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 группу ИСТ-22-1б Зверев А.А.

Проверил: Нетбай Георгий Владимирович

Пермь, 2023

**Содержание**

[Задание 1 4](#_Toc136179213)

[1.1. Постановка задачи 4](#_Toc136179214)

[1.2. Решение задачи, код программы 4](#_Toc136179215)

[1.3. Тестирование работы программы 4](#_Toc136179216)

[Задание 2 4](#_Toc136179217)

[2.1. Постановка задачи 4](#_Toc136179218)

[2.2. Решение задачи, код программы 4](#_Toc136179219)

[2.3. Тестирование работы программы 4](#_Toc136179220)

[Задание 3 5](#_Toc136179221)

[3.1. Постановка задачи 5](#_Toc136179222)

[3.2. Решение задачи, код программы 5](#_Toc136179223)

[3.3. Тестирование работы программы 5](#_Toc136179224)

[Задание 4 5](#_Toc136179225)

[4.1. Постановка задачи 5](#_Toc136179226)

[4.2. Решение задачи, код программы 5](#_Toc136179227)

[4.3. Тестирование работы программы 5](#_Toc136179228)

[Задание 5 5](#_Toc136179229)

[5.1. Постановка задачи 5](#_Toc136179230)

[5.2. Решение задачи, код программы 5](#_Toc136179231)

[5.3. Тестирование работы программы 5](#_Toc136179232)

[Задание 6 5](#_Toc136179233)

[6.1. Постановка задачи 5](#_Toc136179234)

[6.2. Решение задачи, код программы 6](#_Toc136179235)

[6.3. Тестирование работы программы 6](#_Toc136179236)

[Задание 7 6](#_Toc136179237)

[7.1. Постановка задачи 6](#_Toc136179238)

[7.2. Решение задачи, код программы 6](#_Toc136179239)

[7.3. Тестирование работы программы 6](#_Toc136179240)

[Задание 8 6](#_Toc136179241)

[8.1. Постановка задачи 6](#_Toc136179242)

[8.2. Решение задачи, код программы 6](#_Toc136179243)

[8.3. Тестирование работы программы 6](#_Toc136179244)

[Задание 9 7](#_Toc136179245)

[9.1. Постановка задачи 7](#_Toc136179246)

[9.2. Решение задачи, код программы 7](#_Toc136179247)

[9.3. Тестирование работы программы 7](#_Toc136179248)

[Задание 10 7](#_Toc136179249)

[10.1. Постановка задачи 7](#_Toc136179250)

[10.2. Решение задачи, код программы 7](#_Toc136179251)

[10.3. Тестирование работы программы 7](#_Toc136179252)

[Задание 11 7](#_Toc136179253)

[11.1. Постановка задачи 7](#_Toc136179254)

[11.2. Решение задачи, код программы 7](#_Toc136179255)

[11.3. Тестирование работы программы 8](#_Toc136179256)

[Задание 12 8](#_Toc136179257)

[12.1. Постановка задачи 8](#_Toc136179258)

[12.2. Решение задачи, код программы 8](#_Toc136179259)

[12.3. Тестирование работы программы 8](#_Toc136179260)

[Задание 13 8](#_Toc136179261)

[13.1. Постановка задачи 8](#_Toc136179262)

[13.2. Решение задачи, код программы 8](#_Toc136179263)

[13.3. Тестирование работы программы 8](#_Toc136179264)

[Задание 14 8](#_Toc136179265)

[14.1. Постановка задачи 8](#_Toc136179266)

[14.2. Решение задачи, код программы 8](#_Toc136179267)

[14.3. Тестирование работы программы 8](#_Toc136179268)

[Задание 15 9](#_Toc136179269)

[15.1. Постановка задачи 9](#_Toc136179270)

[15.2. Решение задачи, код программы 9](#_Toc136179271)

[15.3. Тестирование работы программы 9](#_Toc136179272)

# 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.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class Main {

public static void main(String[] args) throws IOException{

File file = new File("t1.txt");

FileWriter fwr = new FileWriter(file, true);

BufferedReader br = new BufferedReader(new FileReader(file));

fwr.write("\n");

