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»

Вариант 2

Выполнил: студент группу ИСТ-22-1б Афонина А.М

Проверил: доцент каф. ВММБ Каменских А.А

Пермь, 2022

**Содержание**

[Задание 1 3](#_Toc136614964)

[1.1. Постановка задачи 3](#_Toc136614965)

[1.2. Решение задачи, код программы 3](#_Toc136614966)

[1.3. Тестирование работы программы 6](#_Toc136614967)

[Задание 2 6](#_Toc136614968)

[1.1. Постановка задачи 6](#_Toc136614969)

[1.2. Решение задачи, код программы 6](#_Toc136614970)

[1.3. Тестирование работы программы 7](#_Toc136614971)

[Задание 3 7](#_Toc136614972)

[1.1. Постановка задачи 7](#_Toc136614973)

[1.2. Решение задачи, код программы 7](#_Toc136614974)

[1.3. Тестирование работы программы 7](#_Toc136614975)

[Задание 4 8](#_Toc136614976)

[1.1. Постановка задачи 8](#_Toc136614977)

[1.2. Решение задачи, код программы 8](#_Toc136614978)

[1.3. Тестирование работы программы 9](#_Toc136614979)

[Задание 5 9](#_Toc136614982)

[1.1. Постановка задачи 9](#_Toc136614983)

[1.2. Решение задачи, код программы 9](#_Toc136614984)

[1.3. Тестирование работы программы 10](#_Toc136614985)

[Задание 6 10](#_Toc136614986)

[1.1. Постановка задачи 10](#_Toc136614987)

[1.2. Решение задачи, код программы 10](#_Toc136614988)

[1.3. Тестирование работы программы 11](#_Toc136614989)

[Задание 7 11](#_Toc136614990)

[1.1. Постановка задачи 11](#_Toc136614991)

[1.2. Решение задачи, код программы 11](#_Toc136614992)

[1.3. Тестирование работы программы 12](#_Toc136614993)

[Задание 8 12](#_Toc136614994)

[1.1. Постановка задачи 12](#_Toc136614995)

[1.2. Решение задачи, код программы 12](#_Toc136614996)

[1.3. Тестирование работы программы 14](#_Toc136614997)

# Задание 1

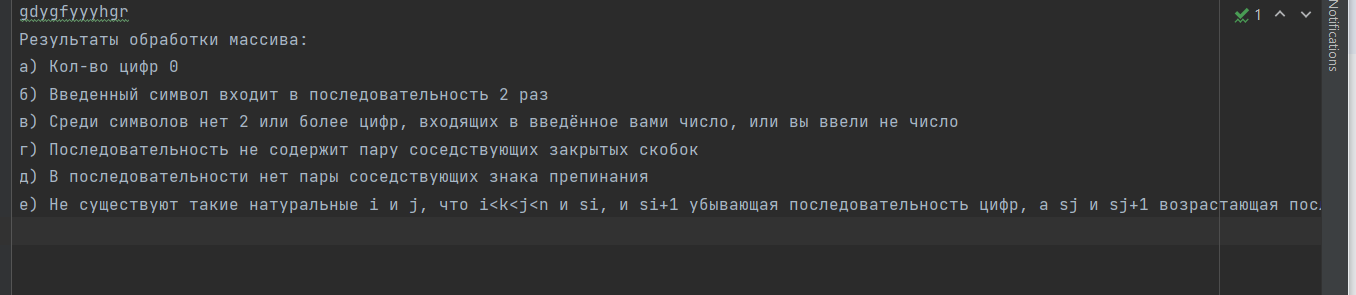
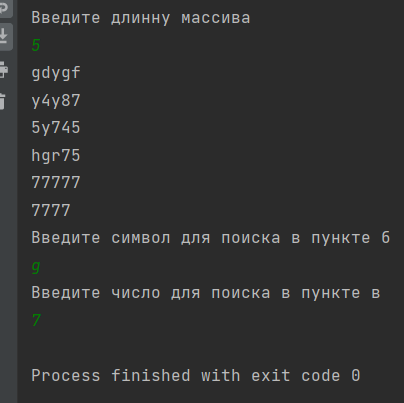
## 1.1. Постановка задачи

