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 «Символы и строки»

по дисциплине «Информатика»

Выполнил: студент группы ИСТ-22-1б Васин М.А.

Проверил: ассистент каф. ВММБ Нетбай Г.В.

Пермь, 2023

**Содержание**

[Задание 1 4](#_Toc136807343)

[1.1. Постановка задачи 4](#_Toc136807344)

[1.2. Решение задачи, код программы 4](#_Toc136807345)

[1.3. Тестирование работы программы 5](#_Toc136807346)

[Задание 2 5](#_Toc136807347)

[2.1. Постановка задачи 5](#_Toc136807348)

[2.2. Решение задачи, код программы 5](#_Toc136807349)

[2.3. Тестирование работы программы 6](#_Toc136807350)

[Задание 3 6](#_Toc136807351)

[3.1. Постановка задачи 6](#_Toc136807352)

[3.2 Решение задачи, код программы 6](#_Toc136807353)

[3.3. Тестирование работы программы 7](#_Toc136807354)

[Задание 4 7](#_Toc136807355)

[4.1. Постановка задачи 7](#_Toc136807356)

[4.2. Решение задачи, код программы 7](#_Toc136807357)

[4.3. Тестирование работы программы 9](#_Toc136807358)

[Задание 5 9](#_Toc136807359)

[5.1. Постановка задачи 9](#_Toc136807360)

[5.2. Решение задачи, код программы 9](#_Toc136807361)

[5.3. Тестирование работы программы 10](#_Toc136807362)

[Задание 6 10](#_Toc136807363)

[6.1. Постановка задачи 10](#_Toc136807364)

[6.2. Решение задачи, код программы 10](#_Toc136807365)

[6.3. Тестирование работы программы 12](#_Toc136807366)

[Задание 7 12](#_Toc136807367)

[7.1. Постановка задачи 12](#_Toc136807368)

[7.2.1 Решение задачи, код программы 12](#_Toc136807369)

[7.3. Тестирование работы программы 13](#_Toc136807370)

[Задание 8 13](#_Toc136807371)

[8.1. Постановка задачи 13](#_Toc136807372)

[8.2. Решение задачи, код программы 13](#_Toc136807373)

[8.3. Тестирование работы программы 14](#_Toc136807374)

[Задание 9 14](#_Toc136807375)

[9.1. Постановка задачи 14](#_Toc136807376)

[9.2. Решение задачи, код программы 14](#_Toc136807377)

[9.3. Тестирование работы программы 14](#_Toc136807378)

[Задание 10 14](#_Toc136807379)

[10.1. Постановка задачи 14](#_Toc136807380)

[10.2. Решение задачи, код программы 14](#_Toc136807381)

[10.3. Тестирование работы программы 15](#_Toc136807382)

[Задание 11 15](#_Toc136807383)

[11.1. Постановка задачи 15](#_Toc136807384)

[11.2. Решение задачи, код программы 15](#_Toc136807385)

[11.3. Тестирование работы программы 16](#_Toc136807386)

[Задание 12 16](#_Toc136807387)

[12.1. Постановка задачи 16](#_Toc136807388)

[12.2. Решение задачи, код программы 16](#_Toc136807389)

[12.3. Тестирование работы программы 17](#_Toc136807390)

[Задание 13 17](#_Toc136807391)

[13.1. Постановка задачи 17](#_Toc136807392)

[13.2. Решение задачи, код программы 17](#_Toc136807393)

[13.3. Тестирование работы программы 18](#_Toc136807394)

[Задание 14 18](#_Toc136807395)

[14.1. Постановка задачи 18](#_Toc136807396)

[14.2. Решение задачи, код программы 18](#_Toc136807397)

[14.3. Тестирование работы программы 19](#_Toc136807398)

# Задание 1

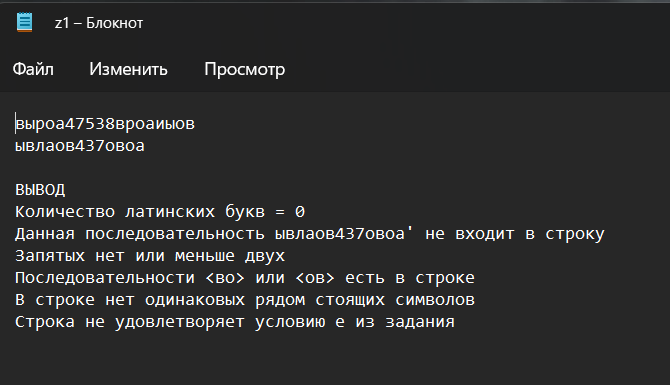
## 1.1. Постановка задачи

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 z1 {  
 public static int A(String input){  
 int lat = 0;  
 for (int i = 0; i < input.length(); i++) {  
 char c = input.charAt(i);  
 if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z')) {  
 lat+=1;  
 }  
 }  
 return lat;  
 }  
 public static Boolean B(String stroka, String posled){  
 if (stroka.contains(posled)){  
 return true;  
 }  
 return false;  
 }  
 public static Boolean C(String stroka){  
 int Kol\_Zap = 0;  
 for (int i = 0; i < stroka.length(); i++) {  
 char c = stroka.charAt(i);  
 if (stroka.contains(",")) {  
 Kol\_Zap+=1;  
 }  
 }  
 if ( Kol\_Zap>=2){  
 return true;  
 }  
 return false;  
 }  
 public static Boolean D(String stroka){  
 if (stroka.contains("во") | stroka.contains("ов") ){  
 return true;  
 }  
 return false;  
 }  
 public static Boolean E(String input){  
  
 for (int i = 0; i < input.length()-3; i++) {  
 char a = input.charAt(i+2);  
 char b = input.charAt(i+1);  
 char c = input.charAt(i);  
 if ((b == a) || (c == b)) {  
 return true;  
 }  
 }  
 return false;  
 }  
 public static boolean f(String stroka){  
 for( int i = 2; i<stroka.length()-2; i++){  
 char a = stroka.charAt(i);  
 char b = stroka.charAt(i+1);  
 if((Character.*isLowerCase*(a) && Character.*isUpperCase*(b))||(Character.*isLowerCase*(b) && Character.*isUpperCase*(a))){  
 for(int j=i+1; j< stroka.length()-1; j++){  
 char c = stroka.charAt(j);  
 char d = stroka.charAt(j+1);  
 if((c=='0' && d=='0')){  
 return true;  
 }  
 }  
 }  
 }  
 return false;  
 }  
 public static void main(String[] args) throws IOException {  
 File file= new File("C:\\Users\\Мишка\\Desktop\\z1.txt");  
 BufferedReader fread = new BufferedReader(new FileReader(file));  
 FileWriter fwrite = new FileWriter(file,true);  
 String str = fread.readLine();  
 String str1 = fread.readLine();  
 fwrite.write("\n\nВЫВОД\nКоличество латинских букв = " + *A*(str)+"\n");  
 if (*B*(str,str1)){  
 fwrite.write("Данная последовательность " + str1 + "' входит в строку\n");  
 }  
 else {  
 fwrite.write("Данная последовательность " + str1 + "' не входит в строку\n");  
 }  
 if (*C*(str)){  
 fwrite.write("Колличество запятых в строке две и более\n");  
 }  
 else {  
 fwrite.write("Запятых нет или меньше двух\n");  
 }  
 if (*D*(str)){  
 fwrite.write("Последовательности <во> или <ов> есть в строке\n");  
 }  
 else {  
 fwrite.write("Последовательностей <во> или <ов> нет в строке\n");  
 }  
 if (*E*(str)){  
 fwrite.write("В строке есть одинаковые рядом стоящие символы\n");  
 }  
 else {  
 fwrite.write("В строке нет одинаковых рядом стоящих символов\n");  
 }  
 if(*f*(str)){  
 fwrite.write("Строка удовлетворяет условию е из задания\n");  
 }  
 else {  
 fwrite.write("Строка не удовлетворяет условию е из задания\n");  
 }  
 fwrite.close(); fread.close();  
 }  
  
