= x \* *k*(1, -7, 3, -4) + *b*(1, -7, 3, -4) &&  
 y <= x \* *k*(1, -2, 3, -1) + *b*(1, -2, 3, -1) &&  
 x >= 1 && x <= 3) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figFive(double x, double y) {  
 if (y <= x \* *k*(3, -1, 5,2) + *b*(3, -1, 5,2) &&  
 y >= x \* *k*(3, -4, 6, -3) + *b*(3, -4, 6, -3) &&  
 y <= x \* *k*(6, -3, 5,2) + *b*(6, -3, 5,2) && x >= 3) {  
 return true;  
 }  
 return false;  
 }  
}

## 8.3. Тестирование работы программы с проверкой

Таблица 8.1

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № 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.util.Scanner;  
  
public class zad\_9 {  
 public void zad\_9() {  
 Scanner scn = new Scanner(System.*in*);  
 System.*out*.println("Введите текст: ");  
 String text = scn.nextLine();  
 String replace = *Replace\_Coms*(text);  
 if (replace.equals(text)) {  
 System.*out*.println("Комментарий в тексте нет");  
 } else {  
 System.*out*.printf("\nКомментарии удалены!\n\nИзмененный текст:\n%s", replace);  
 }  
 }  
  
 public static String Replace\_Coms(String text) {  
 String replace = text.replaceAll("-.+-", "");  
 replace = replace.replaceAll("\_.+\_", "");  
 replace = replace.replaceAll("=.+=", "");  
 return replace;  
 }  
}

## 9.3. Тестирование работы программы с проверкой

Таблица 9.1

Тестирование работы программы и проверка результатов решения

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

# Задание 11

## 11.1. Постановка задачи

Организовать запись данных о поверхности из лабораторной работы по диаграмма из УИР в файл. Файл содержит шапку о содержимом: формула поверхности, интервалы определения по координатам x и y.

Запись выполнить в определенном формате с понятными заголовками: массив координат x, массив координат y, матрица значений поверхности (z).

## 11.2. Решение задачи, код программы

import java.io.IOException;  
import java.io.PrintWriter;  
import static java.lang.Math.\*;  
  
  
public class zad\_11 extends MyFiles{  
 public static void zad\_11() throws IOException {  
 *CreateNewFile*("D:\\Java\\laba-10","laba\_10\_zad\_11.txt");  
 *Calc*("laba\_10\_zad\_11.txt");  
  
 }  
  
 public static void Calc(String FileName) throws IOException {  
 PrintWriter pw = new PrintWriter(FileName);  
 pw.println("Постановка: f(x) = cos(x)^2-x\*sin(y), где -2 <= x <= 3 и -1 <= y <= 4\n");  
 pw.println("Вычисления начинаются с шагом x += 0.1 и y += 0.2:\n");  
 pw.println("--------------------------------------------------------------------------\n");  
 pw.write("Массив координат X:\n{");  
 for (double x = -2; x <= 3; x += 0.1){  
 pw.printf("%.1f ",x);  
 }  
 pw.write("}\n");  
 pw.println("--------------------------------------------------------------------------\n");  
 pw.write("Массив координат Y:\n{");  
 for (double y = -1; y <= 4; y += 0.2){  
 pw.printf("%.1f ",y);  
 }  
 pw.write("}\n");  
 pw.println("--------------------------------------------------------------------------\n");  
 pw.println("Матрица значений Z:");  
 for (double x = -2; x <= 3; x += 0.1){  
 for (double y = -1; y <= 4; y += 0.2) {  
 if (x < 0){  
 pw.printf("%.3f\t\t",x);  
 } else {  
 pw.printf("%.3f \t\t",x);  
 }  
 if ((*pow*(*cos*(x),2)-x\**sin*(y)) < 0){  
 pw.printf("%.3f",(*pow*(*cos*(x),2)-x\**sin*(y)));  
 } else {  
 pw.printf("%.3f ",(*pow*(*cos*(x),2)-x\**sin*(y)));  
 }  
 }  
 pw.write("\n");  
 }  
 pw.close();  
 }  
}