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.\*;  
import java.io.\*;  
  
public class main10\_1 {  
 public static char vvod() {  
 Scanner input = new Scanner(System.*in*);  
 String x = input.nextLine();  
 char t;  
 while (true) {  
 if (x.length() != 1) {  
 System.*out*.println("Элемент массива должен быть одним символом");  
 x = input.nextLine();  
 } else {  
 t = x.charAt(0);  
 break;  
 }  
 }  
 return t;  
 }  
  
 public static String vvodstr() {  
 Scanner input = new Scanner(System.*in*);  
 return input.nextLine();  
 }  
  
 public static void mas(char[] arr) {  
 try (FileReader reader = new FileReader("10\_1.txt")) {  
 int c;  
 while ((c = reader.read(arr)) > 0) {  
  
 if (c < 10) {  
 arr = Arrays.*copyOf*(arr, c);  
 }  
 System.*out*.println(arr);  
 }  
 } catch (IOException ex) {  
  
 System.*out*.println(ex.getMessage());  
 }  
 }  
  
 public static int a(char[] arr) {  
 int digitsCount = 0;  
 for (Character character : arr) {  
 if (String.*valueOf*(character).matches("[0-9]")) {  
 digitsCount++;  
 }  
 }  
 return ( digitsCount);  
 }  
  
 public static int b(char[] arr, char s){  
 int countb = 0;  
 for (Character character : arr) {  
 if (character == s) {  
 countb += 1;  
 }  
 }  
 return countb;  
 }  
  
 public static boolean c ( char[] arr, String inputNumber){  
 int count = 0;  
 for (char ch : arr) {  
 if (Character.*isDigit*(ch) && inputNumber.contains(String.*valueOf*(ch))) {  
 count++;  
 if (count >= 2) {  
 return true;  
 }  
 }  
 }  
 return false;  
 }  
  
 public static boolean d ( char[] arr){  
 for (int i = 0; i < arr.length - 1; i++) {  
 if ((arr[i] == '(' && arr[i + 1] == ')') || (arr[i] == '{' && arr[i + 1] == '}') || (arr[i] == '[' && arr[i + 1] == ']')) {  
 return true;  
 }  
 }  
 return false;  
 }  
  
  
 public static boolean e ( char[] arr){  
 for (int i = 1; i < arr.length; i++) {  
 if (Character.*getType*(arr[i]) == Character.*getType*(arr[i-1]) && Character.*getType*(arr[i]) == Character.*CONNECTOR\_PUNCTUATION*){  
 return true;  
 }  
 }  
 return false;  
 }  
  
 public static boolean f ( char[] arr){  
 for (int i = 0; i < arr.length - 3; i++) {  
 if (Character.*isDigit*(arr[i]) &&  
 Character.*isDigit*(arr[i + 1]) &&  
 Character.*isDigit*(arr[i + 2])) {  
 int num1 = arr[i];  
 int num2 = arr[i + 1];  
 int num3 = arr[i + 2];  
 if (num1 > num2 && num2 > num3) {  
 return true;  
 }  
 }  
 }  
 return false;  
 }  
  
 public static void main (String[]args) throws IOException {  
 Scanner input = new Scanner(System.*in*);  
 System.*out*.println("Введите длинну массива");  
 int len = input.nextInt();  
 char[] arr = new char[len];  
 *mas*(arr);  
 System.*out*.println("Введите символ для поиска в пункте б");  
 char b = *vvod*();  
 System.*out*.println("Введите число для поиска в пункте в");  
 String c = *vvodstr*();  
 FileWriter writer = new FileWriter("10\_1.txt", true);  
 PrintWriter printer = new PrintWriter(writer);  
 printer.println("\nРезультаты обработки массива:");  
  
 printer.printf("а) Кол-во цифр %d\n", *a*(arr));  
 printer.println("б) Введенный символ входит в последовательность " + *b*(arr, b) + " раз");  
  
 if (*c*(arr, c)) {  
 printer.println("в) Среди символов есть 2 или более цифры, входящие в число " + c);  
 } else {  
 printer.println("в) Среди символов нет 2 или более цифр, входящих в введённое вами число, или вы ввели не число");  
 }  
  