}

## 1.3. Тестирование работы программы



# Задание 2

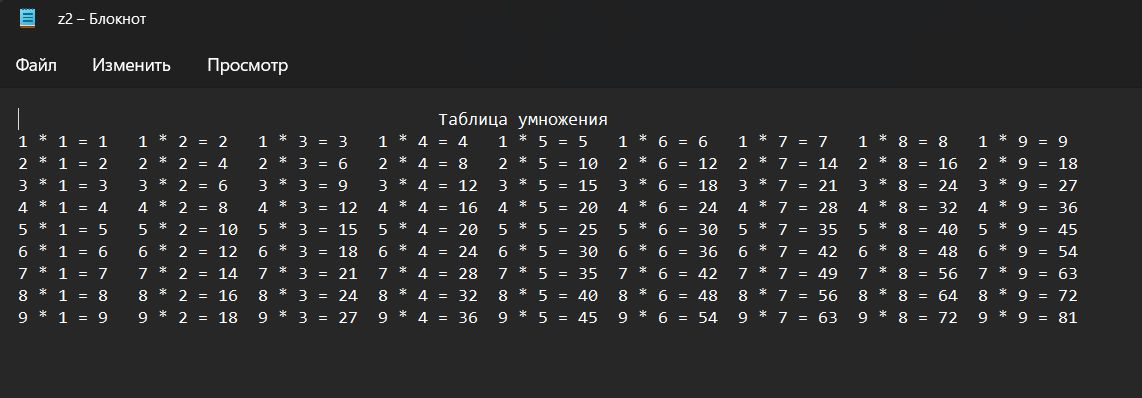
## 2.1. Постановка задачи

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 z2 {  
 public static void main(String[] args) throws IOException {  
 File file = new File("C:\\Users\\Мишка\\Desktop\\z2.txt");  
 FileWriter fwrite = new FileWriter(file);  
 fwrite.write("\t\t\t\t\t\t\tТаблица умножения\n");  
 int s=0;  
 for (int i = 1; i < 10; i++) {  
 for (int j = 1; j < 10; j++) {  
 s = i \* j;  
 fwrite.write(i + " \* " + j + " = " + s + "\t");  
 }  
 fwrite.write("\n");  
 }  
 fwrite.close();  
 }  
}

## 2.3. Тестирование работы программы



# Задание 3

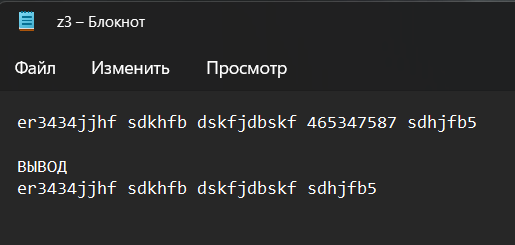
## 3.1. Постановка задачи

Напечатать все слова из текстового файла, состоящие только из маленьких латинских букв

## 3.2 Решение задачи, код программы

import java.io.\*;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
public class z3 {  
 public static void main(String[] args) throws IOException{  
 File file = new File("C:\\Users\\Мишка\\Desktop\\z1.txt");  
 BufferedReader fread = new BufferedReader(new FileReader(file));  
 FileWriter fwrite = new FileWriter(file,true);  
 Pattern pattern = Pattern.*compile*("[a-z]+");  
 String [] line = fread.readLine().split("\\s+");  
 fwrite.write("\n\nВЫВОД\n");  
 for (int i=0;i<line.length;i++){  
 Matcher mat = pattern.matcher(line[i]);  
 if (mat.find()){  
 fwrite.write(line[i]+"\s");  
 }  
 }  
 fwrite.close();  
 }  
}

## 3.3. Тестирование работы программы



# Задание 4

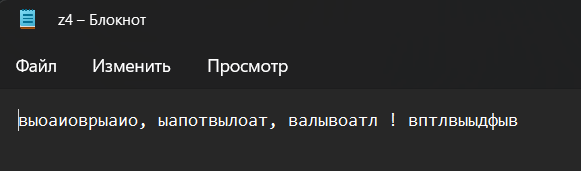
## 4.1. Постановка задачи

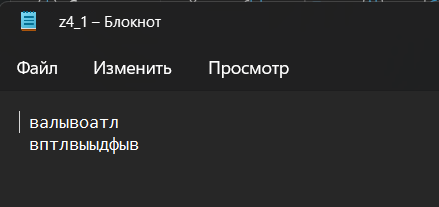
Выделить в текстовом файле все слова, разделенные символами-разделителями «\_.,;:\n\t!?», и все слова, в которых гласных русских букв меньше чем согласных букв, построчно записать в другой текстовый файл.

## 4.2. Решение задачи, код программы

import java.util.ArrayList;  
import java.io.\*;  
public class z4 {  
 public static void main(String[] args) throws IOException {  
 File inputFile = new File("C:\\Users\\Мишка\\Desktop\\z4.txt");  
 ArrayList<String> words = new ArrayList<>();  
 try (BufferedReader reader = new BufferedReader(new FileReader(inputFile))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] lineWords = line.split("[\_.,;:\\n\\t!?]");  
 for (String word : lineWords) {  
 if (*isBalanced*(word)) {  
 words.add(word);  
 }  
 }  
 }  
 }  
 File outputFile = new File("C:\\Users\\Мишка\\Desktop\\z4\_1.txt");  
 try (PrintWriter writer = new PrintWriter(outputFile)) {  
 for (String word : words) {  
 writer.println(word);  
 }  
 }  
 }  
 public static boolean isBalanced(String word) {  
 int numglas = 0;  
 int numsoglas = 0;  
 String glasnie = "аеёиоуыэюяАЕЁИОУЫЭЮЯ";  
 String soglasnie = "бвгджзйклмнпрстфхцчшщъьБВГДЖЗЙКЛМНПРСТФХЦЧШЩЪЬ";  
 for (char c : word.toCharArray()) {  
 if (glasnie.indexOf(c) != -1) {  
 numglas++;  
 } else if (soglasnie.indexOf(c) != -1) {  
 numsoglas++;  
 }  
 }  
  