String inStr = br.readLine();

char [] arr = inStr.toCharArray();

fwr.write("Количество латинских прописных букв: " + num3\_a(arr) + "\n");

fwr.write("Буква ю " + (num3\_b(arr) ? "входит" : "не входит") + " в последовательность\n");

fwr.write("В массиве " + (num3\_c(arr) ? "есть" : "нет") + " 2 и более запятых\n");

fwr.write("В массиве " + (num3\_d(arr) ? "есть" : "нет") + " сочетания символов во или ов\n");

fwr.write("В массиве " + (num3\_e(arr) ? "есть" : "нет") + " одинаковые подряд стоящие цифры\n");

fwr.write("Существуют " + (num3\_f(arr) ? "" : "не ") + "такие натуральные i и j, что 1 < i < j < n и s[i] и s[i+1] это одинаковые буквы отличающиеся регистром, а s[i], s[i+1] это '0'\n");

fwr.flush();

fwr.close();

}

public static int num3\_a(char[] arr) {

int Counter = 0;

for (char c : arr) {

if (Character.isUpperCase(c)) {

Counter++;

}

}

return Counter;

}

public static boolean num3\_b(char[] arr) {

boolean check = false;

for (char c : arr) {

if (Character.toLowerCase(c) == 'ю') {

check = true;

break;

}

}

return check;

}

public static boolean num3\_c(char[] arr) {

boolean check = false;

int Count = 0;

for (char c : arr) {

if (c == ',') {

Count++;

if (Count >= 2) {

check = true;

break;

}

}

}

return check;

}

public static boolean num3\_d(char[] arr) {

boolean check = false;

for (int i = 0; i < arr.length - 1; i++) {

if ((Character.toLowerCase(arr[i]) == 'в' && Character.toLowerCase(arr[i + 1]) == 'о') ||

(Character.toLowerCase(arr[i]) == 'о' && Character.toLowerCase(arr[i + 1]) == 'в')) {

check = true;

break;

}

}

return check;

}

public static boolean num3\_e(char[] arr) {

boolean check = false;

for (int i = 0; i < arr.length - 1; i++) {

if (Character.isDigit(arr[i]) && Character.isDigit(arr[i + 1]) && arr[i] == arr[i + 1]) {

check = true;

break;

}

}

return check;

}

public static boolean num3\_f(char[] arr) {

boolean check = false;

for (int i = 1; i < arr.length - 2; i++) {

if (Character.isLetter(arr[i]) && Character.isLetter(arr[i + 1]) && Character.toLowerCase(arr[i]) == Character.toLowerCase(arr[i + 1]) && arr[i] != arr[i + 1] && arr[i] == '0' && arr[i + 1] == '0') {

check = true;

break;

}

}

return check;

}

}

## 1.3. Тестирование работы программы

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 Main {

public static void main(String[] args) throws IOException {

File file1 = new File("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(String.valueOf(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

# Задание 3

## 3.1. Постановка задачи

Напечатать все слова из текстового файла, которые стоят перед знаком препинания.

## 3.2. Решение задачи, код программы

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Main {

public static void main(String[] args) throws IOException{

File file1 = new File("t3.txt");

String line;

Pattern pattern = Pattern.compile("(\\w+)\\p{P}");

BufferedReader br = new BufferedReader(new FileReader(file1));

while ((line = br.readLine()) != null) {

Matcher matcher = pattern.matcher(line);

while (matcher.find()) {

System.out.println(matcher.group(1));

}

}

}

}

## 3.3. Тестирование работы программы

# 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 строчных согласных русских букв, построчно записать в другой текстовый файл.

## 4.2. Решение задачи, код программы

import java.io.\*;

import java.util.HashSet;

import java.util.Set;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Main {

public static void main(String[] args) throws IOException{

File file1 = new File("t4.txt");

File file2 = new File("t4.1.txt");

String line;

Set<String> words = new HashSet<>();

Pattern pattern = Pattern.compile("[\_.,;:\\n\\t!?\\s]+");

Pattern vowelPattern = Pattern.compile("[аеёиоуыэюяaeiouy]+");

Pattern consonantPattern = Pattern.compile("[бвгджзйклмнпрстфхцчшщbcdfghjklmqrstvwxyz]+");

BufferedReader fread = new BufferedReader(new FileReader(file1));

BufferedWriter fwr = new BufferedWriter(new FileWriter(file2));

while ((line = fread.readLine()) != null) {

String[] tokens = pattern.split(line);

for (String token : tokens) {

if (token.matches("[а-яёAEIOUYaeiouy]+")) {

Matcher vowelMatcher = vowelPattern.matcher(token);

Matcher consonantMatcher = consonantPattern.matcher(token);

if (vowelMatcher.find() && consonantMatcher.find()) {

words.add(token.toLowerCase());

}

}

}

}

for (String word : words) {

fwr.write(word);

fwr.newLine();

}

fwr.flush();

fwr.close();

}

}

## 4.3. Тестирование работы программы

# Задание 5

## 5.1. Постановка задачи

Дано два файла с предложениями. Дописать в конец второго файла предложения из первого файла, в которых есть не менее двух запятых.