 if (*d*(arr)) {  
 printer.println("г) Последовательность содержит пару соседствующих закрытых скобок");  
 } else {  
 printer.println("г) Последовательность не содержит пару соседствующих закрытых скобок");  
 }  
  
 if (*e*(arr)) {  
 printer.println("д) В последовательности есть пара соседстсвующих знака препинания");  
 } else {  
 printer.println("д) В последовательности нет пары соседствующих знака препинания");  
 }  
  
 if (*f*(arr)) {  
 printer.println("е) Существуют такие натуральные i и j, что i<k<j<n и " +  
 "si, и si+1 убывающая последовательность цифр, a sj и sj+1 возрастающая последовательность чисел");  
 } else {  
 printer.println("е) Не существуют такие натуральные i и j, что i<k<j<n и " +  
 "si, и si+1 убывающая последовательность цифр, a sj и sj+1 возрастающая последовательность чисел");  
 }  
 printer.flush();  
 printer.close();  
 }  
 }

## 1.3. Тестирование работы программы



# Задание 2

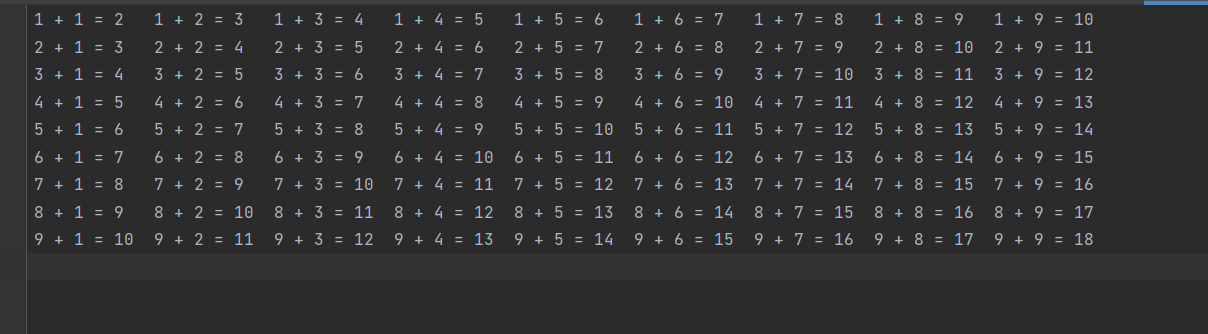
## 1.1. Постановка задачи

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.

## 1.2. Решение задачи, код программы

import java.io.FileWriter;  
public class main10\_2 {  
 public static void main(String[] args) throws IOException{  
 File file = new File("10\_2.txt");  
 FileWriter FW = new FileWriter(file);  
 int a;  
 int b;  
 int c ;  
 for (a = 1; a<=9;a++) {  
 for ( b = 1; b <= 9; b++) {  
 c = a + b;  
 FW.write(a + " + " + b + " = " + c + "\t");  
 }  
 FW.write("\n");  
 }  
 FW.close();  
 }  
}

## 1.3. Тестирование работы программы



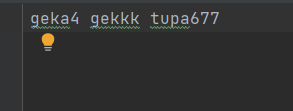
# Задание 3

## 1.1. Постановка задачи

 Напечатать все слова из текстового файла, содержащие хотя бы одну цифру.

## 1.2. Решение задачи, код программы

import java.io.\*;  
import java.util.ArrayList;  
  
public class main10\_3 {  
 public static void main(String[] args) throws IOException {  
 File file = new File("10\_3.txt");  
 ArrayList<String> words = new ArrayList<>();  
 try (BufferedReader BR = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = BR.readLine()) != null) {  
 String[] lineWords = line.split(" ");  
 for (String word : lineWords) {  
 if (word.matches(".\*\\d.\*")) {  
 words.add(word);  
 }  
  
  
 }  
 }  
 for (String word : words) {  
 System.*out*.println(word);  
 }  
 }  
 }}