 return numglas < numsoglas;  
 }  
}

## 4.3. Тестирование работы программы





# Задание 5

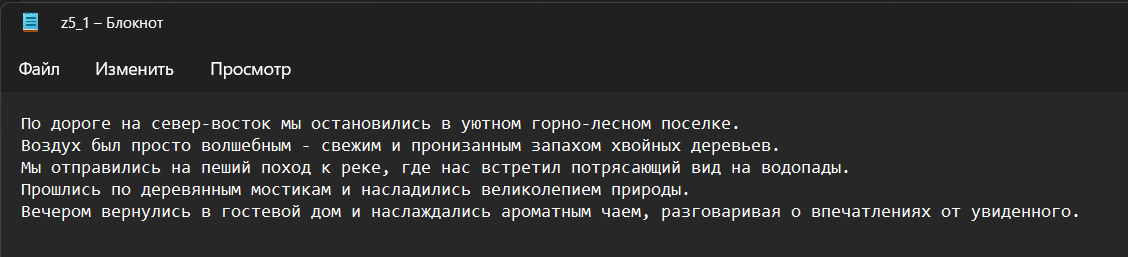
## 5.1. Постановка задачи

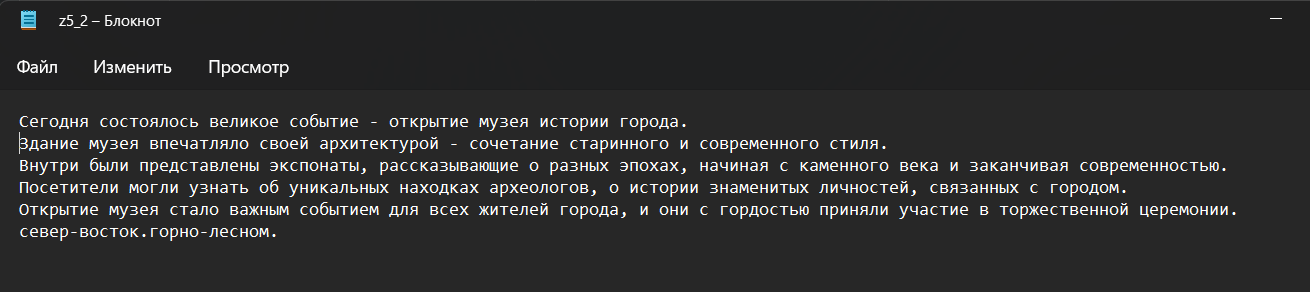
Дано два файла с предложениями. Дописать в конец второго файла предложения из первого файла, в которых есть слова, которые пишутся через дефис.

## 5.2. Решение задачи, код программы

import java.io.\*;  
public class z5 {  
 public static void main(String[] args) throws IOException {  
 File file = new File("C:\\Users\\Мишка\\Desktop\\z5\_1.txt");  
 File file2 = new File("C:\\Users\\Мишка\\Desktop\\z5\_2.txt");  
 BufferedReader fread = new BufferedReader(new FileReader(file));  
 FileWriter fwrite = new FileWriter(file2, true);  
 StringBuilder line = new StringBuilder();  
 while (fread.ready()) {  
 line.append(fread.readLine());  
 }  
 String[] predlozhenie = String.*valueOf*(line).split("\\.");  
 for (int i = 0; i < predlozhenie.length; i++) {  
 String pred1 = predlozhenie[i];  
 String[] slovo = predlozhenie[i].split("\\s+");  
 for (int j = 0; j < slovo.length; j++) {  
 if (slovo[j].length() >= 2) {  
 if (slovo[j].contains("-")) {  
 fwrite.write(slovo[j] + ".");  
 }  
 }  
 }  
 }  
 fwrite.close();  
 }  
}

## 5.3. Тестирование работы программы





# Задание 6

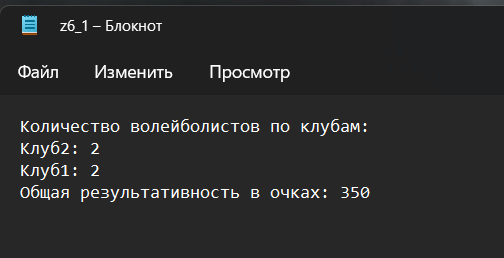
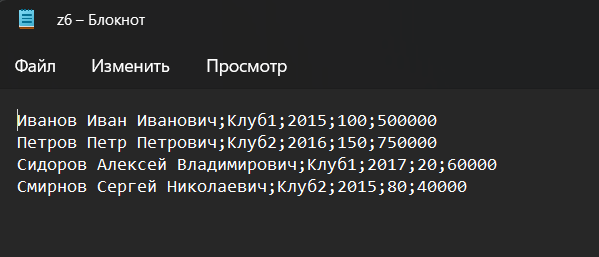
## 6.1. Постановка задачи

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 и их общую результативность в очках, дописать ответы построчно 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;  
import java.util.HashMap;  
import java.util.Map;  
public class z6 {  
 public static void main(String[] args) {  
 String inputFile = "C:\\Users\\Мишка\\Desktop\\z6.txt";  
 String outputFile = "C:\\Users\\Мишка\\Desktop\\z6\_1.txt";  
 try {  
 BufferedReader reader = new BufferedReader(new FileReader(inputFile));  
 BufferedWriter writer = new BufferedWriter(new FileWriter(outputFile, true));  
 String line;  
 Map<String, Integer> clubCountMap = new HashMap<>();  
 int totalPoints = 0;  
 while ((line = reader.readLine()) != null) {  
 String[] data = line.split(";");  
 if (data.length >= 2) {  
 String club = data[1];  
 int points = Integer.*parseInt*(data[3]);  
 clubCountMap.put(club, clubCountMap.getOrDefault(club, 0) + 1);  
 totalPoints += points;  
 }  
 }  
 reader.close();  
 writer.newLine();  
 writer.write("Количество волейболистов по клубам:");  
 writer.newLine();  
 for (Map.Entry<String, Integer> entry : clubCountMap.entrySet()) {  
 String club = entry.getKey();  
 int count = entry.getValue();  
 writer.write(club + ": " + count);  
 writer.newLine();  
 }  
 writer.write("Общая результативность в очках: " + totalPoints);  
 writer.newLine();  
 writer.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

## 6.3. Тестирование работы программы



# Задание 7

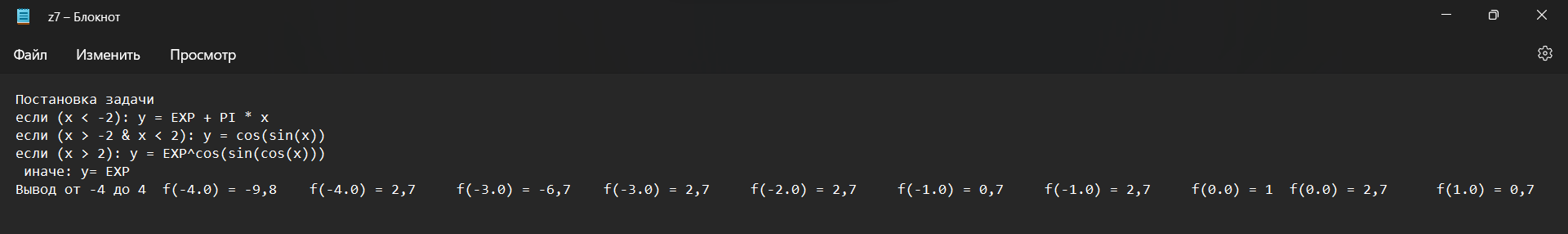
## 7.1. Постановка задачи

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.