## 5.2. Решение задачи, код программы

import java.io.\*;

public class Main {

public static void main(String[] args) throws IOException{

File file1 = new File("t5.txt");

File fileout = new File("t5.2.txt");

String line;

BufferedReader fread1 = new BufferedReader(new FileReader(file1));

BufferedWriter fwr = new BufferedWriter(new FileWriter(fileout, true));

while ((line = fread1.readLine()) != null) {

if (line.split(",").length >= 2) {

fwr.write(line);

fwr.newLine();

}

}

fwr.flush();

fwr.close();

}

}

## 5.3. Тестирование работы программы

# Задание 6

## 6.1. Постановка задачи

Входной файл содержит сведения о кондитерских изделиях: название изделия, название кондитерской, граммы, цена. Подсчитать для каждой кондитерской количество кондитерских изделий массой от 150 до 250 грамм и стоимостью от 90 до 199 руб., дописать ответ в конец файла построчно.

## 6.2. Решение задачи, код программы

iimport java.io.\*;

import java.util.Arrays;

public class Main {

public static void main(String[] args) throws IOException{

File file1 = new File("t6.txt");

File fileout = new File("t6.1.txt");

String line;

BufferedReader fread = new BufferedReader(new FileReader(file1));

BufferedWriter fwr = new BufferedWriter(new FileWriter(fileout, true));

String current = "";

int cnt = 0;

while ((line = fread.readLine()) != null) {

String[] parts = line.split(",");

String confectionery = parts[1];

int grams = Integer.parseInt(parts[2]);

int price = Integer.parseInt(parts[3]);

if (grams >= 150 && grams <= 250 && price >= 90 && price <= 199) {

if (!confectionery.equals(current)) {

if (!current.equals("")) {

fwr.write(current + ": " + cnt);

fwr.newLine();

}

current = confectionery;

cnt = 0;

}

cnt++;

}

}

if (!current.equals("")) {

fwr.write(current + ": " + cnt);

fwr.newLine();

}

fwr.flush();

fwr.close();

}

}

## 6.3. Тестирование работы программы

# 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.File;

import java.io.FileWriter;

import java.io.IOException;

import java.text.DecimalFormat;

import static java.lang.Math.\*;

public class Main {

public static void main(String[] args) throws IOException {

File file1 = new File("t7.txt");

FileWriter fwr = new FileWriter(file1);

fwr.write("Постановка задачи\n\nx >= -3 && x <= 3; x += 0.5\nесли x <= -1, то f(x) = cos(x) - sin(x)\nиначе f(x) = tan(x)\n\nВЫВОД\n\n"

+"Функция\t\s\s\s\sЗначение f(x)\n\n");

DecimalFormat d = new DecimalFormat("#.#");

double x = -3;

while (x <= 3){

if (x < 0) fwr.write( "f(" + x + ")\t=\t" + d.format(cos(PI\*x)) + "\n");

else fwr.write("f(" + x + ")\t=\t" + d.format(x\*x) + "\n");

x += 0.5;

}

fwr.flush();

fwr.close();

}

}

## 7.3. Тестирование работы программы

# 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

i

## 8.3. Тестирование работы программы

# 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

mport java.io.\*;

import java.util.Arrays;

public class Main {

public static void main(String[] args) throws IOException {

int count = 1;

File file = new File("t9.txt");

FileWriter fwr = new FileWriter(file, true);

BufferedReader fread = new BufferedReader(new FileReader(file));

String[][] publishers = new String[count][18];

for(int j = 0; j < 1; j++){

while (fread.ready()) {

String line = fread.readLine();

String[] mas = line.split(";");

publishers[j][0] = mas[0];

publishers[j][1] = mas[1];

publishers[j][2] = mas[2];

publishers[j][3] = mas[3];

publishers[j][4] = mas[4];

publishers[j][5] = mas[5];

publishers[j][6] = mas[6];

publishers[j][7] = mas[7];

publishers[j][8] = mas[8];

publishers[j][9] = mas[9];

publishers[j][10] = mas[10];

publishers[j][11] = mas[11];

publishers[j][12] = mas[12];

publishers[j][13] = mas[13];

publishers[j][14] = mas[14];

publishers[j][15] = mas[15];

publishers[j][16] = mas[16];

publishers[j][17] = mas[17];