## 11.3. Тестирование работы программы с проверкой

Таблица 11.1

Тестирование работы программы и проверка результатов решения

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

# Задание 12

## 12.1. Постановка задачи

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.IOException;  
import java.io.PrintWriter;  
import java.util.ArrayList;  
  
public class zad\_12 extends MyFiles {  
 public static void zad\_12() throws IOException {  
 String file ="laba\_10\_zad\_12.txt";  
 String[] messages = *ReadFile*(file);  
 PrintWriter pw = new PrintWriter(file);  
 pw.println("Исходный текст: ");  
 for (String message : messages) {  
 pw.println(message);  
 }  
 pw.println();  
 for (String message : *replaces*(messages, file)){  
 pw.println(message);  
 }  
 pw.close();  
 }  
  
 public static ArrayList<String> replaces(String[] messages, String FileName) throws IOException {  
 ArrayList<String> arr = new ArrayList<>();  
 arr.add("Исправленный текст: ");  
 String[] patterns = {"[бБ]удем говорить", "[нН]у", "[тТ]уто", "[Кк]ороче", "[Тт]ипа"};  
 String[] sleng = {"[кК]ат[ак][а**-**я]+", "[кК]ек[а**-**я]+", "[кК]раш[а**-**я]\*", "[лЛ]ол[а**-**я]\*",  
 "[нН]уб[а**-**я]\*", "ОМГ", "[Сс]трим[а**-**я]\*", "[фФ]лекс[а**-**я]\*",  
 "[чЧ]ил[а**-**я]\*", "[пП]адр[а**-**я]+", "[дД]нокл[а**-**я]\*"};  
 String[] trans = {"(играть)", "(смеяться)", "(пример подражания)", "(очень смешно)",  
 "(новичек)", "(крайнее удивление)", "(трансляция)", "(быть увлеченным)",  
 "(отдых)", "(подруга)", "(одноклассник)"};  
 for (String message : messages) {  
 message = message.replaceAll("\\s{2,}", " ");  
 for (String pattern : patterns) {  
 message = message.replaceAll(pattern + "[.,!?]?\\s\*", "");  
 }  
 String[] words = message.split("\\s+");  
 for (String word : words) {  
 String clearWord = word.replaceAll("[^А**-**Яа**-**я]", "");  
 message = message.replaceAll(clearWord + ",?\\s+,?" + clearWord, clearWord);  
 message = message.replaceAll(clearWord.substring(0, 1).toUpperCase()  
 + clearWord.substring(1) + ",?\\s+,?" + clearWord, clearWord);  
 message = message.replaceAll(clearWord + ",?\\s+,?" + clearWord.substring(0, 1).toUpperCase()  
 + clearWord.substring(1), clearWord.substring(0, 1).toUpperCase()  
 + clearWord.substring(1));  
 }  
 words = message.split("\\s+");  
 String NEWmessage = "";  
 for (int i = 0; i < words.length; i++) {  
 String word = words[i];  
 if (word.equals(words[0])) {  
 word = word.substring(0, 1).toUpperCase() + word.substring(1);  
 } else if (words[i - 1].charAt(words[i - 1].length() - 1) == '.' ||  
 words[i - 1].charAt(words[i - 1].length() - 1) == '?' ||  
 words[i - 1].charAt(words[i - 1].length() - 1) == '!') {  
 word = word.substring(0, 1).toUpperCase() + word.substring(1);  
 }  
 String clearWord = word.replaceAll("[^А**-**Яа**-**я]", "");  
 for (int k = 0; k < trans.length; k++) {  
 word = word.replaceAll(sleng[k], clearWord + trans[k]);  
 }  
 NEWmessage += word + " ";  
 }  
 arr.add(NEWmessage);  
 }  
 return arr;  
 }  
}

## 12.3. Тестирование работы программы с проверкой

Таблица 12.1

Тестирование работы программы и проверка результатов решения

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

# Задание 13

## 13.1. Постановка задачи

В первом файле хранится k матриц из n строк и n+1 столбцов каждая (последний столбец – столбец свободных членов). Во втором файле хранится k столбцов решения СЛАУ, записанных в файле 1. Проверить соответствие результатов (т.е. при умножении матрица коэффициентов на столбец решения, мы должны получить столбец свободных членов). Если решение не соответствует СЛАУ, тогда заменить его на правильное, иначе оставить без изменения (правильные данные записываются в третий файл).