## 7.2.1 Решение задачи, код программы

import java.io.\*;  
import java.text.DecimalFormat;  
public class z7 {  
 public static void main(String[] args) throws IOException {  
 File file =new File("C:\\Users\\Мишка\\Desktop\\z7.txt");  
 FileWriter fwrite = new FileWriter(file);  
 DecimalFormat decimalFormat =new DecimalFormat("#.#");  
 fwrite.write("Постановка задачи\n" +  
 "если (x < -2): y = EXP + PI \* x\n" +  
 "если (x > -2 & x < 2): y = cos(sin(x))\n" +  
 "если (x > 2): y = EXP^cos(sin(cos(x)))\n" +  
 " иначе: y= EXP\n");  
 fwrite.write("Вывод от -4 до 4\t");  
 for (double x = -4;x<=4;x+=1){  
 if (x < -2) fwrite.write("f("+x+") = "+decimalFormat.format(*Q*(x))+"\t");  
 if (x > -2 & x < 2) fwrite.write("f("+x+") = "+decimalFormat.format(*R*(x))+"\t");  
 if (x > 2) fwrite.write("f("+x+") = "+decimalFormat.format(*T*(x))+"\t");  
 else fwrite.write("f("+x+") = "+decimalFormat.format(Math.*E*)+"\t");  
 }  
 fwrite.close();  
 }  
 public static double Q(double x) {  
 double a1;  
 a1 = Math.*E* + Math.*PI* \* x;  
 return a1;  
 }  
 public static double R(double x) {  
 double a2;  
 a2 = Math.*cos*(Math.*sin*(x));  
 return a2;  
 }  
 public static double T(double x) {  
 double a3;  
 a3 = Math.*pow*(Math.*E*, Math.*cos*(Math.*sin*(Math.*cos*(x))));  
 return a3;  
 }  
}

## 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.

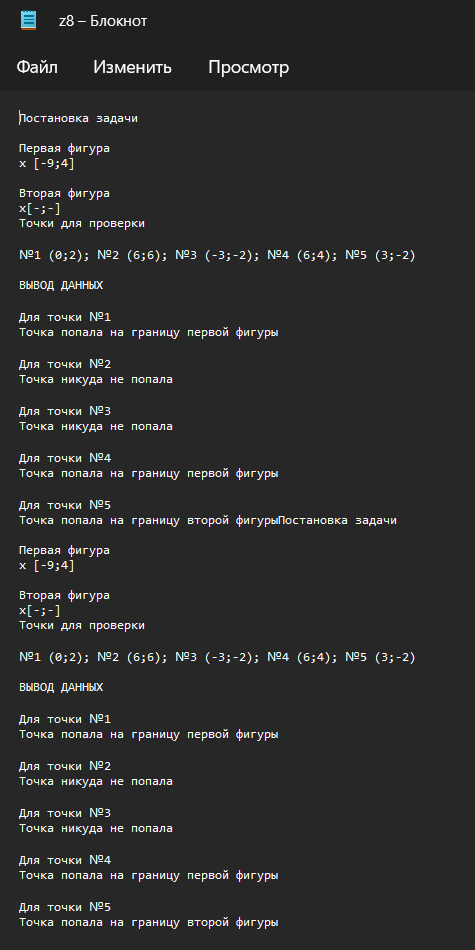
Массив точек для проверки: номер точки, координата x; координата y.

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

import java.io.\*;  
import static java.lang.Math.\*;  
public class z8 {  
 public static String hit(double x, double y) {  
 if (((x > -5) && (x < -4) && (y < x + 4) && (y > -x - 6)) ||  
 ((x > -4) && (x < -3) && (y < x + 4) && (y > 2 \* x + 6)) ||  
 ((x > -3) && (x < -2) && (y < *sqrt*(4 - *pow*(x + 1, 2)) + 1) && (y > -2 \* x - 6)) ||  
 ((x > -2) && (x < -1) && (y < *sqrt*(4 - *pow*(x + 1, 2)) + 1) && (y > x)) ||  
 ((x > -1) && (x < 0) && ((y < 2.0 / 3 \* x + 11.0 / 3) && (y > 5.0 / 3 \* x + 5.0 / 3) ||  
 (y < x + 1 && (*pow*(x + 1, 2) + *pow*(y - 1, 2) < 4))))) {  
 return "Точка попала в область первой фигуры\n";  
 }  
 else if (((x > 0) && (x < 1) && (y < 0.5 \* x - 5) && (y > 0.25 \* x - 5)) ||  
 ((x > 1) && (x < 2) && (y < 0.5 \* x - 5) && (y > 0.25 \* x - 5)) ||  
 ((x > 2) && (x < 3) && ((y < -x + 1) && (Math.*pow*(x - 4, 2) + Math.*pow*(y + 1, 2) <= 4) || (y < 0.5 \* x - 5) && (y > 0.25 \* x - 5))) ||  
 ((x > 3) && (x < 4) && ((y < x - 5) && (Math.*pow*(x - 4, 2) + Math.*pow*(y + 1, 2) <= 4) || (y < 0.5 \* x - 5) && (y > 0.25 \* x - 5))) ||  
 ((x > 4) && (x < 5) && ((Math.*pow*(x - 4, 2) + Math.*pow*(y + 1, 2) < 4) && (y > -x + 5) || (y < 0 && y >= -2) || (y < -3 \* x + 10) && (y > -x))) ||  
 ((x > 5) && (x < 6) && (Math.*pow*(x - 4, 2) + Math.*pow*(y + 1, 2) < 4) && (y > -2)) ||  
 ((x > 6) && (x < 7) && (y < -x + 5) && (y > -2))) {  
 return "Точка попала в область второй фигуры\n";  
 }  
 else if (((x >= -5) && (x <= -4) && (y <= x + 4) && (y >= -x - 6)) ||  
 ((x >= -4) && (x <= -3) && (y <= x + 4) && (y >= 2 \* x + 6)) ||  
 ((x >= -3) && (x <= -2) && (y <= Math.*sqrt*(4 - Math.*pow*(x + 1, 2)) + 1) && (y >= -2 \* x - 6)) ||  
 ((x >= -2) && (x <= -1) && (y <= Math.*sqrt*(4 - Math.*pow*(x + 1, 2)) + 1) && (y >= x)) ||  
 ((x >= -1) && (x <= 0) && ((y <= 2.0 / 3 \* x + 11.0 / 3) && (y >= 5.0 / 3 \* x + 5.0 / 3) ||  
 (y <= x + 1 && (Math.*pow*(x + 1, 2) + Math.*pow*(y - 1, 2) <= 4)))) ||  
 ((x >= 1) && (x <= 2) && ((y <= 2.0 / 3 \* x + 11.0 / 3) && (y >= 5.0 / 3 \* x + 5.0 / 3) ||  
 (y <= x + 1 && y >= 1.0 / 3 \* x + 2.0 / 3))) ||  
 ((x >= 2) && (x <= 3) && (y <= x + 1 && y >= 1.0 / 3 \* x + 2.0 / 3)) ||  
 ((x >= 3) && (x <= 4) && (y <= x + 1 && y >= 1.0 / 3 \* x + 2.0 / 3)) ||  
 ((x >= 4) && (x <= 5) && (y <= -1.0 / 2 \* x + 7 && y >= x - 2)) ||  
 ((x >= 5) && (x <= 6) && (y <= -1.0 / 2 \* x + 7 && y >= x - 2))) {  
 return "Точка попала на границу первой фигуры";  
 }  
 else if (((x >= 0) && (x <= 1) && (y <= 0.5 \* x - 5) && (y >= 0.25 \* x - 5)) ||  
 ((x >= 1) && (x <= 2) && (y <= 0.5 \* x - 5) && (y >= 0.25 \* x - 5)) ||  
 ((x >= 2) && (x <= 3) && ((y <= -x + 1) && (Math.*pow*(x - 4, 2) + Math.*pow*(y + 1, 2) <= 4) || (y <= 0.5 \* x - 5) && (y >= 0.25 \* x - 5))) ||  
 ((x >= 3) && (x <= 4) && ((y <= x - 5) && (Math.*pow*(x - 4, 2) + Math.*pow*(y + 1, 2) <= 4) || (y <= 0.5 \* x - 5) && (y >= 0.25 \* x - 5))) ||  
 ((x >= 4) && (x <= 5) && ((Math.*pow*(x - 4, 2) + Math.*pow*(y + 1, 2) <= 4) && (y >= -x + 5) || (y <= 0 && y >= -2) || (y <= -3 \* x + 10) && (y >= -x))) ||  
 ((x >= 5) && (x <= 6) && (Math.*pow*(x - 4, 2) + Math.*pow*(y + 1, 2) <= 4) && (y >= -2)) ||  
 ((x >= 6) && (x <= 7) && (y <= -x + 5) && (y >= -2))) {  
 return "Точка попала на границу второй фигуры";  
 }  
 else {  
 return "Точка никуда не попала";  
 }  
 }  
 public static void main(String[] args) throws IOException {  
 File file1 = new File("C:\\Users\\Мишка\\Desktop\\z8.txt");  
 FileWriter fwr = new FileWriter(file1, true);  
 fwr.write("Постановка задачи\n\nПервая фигура\nx [-9;4]\n" +  
 "\nВторая фигура\nx[-;-]\n" +  
 "Точки для проверки\n\n№1 (0;2); №2 (6;6); №3 (-3;-2); №4 (6;4); №5 (3;-2)\n\nВЫВОД ДАННЫХ\n\n");  
 fwr.write("Для точки №1\n");  
 fwr.write(*hit*(0, 2)+"\n\n");  
 fwr.write("Для точки №2\n");  
 fwr.write(*hit*(6, 6)+"\n\n");  
 fwr.write("Для точки №3\n");  
 fwr.write(*hit*(-3, -2)+"\n\n");  
 fwr.write("Для точки №4\n");  
 fwr.write(*hit*(6, 4)+"\n\n");  
 fwr.write("Для точки №5\n");  
 fwr.write(*hit*(3, -2));  
 fwr.flush();  
 fwr.close();  
 }  
}