## 1.3. Тестирование работы программы



# Задание 4

## 1.1. Постановка задачи

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

## 1.2. Решение задачи, код программы

import java.util.ArrayList;  
import java.io.\*;  
  
public class main10\_4 {  
 public static void main(String[] args) throws IOException {  
 File inputFile = new File("4\_1.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("4\_2.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;  
 for (char c : word.toCharArray()) {  
 if ("aeiouAEIOU".indexOf(c) != -1) {  
 numglas++;  
 } else if (Character.*isLetter*(c)) {  
 numsoglas++;  
 }  
 }  
 return numglas == numsoglas;  
 }  
}

## 1.3. Тестирование работы программы

## 

## 

# Задание 5

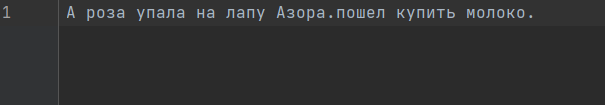
## 1.1. Постановка задачи

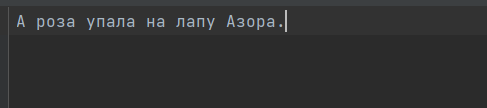
Дано два файла с предложениями. Дописать в конец второго файла предложения из первого файла, в которых есть симметричные слова.

## 1.2. Решение задачи, код программы

import java.util.ArrayList;  
import java.io.\*;  
  
public class main10\_5 {  
 public static void main(String[] args) throws IOException {  
 File inputFile = new File("5\_1.txt");  
 ArrayList<String> predlojenia = new ArrayList<>();  
 try (BufferedReader reader = new BufferedReader(new FileReader(inputFile))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] lineSent = line.split("[.!?]");  
 for (String pred : lineSent) {  
 if (*hasSymmetricWord*(pred)) {  
 predlojenia.add(pred);  
 }  
 }  
 }  
 }  
  
 File outputFile = new File("5\_2.txt");  
 try (PrintWriter writer = new PrintWriter(outputFile)) {  
 for (String pred : predlojenia) {  
 writer.println(pred);  
 }  
 }  
 }  
  
 private static boolean hasSymmetricWord(String pred) {  
 String[] words = pred.toLowerCase().split("\\W+");  
 for (String word : words) {  
 if (*isSymmetric*(word)) {  
 return true;  
 }  
 }  
 return false;  
 }  
  
 private static boolean isSymmetric(String word) {  
 int length = word.length();  
 for (int i = 0; i < length / 2; i++) {  
 if (word.charAt(i) != word.charAt(length - 1 - i)) {  
 return false;  
 }  
 }  
 return true;  
 }  
}

## 1.3. Тестирование работы программы





# Задание 6

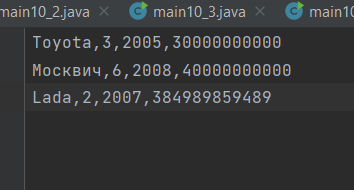
## 1.1. Постановка задачи

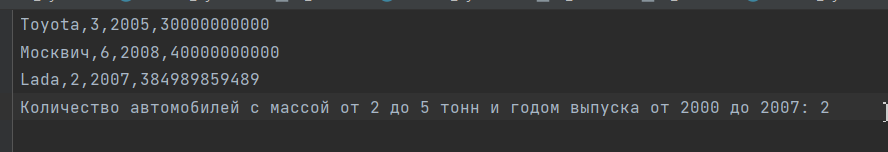
 Входной файл содержит сведения о спортивных товарах: наименование, вид спорта, масса, год выпуска, цена. Найти товары с массой от 15 до 25 кг, с ценой от 10000 до 50000, подсчитать их количество и дописать ответ в конец файла.