## 13.2. Решение задачи, код программы

import java.io.IOException;  
import java.io.PrintWriter;  
import java.util.ArrayList;  
  
public class zad\_13 extends MyFiles {  
 public static void zad\_13() throws IOException {  
 String[] file1 = *ReadFile*("laba\_10\_zad\_13\_1.txt");  
 String[] file2 = *ReadFile*("laba\_10\_zad\_13\_2.txt");  
 String[][][] masses = *FileMas\_x3*(file1);  
 String[][][] argums = *FileMas\_x3*(file2);  
 *CreateNewFile*("D:\\Java\\laba-10","laba\_10\_zad\_13\_3.txt");  
 PrintWriter pw = new PrintWriter("laba\_10\_zad\_13\_3.txt");  
 String[] trueArgs;  
 for(int i = 0; i < masses.length; i++){  
 trueArgs = *Check*(masses[i],argums[i][0]);  
 for(int j = 0; j < trueArgs.length; j++){  
 pw.print(trueArgs[j]+"\t");  
 }  
 pw.println("\n");  
 }  
 pw.close();  
 }  
  
  
  
 public static String[][][] FileMas\_x3(String[] lines){  
 int i = 0;  
 ArrayList<ArrayList<String>> Arr = new ArrayList<>();  
 while(i < lines.length) {  
 ArrayList<String> arr = new ArrayList<>();  
 while (!lines[i].equals("")) {  
 arr.add(lines[i]);  
 i++;  
 if(i == lines.length){  
 break;  
 }  
 }  
 Arr.add(arr);  
 i++;  
 }  
 String[][][] Mas = new String[Arr.size()][Arr.get(0).size()][Arr.get(0).get(0).split("\\s+").length];  
 for(int k = 0; k < Mas.length;k++){  
 for(int j = 0; j < Mas[0].length;j++){  
 Mas[k][j] = Arr.get(k).get(j).split("\\s+");  
 }  
 }  
 return Mas;  
 }  
  
 public static String[] Gauss(String[][] mas){  
 int len = mas[0].length - 1;  
 double[][] copy = new double[len][len + 1];  
 for(int i = 0; i < len; i++) {  
 for (int j = 0; j < len + 1; j++){  
 copy[i][j] = Double.*parseDouble*(mas[i][j]);  
 }  
 }  
 for(int n = 0; n < len; n++) {  
 for (int i = n; i < len; i++) {  
 double num = copy[i][n];  
 for (int j = n; j < len + 1; j++) {  
 copy[i][j] = copy[i][j]/num;  
 if(Double.*isInfinite*(copy[i][j]) || Double.*isNaN*(copy[i][j]) ){  
 copy[i][j] = 0;  
 }  
 }  
 }  
 for (int i = n + 1; i < len; i++) {  
 for (int j = n; j < len + 1; j++) {  
 copy[i][j] -= copy[n][j];  
 }  
 }  
 }  
 double[] args = new double[len];  
 for(int i = 0; i < len; i++){  
 args[i] = 1;  
 }  
 args[len-1] = copy[len-1][len];  
 for (int i = len - 2; i >= 0; i--) {  
 double sum = 0;  
 for (int j = len - 1; j > i; j--) {  
 sum += copy[i][j] \* args[j];  
 }  
 args[i] = copy[i][len] - sum;  
 if(Double.*isInfinite*(args[i]) || Double.*isNaN*(args[i]) ){  
 args[i] = 0;  
 }  
 }  
 String[] strArgs = new String[len];  
 for(int i = 0; i < args.length;i++){  
 strArgs[i] = Double.*toString*(args[i]);  
 }  
 return strArgs;  
 }  
  