## 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

import java.io.\*;  
public class z9{  
 public static void main(String[] args) throws IOException {  
 File file = new File("C:\\Users\\Мишка\\Desktop\\z9.txt");  
 BufferedReader fread = new BufferedReader(new FileReader(file));  
 BufferedWriter fwrite = new BufferedWriter(new FileWriter(file,true));  
 int abc=0;  
 while (fread.ready()){  
 fread.readLine();  
 abc++;  
 }  
 fread.close();  
 int i =0;  
 String[][] room = new String[abc][15];  
 BufferedReader fread1 = new BufferedReader(new FileReader(file));  
 while (i!=abc-1) {  
 String[] clasi = fread1.readLine().split(",");  
 int c = 0;  
 room[i][0] = clasi[c++];  
 room[i][1] = clasi[c++];  
 room[i][2] = clasi[c++];  
 room[i][3] = clasi[c++];  
 room[i][4] = clasi[c++];  
 room[i][5] = clasi[c++];  
 room[i][6] = clasi[c++];  
 room[i][7] = clasi[c++];  
 room[i][8] = clasi[c++];  
 room[i][9] = clasi[c++];  
 room[i][10] = clasi[c++];  
 if (room[i][10].equalsIgnoreCase("да")) {  
 room[i][11] = clasi[c++];  
 } else {  
 room[i][11] = "0";  
 }  
 room[i][12] = clasi[c++];  
 room[i][13] = clasi[c++];  
 room[i][14] = clasi[c++];  
 i++;  
 }  
 int ShetZifr = 0;  
 for (int k = 0; k < abc; k++) {  
 boolean isOnlyDigits = true;  
 for (int j = 0; j < room[k][1].length() && isOnlyDigits; j++) {  
 if (!Character.*isDigit*(room[k][1].charAt(j))) {  
 isOnlyDigits = false;  
 }  
 }  
 if (isOnlyDigits) {  
 ShetZifr += 0;  
 }  
 }  
 int ScetKlinpoL = 0;  
 int ScetKlinpoZn = 0;  
 int ScetKlinpoAMZ = 0;  
 int ScetKlinpoHTZ = 0;  
 for (int k = 0; k < abc; k++) {  
 if (room[k][4].*valueOf*(k).charAt(0) > 2) {  
 switch (room[k][0]) {  
 case "Лененский":  
 ScetKlinpoL += 1;  
 break;  
 case "Центральный":  
 ScetKlinpoZn += 1;  
 break;  
 case "Чмз":  
 ScetKlinpoAMZ += 1;  
 break;  
 case "Чтз":  
 ScetKlinpoHTZ += 1;  
 break;  
 }  
 }  
 }  
 int KL = 0;  
 for (int k = 0; k < abc; k++) {  
 if ((room[k][3].*valueOf*(k).charAt(0) / 100 \* 30 > room[k][5].*valueOf*(i).charAt(0)) && room[k][6].*valueOf*(k).charAt(0) > 100) {  
 KL += 1;  
 }  
 }  
 System.*out*.println("Количество клиник повышение прошли более 30 % врачей и более 100 пациентов " + KL);  
 int Kw = 0;  
 for (int k = 0; k < abc; k++) {  
 if ((room[k][14].*valueOf*(k).charAt(0) > 5000000)) {  
 if (room[k][9] == "есть") {  
 Kw += 1;  
 }  
 }  
 }  
 fwrite.write("Количество клиник с номером \n" + ShetZifr);  
 fwrite.write("Количество клиник по Центральному р-ну \n" + ScetKlinpoZn);  
 fwrite.write("Количество клиник по Индустриальному р-ну \n" + ScetKlinpoHTZ);  
 fwrite.write("Количество клиник по Мотовилихинском р-ну \n" + ScetKlinpoAMZ);  
 fwrite.write("Количество клиник по Ленинскому р-ну \n" + ScetKlinpoL);  
 fwrite.write("Количество клиник есть рентген и общая стоимость оборудования более 5000000 \n" + Kw);  
 fwrite.close();  
 }  
}

## 9.3. Тестирование работы программы

Не работает

# 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 …

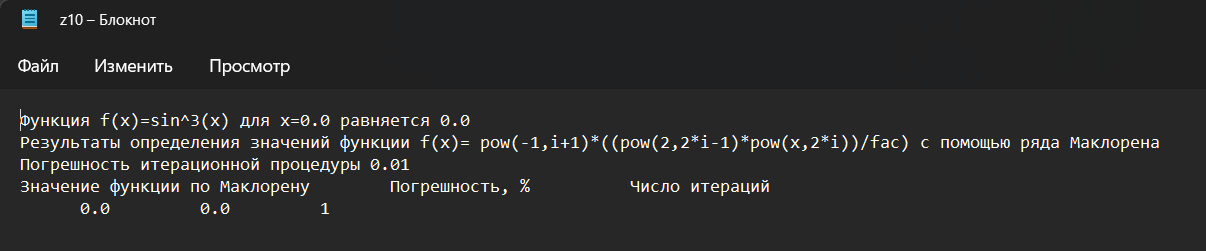
Значение функции по Маклорену Погрешность, % Число итераций