}

int countHighRate = 0;

int countBestseilers = 0;

int countAthrs = 0;

int countTrsl = 0;

int countPhilosophy = 0;

int countAdv = 0;

for (int i = 0; i < count; i++) {

String city = publishers[i][1];

double rating = Double.parseDouble(publishers[i][3]);

int bestsellers = Integer.parseInt(publishers[i][10]);

int staff = Integer.parseInt(publishers[i][5]);

int authors = Integer.parseInt(publishers[i][8]);

double translationEducationPercentage = (Double.parseDouble(publishers[i][6]) / staff) \* 100;

double philologyEducationPercentage = (Double.parseDouble(publishers[i][7]) / staff) \* 100;

double advPercentage = (Double.parseDouble(publishers[i][13]) / Double.parseDouble(publishers[i][11])) \* 100;

if (rating > 2.5) {

countHighRate++;

}

if (bestsellers >= 5) {

countBestseilers++;

}

if (authors > 3 \* staff) {

countAthrs++;

}

if (translationEducationPercentage >= 20) {

countTrsl++;

}

if (philologyEducationPercentage >= 30) {

countPhilosophy++;

}

if (advPercentage < 20) {

countAdv++;

}

}

fwr.write("\nКоличество издательств по городам с рейтингом больше 2.5: " + countHighRate + "\n");

fwr.write("Количество издательств по городам с не менее 5 бестселлерами: " + countBestseilers+ "\n");

fwr.write("Количество издательств, у которых количество авторов в 3 и более раза превышает количество сотрудников: " + countAthrs+ "\n");

fwr.write("Количество издательств по странам, у которых 20% сотрудников имеют профессиональное образование в области перевода: " + countTrsl+ "\n");

fwr.write("Количество издательств по странам, у которых не менее 30% сотрудников имеют филологическое образование: " + countPhilosophy+ "\n");

fwr.write("Количество издательств по странам, у которых расходы на рекламу составляют менее 20% от среднего дохода: " + countAdv+ "\n");

fwr.flush();

fwr.close();

}

}

}

## 9.3. Тестирование работы программы

# 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;

public class Main {

public static void main(String[] args) throws IOException {

File file1 = new File("t10.txt");

FileWriter fwr = new FileWriter(file1, true);

Scanner in = new Scanner(System.in);

double x = 0;

System.out.println("Введите значение x");

while (Math.abs(x = in.nextDouble()) >= 2.4){

System.out.println("Ошибка! Введите другое значение x");

}

int k = 0;

double s = 0;

for (double ep = 0.0001; ep <= 0.01; ep\*=10){

s = 0;

k = 0;

while (Math.abs(Math.exp(x)\*(1+x)-s) > ep){

s += (Math.pow(x,k)\*(k + 1))/(factorial1(k));

k++;

}

fwr.write("Функция f(x) = e^x(1 + x) для x = " + x + " равняется " + s + "\n"

+"Результаты определения значений функции f(x) = e^x(1 + x) с помощью ряда Маклорена\n"

+"Погрешность итерационной процедуры "+ep+"\n"

+"Значение функции по Маклорену\t\tПогрешность, %\tЧисло итераций\n"

+"\t"+s+"\t\t\t\t"+ep\*100+"\t\t\t"+k+"\n\n");

}

fwr.flush();

fwr.close();

}

public static int factorial1(int enter1){

int res1 = 1;

for (int i = 1; i <= enter1; i ++){

res1 \*= i;

}

return res1;

}

}

}

## 10.3. Тестирование работы программы

# 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

i

## 11.3. Тестирование работы программы

# 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.\*;

public class Main {

public static void main(String[] args) throws IOException {

File file1 = new File("t12.txt");

BufferedReader fread = new BufferedReader(new FileReader(file1));

FileWriter fwr = new FileWriter(file1, true);

String text = fread.readLine();

String[] strings = text.split(";");

String[] searchWords = {

"учащийся",

"обучающийся",

"ученик",

"студент",

"бакалавр",

"магистр",

"2020"

};