 public static String[] Check(String[][] mas, String[] args){  
 int len = mas[0].length - 1;  
  
 int count = 0;  
 for (int i = 0; i < len; i++) {  
 double sum = 0;  
 for (int j = 0; j < len; j++) {  
 sum += Double.*parseDouble*(mas[i][j]) \* Double.*parseDouble*(args[j]);  
 }  
 if ((sum > Double.*parseDouble*(mas[i][len]) - 0.001) &&  
 (sum < Double.*parseDouble*(mas[i][len]) + 0.001)){  
 count++;  
 }  
 }  
 if (count != len){  
 args = *Gauss*(mas);  
 }  
 return args;  
 }  
}

## 13.3. Тестирование работы программы с проверкой

Таблица 13.1

Тестирование работы программы и проверка результатов решения

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

# 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.IOException;  
import java.io.PrintWriter;  
  
public class zad\_14 extends MyFiles{  
 public static void zad\_14() throws IOException {  
 String[] file = *ReadFile*("Variant\_7,17.txt");  
 String[] nums = new String[file.length - 11];  
 *LinesSplit*(file, nums, 11);  
 String[][] mas = *MasNumsSplit*(nums);  
 String[] MaxMinMid = *MaxMinMid*(mas);  
 PrintWriter pw = new PrintWriter("Variant\_7,17.txt");  
 for(int i = 0; i < file.length; i++){  
 pw.println(file[i]);  
 }  
 pw.printf("\nОтветы на вопросы:\n");  
 pw.println("Вопрос №1");  
 pw.printf("Максимальное значение по UX: %s\n", MaxMinMid[0]);  
 pw.printf("Минимальное значение по UX: %s\n", MaxMinMid[1]);  
 pw.printf("Среднее значение по UX: %s\n\n", MaxMinMid[2]);  
 pw.println("Вопрос №2");  
 for(int i = 0; i < mas.length; i++){  
 pw.printf("Отклонение %d: %f\n",i+1,Double.*parseDouble*(MaxMinMid[2])-Double.*parseDouble*(mas[i][2]));  
 }  
 pw.close();  
 }  
 public static String[] MaxMinMid (String[][] mas){  
 String[] arr = new String[3];  
 double max, min, mid, sum;  
 max = Math.*pow*(10, -1000000000);  
 min = Math.*pow*(10, 1000000000);  
 sum = 0;  
 for (int i = 0; i < mas.length; i++){  
 if (Double.*parseDouble*(mas[i][2]) > max){  
 max = Double.*parseDouble*(mas[i][2]);  
 } else if (Double.*parseDouble*(mas[i][2]) < min){  
 min = Double.*parseDouble*(mas[i][2]);  
 }  
 sum += Double.*parseDouble*(mas[i][2]);  
 }  
 mid = sum/ mas.length;  
 arr[0] = Double.*toString*(max);  
 arr[1] = Double.*toString*(min);  
 arr[2] = Double.*toString*(mid);  
 return arr;  
 }  
  
 public static String[][] MasNumsSplit (String[] nums){  
 String[][] mas = new String[nums.length][nums[0].split("\\s+").length];  
 String[] line;  
 for (int i = 0; i < nums.length; i++){  
 line = nums[i].split("\\s+");  
 mas[i] = line;  
 }  
 return mas;  
 }  
  
  
 public static void LinesSplit (String[] file,String[] mas, int num){  
 for (int i = 0; i < mas.length; i++){  
 String line = file[num + i].replaceAll("-0\\."," -0\\.");  
 mas[i] = line;  
 }  
 }  
}

## 14.3. Тестирование работы программы с проверкой

Таблица 14.1

Тестирование работы программы и проверка результатов решения

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