10 4 1

## 10.2. Решение задачи, код программы

import static java.lang.Math.\*;  
import java.io.\*;  
import java.util.Scanner;  
public class z10 {  
 static Scanner *scan* = new Scanner(System.*in*);  
 public static void main(String[] args) throws IOException {  
 double x, er, xz, reazlt, output, sum=0;  
 int step;  
 long fac;  
 File file= new File("C:\\Users\\Мишка\\Desktop\\z10.txt");  
 FileOutputStream file\_Out\_Str = new FileOutputStream(file);  
 OutputStreamWriter fWr = new OutputStreamWriter(file\_Out\_Str);  
 BufferedWriter file\_BW = new BufferedWriter(fWr);  
 System.*out*.print("Введите x: ");  
 x=*scan*.nextDouble();  
 while (x>=1){  
 System.*out*.print("x не должен быть больше или равен 1, повторите ввод: ");  
 x=*scan*.nextDouble();  
 }  
 System.*out*.print("Введите отрицательную степень погрешности (от -2 до -4): ");  
 step=*scan*.nextInt();  
 while (step<-4 || step>-2){  
 System.*out*.print("Вы ввели неправильную степень, повторите ввод: ");  
 step=*scan*.nextInt();  
 }  
 er=*pow*(10,step);  
 xz=*pow*(*sin*(x),3);  
 for (int i=1; i<=99;i++) {  
 fac = 2 \* i;  
 for (long j = (fac - 1); j > 1; j--) {  
 fac = fac \* j;  
 }  
 reazlt=*pow*(-1,i+1)\*((*pow*(2,2\*i-1)\**pow*(x,2\*i))/fac);  
 sum=sum+reazlt;  
 if (*abs*(xz-sum)<er){  
 output=xz-sum;  
 file\_BW.write("Функция f(x)=sin^3(x) для х="+x+" равняется "+xz);  
 file\_BW.write("\nРезультаты определения значений функции f(x)= pow(-1,i+1)\*((pow(2,2\*i-1)\*pow(x,2\*i))/fac) с помощью ряда Маклорена");  
 file\_BW.write("\nПогрешность итерационной процедуры "+er);  
 file\_BW.write("\nЗначение функции по Маклорену\t\t Погрешность, %\t\t Число итераций");  
 file\_BW.write("\n\t"+sum+"\t\t"+output+"\t\t"+i);  
 break;  
 }  
 }  
 file\_BW.close();  
 }  
}

## 10.3. Тестирование работы программы



# Задание 11

## 11.1. Постановка задачи

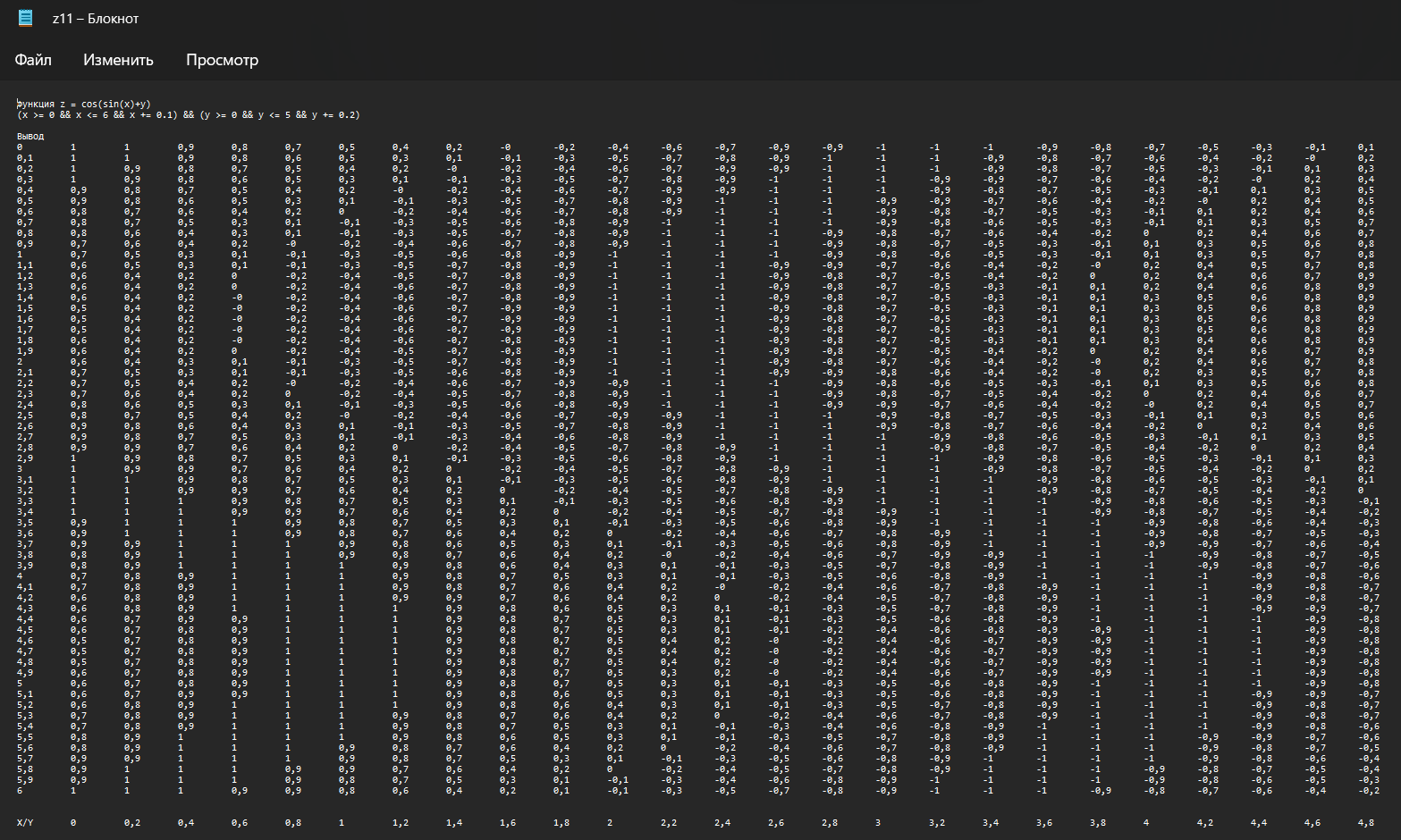
Организовать запись данных о поверхности из лабораторной работы по диаграмма из УИР в файл. Файл содержит шапку о содержимом: формула поверхности, интервалы определения по координатам x и y.

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

## 11.2. Решение задачи, код программы

import java.io.\*;  
import java.text.DecimalFormat;  
import static java.lang.Math.\*;  
public class z11 {  
 public static void main(String[] args) throws IOException {  
 File file = new File("C:\\Users\\Мишка\\Desktop\\z11.txt");  
 BufferedWriter fwrite = new BufferedWriter(new FileWriter(file));  
 fwrite.write("Функция z = cos(sin(x)+y)\n" +  
 "(x >= 0 && x <= 6 && x += 0.1) && (y >= 0 && y <= 5 && y += 0.2)\n\nВывод\n");  
 DecimalFormat dec = new DecimalFormat("####.#");  
 DecimalFormat dec1 = new DecimalFormat("#.#");  
 for (double x = 0;x<=6;x+=0.1){  
 fwrite.write(dec1.format(x)+"\t");  
 for (double y = 0;y<=5;y+=0.2){  
 double z = *cos*(*sin*(x)+y);  
 fwrite.write(dec.format(z)+"\t");  
 }  
 fwrite.write("\n");  
 }  
 fwrite.write("\n\nX/Y\t");  
 for (double y = 0;y<=5;y+=0.2){  
 fwrite.write(dec1.format(y)+"\t");  
 }  
 fwrite.close();  
 }  
}