String[] replacements = {

"студент",

"отчисленный",

"специалист",

"выпускник",

"магистр",

"аспирант",

"2021"

};

for (int i = 0; i < strings.length; i++) {

String originalString = strings[i];

String modifiedString = originalString;

for (int j = 0; j < searchWords.length; j++) {

modifiedString = modifiedString.replaceAll(searchWords[j], replacements[j]);

}

fwr.write("\nИзмененная строка: " + modifiedString + "\n");

}

fwr.flush();

fwr.close();

}

}

## 12.3. Тестирование работы программы

# 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.BufferedWriter;

import java.io.FileWriter;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Paths;

import java.util.ArrayList;

import java.util.List;

public class Main {

public static void main(String[] args) throws IOException {

String out = "t133.txt";

List<double[][]> f1 = readM("t131.txt");

List<double[]> f2 = readSLAU("t132.txt");

List<double[]> corrected = new ArrayList<>();

for (int i = 0; i < f1.size(); i++) {

double[][] array = f1.get(i);

double[] array1 = f2.get(i);

double[] array2 = check(array, array1);

corrected.add(array2);

}

write(out, corrected);

}

private static List<double[][]> readM(String file) throws IOException {

List<double[][]> array1 = new ArrayList<>();

List<String> line = Files.readAllLines(Paths.get(file));

int n = Integer.parseInt(line.get(0));

line.remove(0);

while (!line.isEmpty()) {

double[][] matrix = new double[n][n + 1];

for (int i = 0; i < n; i++) {

String[] value = line.get(0).split(" ");

for (int j = 0; j < n + 1; j++) {

matrix[i][j] = Double.parseDouble(value[j]);

}

line.remove(0);

}

array1.add(matrix);

}

return array1;

}

private static List<double[]> readSLAU(String file) throws IOException {

List<double[]> arrayS = new ArrayList<>();

List<String> line = Files.readAllLines(Paths.get(file));

while (!line.isEmpty()) {

String[] values = line.get(0).split(" ");

double[] s = new double[values.length];

for (int i = 0; i < values.length; i++) {

s[i] = Double.parseDouble(values[i]);

}

arrayS.add(s);

line.remove(0);

}

return arrayS;

}

private static double[] check(double[][] array, double[] array1) {

int n = array.length;

double[] res = new double[n];

for (int i = 0; i < n; i++) {

double sum = 0;

for (int j = 0; j < n; j++) {

sum += array[i][j] \* array1[j];

}

res[i] = sum;

}

boolean f = true;

for (int i = 0; i < n; i++) {

if (Math.abs(res[i] - array[i][n]) > 1e-6) {

f = false;

break;

}

}

return f ? array1 : res;

}

private static void write(String file, List<double[]> resh) throws IOException {

try (BufferedWriter fwr = new BufferedWriter(new FileWriter(file))) {

for (double[] array1 : resh) {

for (int i = 0; i < array1.length; i++) {

fwr.write(Double.toString(array1[i]));

if (i < array1.length - 1) {

fwr.write(" ");

}

}

fwr.newLine();

}

}

}

}

## 13.3. Тестирование работы программы

# 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.nio.file.\*;

import java.util.\*;

public class Main {

public static void main(String[] args) throws IOException{

List<Double> list = new ArrayList<>();

BufferedReader fread = Files.newBufferedReader(Paths.get("Variant\_10,20.txt"));

String str;

while ((str = fread.readLine()) != null) {

if (str.trim().length() == 0) continue;

String[] a = str.trim().split("\\s+");

if (a.length >= 5 && yep(a[4])) {

list.add(Double.parseDouble(a[4]));

}

}

double min = Collections.min(list);

double max = Collections.max(list);

double sum = list.stream().mapToDouble(Double::doubleValue).sum();

double avg = sum / list.size();

BufferedWriter fwr = Files.newBufferedWriter(Paths.get("Variant\_10,20.txt"), StandardOpenOption.APPEND);

fwr.newLine();

fwr.write("Min: " + min + ", Max: " + max + ", Avg: " + avg);

fwr.newLine();

for (double value : list) {

double dev = value - avg;

fwr.write("Deviation: " + dev);

fwr.newLine();

}

fwr.flush();

fwr.close();

}

private static boolean yep(String str) {

if (str == null || !Character.isDigit(str.charAt(0)))

return false;

Double.parseDouble(str);

return true;

}

}

## 14.3. Тестирование работы программы

# 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

## 15.3. Тестирование работы программы