## 11.3. Тестирование работы программы



# Задание 12

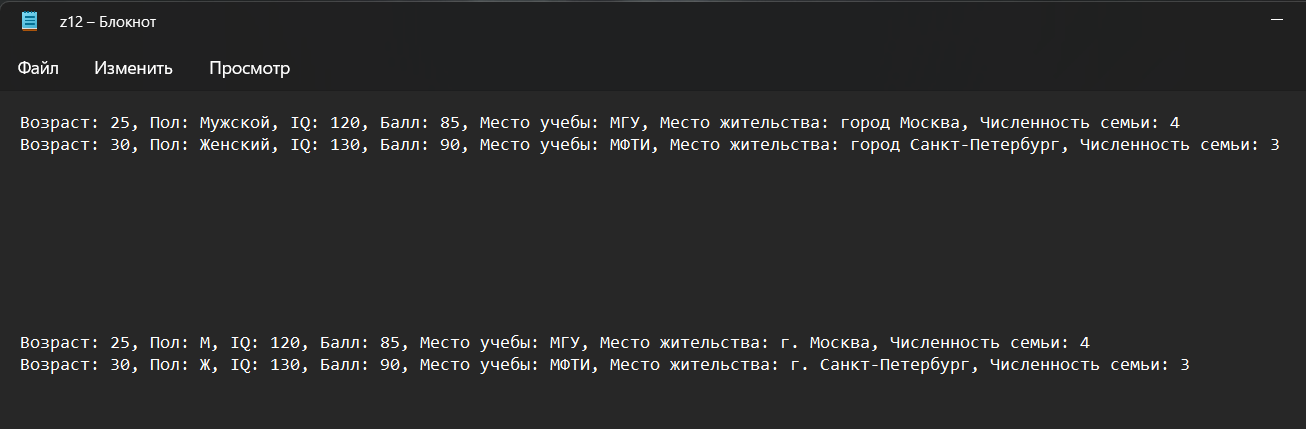
## 12.1. Постановка задачи

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.\*;  
import java.util.\*;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
public class z12 {  
 public static void main(String[] args) throws IOException {  
 String line;  
 ArrayList<Object> list = new ArrayList<>();  
 File file= new File("C:\\Users\\Мишка\\Desktop\\z12.txt");  
 FileInputStream file\_In\_Str = new FileInputStream(file);  
 InputStreamReader fRe=new InputStreamReader(file\_In\_Str);  
 BufferedReader file\_BR = new BufferedReader(fRe);  
 FileOutputStream file\_Out\_Str = new FileOutputStream(file,true);  
 OutputStreamWriter fWr = new OutputStreamWriter(file\_Out\_Str);  
 BufferedWriter file\_BW = new BufferedWriter(fWr);  
 file\_BW.newLine();  
 String[] Patt = {"Мужской", "Женский", "край", "город", "область"};  
 String[] ZamPatt = {"М", "Ж", "кр.", "г.", "обл."};  
 Pattern pat1;  
 Matcher mat1;  
 while((line = file\_BR.readLine()) != null) {  
 String[] str=line.split("\\.");  
 Collections.*addAll*(list, str);  
 }  
 file\_BW.newLine();  
 String[] array = list.toArray(new String[list.size()]);  
  
 for (int i=0;i<array.length;i++){  
 for (int j=0;j<Patt.length;j++){  
 pat1=Pattern.*compile*(Patt[j]);  
 mat1=pat1.matcher(array[i]);  
 if (mat1.find()) array[i]=mat1.replaceAll(ZamPatt[j]);  
 }  
 }  
 for (String s : array) {  
 file\_BW.write("\n"+s);  
 }  
 file\_BW.close();  
 }  
}

## 12.3. Тестирование работы программы



# Задание 13

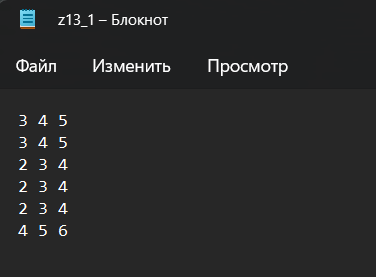
## 13.1. Постановка задачи

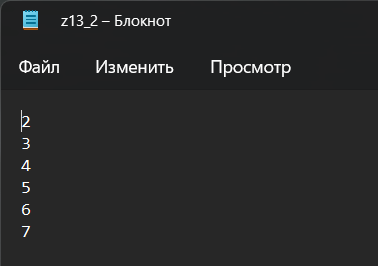
В первом файле хранится k матриц размерности m ×n, во втором – k матриц размерности n ×1. Перемножить матрицы из первого и второго файла. Записать третий файл в виде компонент структуры, где каждая компонента содержит три матрицы: первая размерности m ×n из первого файла; вторая размерности n ×1 из второго файла; третья, матрица размерности m ×1, результат произведения. Вывести на экран содержимое всех файлов.

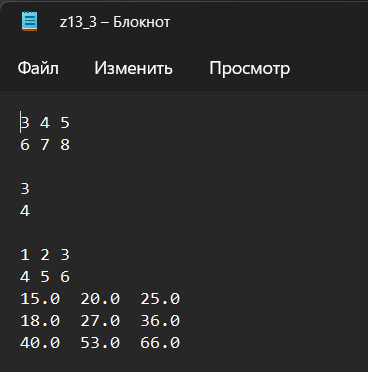
## 13.2. Решение задачи, код программы

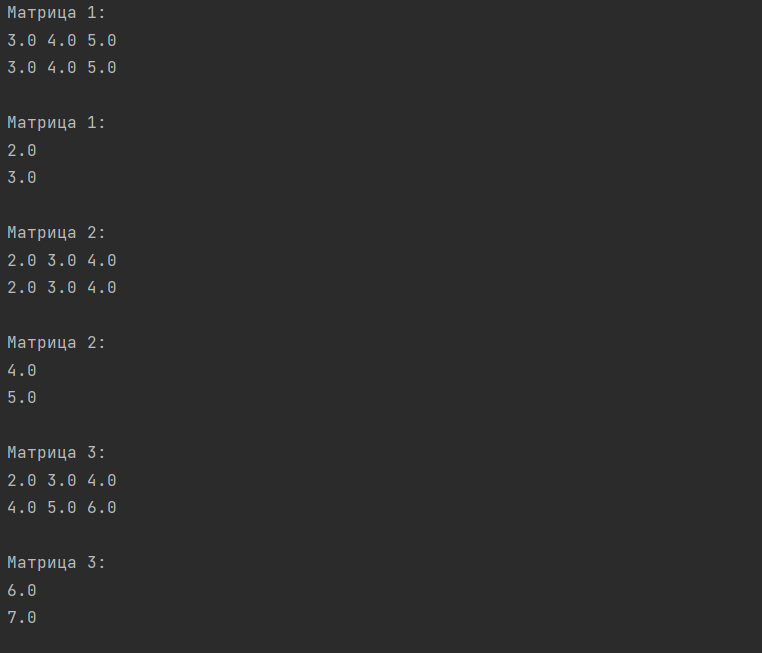
import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.io.BufferedWriter;  
public class z13 {  
 public static void main(String[] args) {  
 int k = 3;  
 int n = 2;  
 int m = 3;  
 double[][][] matrices = *readMatricesFromFile*("C:\\Users\\Мишка\\Desktop\\z13\_1.txt", k, n, m);  
 double[][] matrices1 = *readMatricesFromFile1*("C:\\Users\\Мишка\\Desktop\\z13\_2.txt", k, n);  
 String filePath = "C:\\Users\\Мишка\\Desktop\\z13\_3.txt";  
 for (int i = 0; i < k; i++) {  
 System.*out*.println("Матрица " + (i + 1) + ":");  
 for (int j = 0; j < n; j++) {  
 for (int l = 0; l < m; l++) {  
 System.*out*.print(matrices[i][j][l] + " ");  
 }  
 System.*out*.println();  
 }  
 System.*out*.println();  
 System.*out*.println("Матрица " + (i + 1) + ":");  
 for (int j = 0; j < n; j++) {  
 System.*out*.println(matrices1[i][j]);  
 }  
 System.*out*.println();  
 }  
 for (int i = 0; i < k; i++) {  
 double[][] vspom = new double[n][m];  
 double[] vspom2 = new double[n];  
 for (int j = 0; j < n; j++) {  
 for (int l = 0; l < m; l++) {  
 vspom[j][l] = matrices[i][j][l];  
 }  
 vspom2[j] = matrices1[i][j];  
 }  
 System.*out*.println("Умножение матрицы " + (i + 1) + " из первого файла и матрицы " + (i + 1) + " из второго файла:");  
 double[] matrixesitog =*multiplyMatrices*(vspom,vspom2);  
 *writeArrayToFile*(matrixesitog, filePath);  
 System.*out*.println("Матрица успешно записана в файл.");  
 for (int p = 0; p < matrixesitog.length; p++) {  
 System.*out*.println(matrixesitog[p]);  
 }  
 System.*out*.println();  
 }  
 }  
 public static void writeArrayToFile(double[] array, String filename) {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(filename, true))) {  
 StringBuilder line = new StringBuilder();  
 for (double element : array) {  
 line.append(element).append("\t");  
 }  
 writer.write(line.toString().trim());  
 writer.newLine();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 public static double[] multiplyMatrices(double[][] matrix1, double[] matrix2) {  
 int n = matrix1.length;  
 int m = matrix1[0].length;  
 if (n != matrix2.length) {  
 throw new IllegalArgumentException("Несовместимые размерности матрицы и массива");  
 }  
 double[] result = new double[m];  
 for (int i = 0; i < m; i++) {  
 for (int j = 0; j < n; j++) {  
 result[i]+=matrix1[j][i]\*matrix2[j];  
 }  
 }  
 return result;  
 }  
 public static double[][][] readMatricesFromFile(String filename, int k, int n, int m) { //чтение матриц вида n на m  
 double[][][] matrices = new double[k][n][m];  
 try (BufferedReader reader = new BufferedReader(new FileReader(filename))) {  
 String line;  
 int currentMatrix = 0;  
 int currentRow = 0;  
 while ((line = reader.readLine()) != null) {  
 String[] values = line.trim().split("\\s+");  
 for (int i = 0; i < values.length; i++) {  
 matrices[currentMatrix][currentRow][i] = Double.*parseDouble*(values[i]);  
 }  
 currentRow++;  
 if (currentRow == n) {  
 currentMatrix++;  
 currentRow = 0;  
 }  
 if (currentMatrix == k) {  
 break;  
 }  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 return matrices;  
 }  
 public static double[][] readMatricesFromFile1(String filename, int k, int n) {  
 double[][] matrices1 = new double[k][n];  
 try (BufferedReader reader = new BufferedReader(new FileReader(filename))) {  
 String line;  
 int currentMatrix = 0;  
 int currentRow = 0;  
  
 while ((line = reader.readLine()) != null) {  
 double value = Double.*parseDouble*(line.trim());  
 matrices1[currentMatrix][currentRow] = value;  
 currentRow++;  
 if (currentRow == n) {  
 currentMatrix++;  
 currentRow = 0;  
 }  
 if (currentMatrix == k) {  
 break;  
 }  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 return matrices1;  
 }  
}

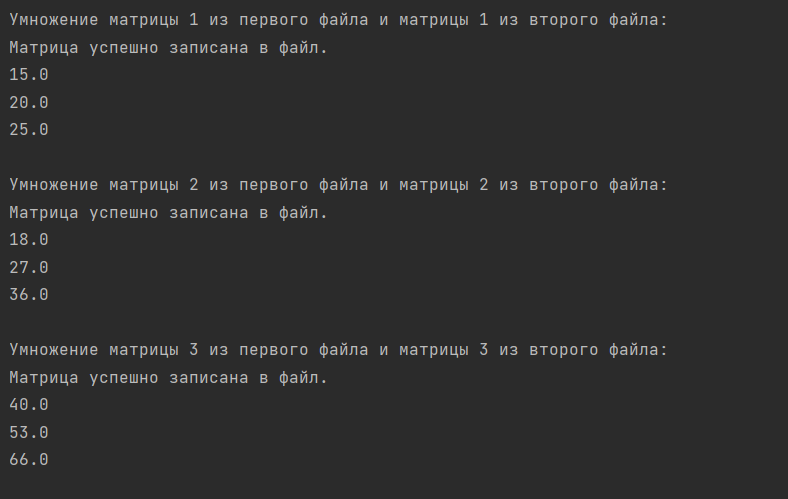
## 13.3. Тестирование работы программы











# 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.BufferedReader;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.List;  
public class z14 {  
 public static void main(String[] args) throws IOException {  
 String filename = "C:\\Users\\Мишка\\Desktop\\Variant\_5,15.txt";  
 List<List<Double>> data = new ArrayList<>();  
 BufferedReader fread = new BufferedReader(new FileReader(filename));  
 String line;  
 boolean flag = false;  
 while ((line = fread.readLine()) != null) {  
 if (line.contains("UX")) {  
 flag = true;  
 continue;  
 }  
 if (flag) {  
 String[] correct = fread.readLine().split("-0\\.");  
 for (int i = 1; i < correct.length; i++) {  
 correct[i] = "\\s-0." + correct[i];  
 }  
 String s = *getString*(correct).replaceFirst("\\s+", "");  
 String[] values = s.trim().split("\\s+");  
 List<Double> row = new ArrayList<>();  
 for (int i = 1; i < values.length; i++) {  
 row.add(Double.*parseDouble*(values[i]));  
 }  
 data.add(row);  
 }  
 }  
 fread.close();  
 List<Double> maxValues = new ArrayList<>();  
 List<Double> minValues = new ArrayList<>();  
 List<Double> avgValues = new ArrayList<>();  
 List<List<Double>> deviations = new ArrayList<>();  
 for (int i = 0; i < data.get(0).size(); i++) {  
 double max = Double.*MIN\_VALUE*;  
 double min = Double.*MAX\_VALUE*;  
 double sum = 0;  
 for (List<Double> row : data) {  
 double value = row.get(i);  
 max = Math.*max*(max, value);  
 min = Math.*min*(min, value);  
 sum += value;  
 }  
 double avg = sum / data.size();  
 maxValues.add(max);  
 minValues.add(min);  
 avgValues.add(avg);  
 List<Double> deviation = new ArrayList<>();  
 for (List<Double> row : data) {  
 deviation.add(row.get(i) - avg);  
 }  
 deviations.add(deviation);  
 }  
 FileWriter fwr = new FileWriter(filename, true);  
 fwr.write("\nМаксимальные значения: " + maxValues.toString() + "\n");  
 fwr.write("Минимальные значения: " + minValues.toString() + "\n");  
 fwr.write("Средние значения: " + avgValues.toString() + "\n");  
 fwr.write("Отклонения:\n");  
 for (List<Double> deviation : deviations) {  
 fwr.write(deviation.toString() + "\n");  
 }  
 fwr.flush();  
 fwr.close();  
 }  
 public static String getString(String[] mas) {  
 StringBuilder stringBuilder = new StringBuilder();  
 for (int i = 0; i < mas.length; i++) {  
 stringBuilder.append(mas[i]);  
 }  
 return stringBuilder.toString();  
 }  
}

## 14.3. Тестирование работы программы

Не работает