## 1.2. Решение задачи, код программы

import java.io.\*;  
  
public class main10\_6 {  
 public static void main(String[] args) throws IOException {  
 File inputFile = new File("6\_1.txt");  
 File outputFile = new File("6\_2.txt");  
  
 BufferedReader reader = new BufferedReader(new FileReader(inputFile));  
 BufferedWriter writer = new BufferedWriter(new FileWriter(outputFile));  
  
 String line;  
 int count = 0;  
  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(",");  
 String brand = parts[0];  
 double mass = Double.*parseDouble*(parts[1]);  
 int year = Integer.*parseInt*(parts[2]);  
 double price = Double.*parseDouble*(parts[3]);  
  
 if (mass >= 2 && mass <= 5 && year >= 2000 && year <= 2007) {  
 count++;  
 }  
  
 writer.write(line + "\n");  
 }  
  
 writer.write("Количество автомобилей с массой от 2 до 5 тонн и годом выпуска от 2000 до 2007: " + count);  
  
 reader.close();  
 writer.close();  
 }  
}

## 1.3. Тестирование работы программы





# Задание 7

## 1.1. Постановка задачи

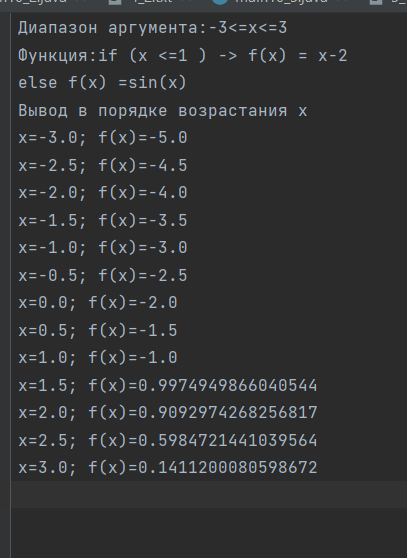
 Записать в файл значения функции f(x) из задания 3 лабораторной работы 5 для всего диапазона переменной в виде: значение x; значение f(x). В начале файла должна быть указана постановка задачи, т.е. диапазон аргумента и вид функции (запись на языке java). Перед выводом цифровых значений должен быть сделан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



## 1.2. Решение задачи, код программы

import java.io.\*;  
  
public class main10\_7 {  
 public static void main(String[] args) throws IOException{  
  
 FileWriter writer = new FileWriter("10\_7.txt");  
 writer.write("Диапазон аргумента:-3<=x<=3\n"+  
 "Функция:if (x <=1 ) -> f(x) = x-2\n"+  
 "else f(x) =sin(x)\n" +  
 "Вывод в порядке возрастания x\n");  
 for (Double x = -3.0; x <= 3.0; x += 0.5) {  
 if (x <=1) {  
 writer.write(("x=" + x + "; f(x)=" + (x- 2) + "\n"));  
 } else writer.write(("x=" + x + "; f(x)=" + Math.*sin*(x) + "\n"));  
 }  
 writer.close();  
 }  
 }

## 1.3. Тестирование работы программы



# Задание 8

## 1.1. 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.



## 1.2. Решение задачи, код программы

import java.io.\*;  
import java.util.Objects;  
  
public class main10\_8 {  
 public static String hittt( int x, int y) {  
 if((x > -3 & x < 2 & y > -(2 / 3) \* x - 3 & y > (7 / 2) \* x - 3 & y < 4 & y > -3) || (x > -3 & x < 0 & y > -(2 / 3) \* x - 3 & y < x + 2 & y < 1 & y > -3) || (y < x + 2 & x > -3 & x < -1 & x > -1 \* (Math.*sqrt*(4 - Math.*pow*(y + 3, 2))) - 1 & y < 5 & y > 1) || (y < Math.*sqrt*(4 - Math.*pow*(x + 1, 2)) + 3 & x > -3 & x < -1 & y < -(1 / 2) \* x + (5 / 2) & y < 5) || (x > -2 & x < 2 & y < 4 & y < -(1 / 2) \* x + (5 / 2) & y < 2 \* x) & (x > 1 & x < 2 & y < 4 & y > -3 & y <2 \* x & y > (7 / 2) \* x - 3)){  
 return"попало в область 1 фигуры";  
 } else if ((x >= -3 & x <= 2 & y >= -(2 / 3) \* x - 3 & y >= (7 / 2) \* x - 3 & y <= 4 & y >= -3) || (x > -3 & x <= 0 & y >= -(2 / 3) \* x - 3 & y <= x + 2 & y <= 1 & y >= -3) || (y <= x + 2 & x >= -3 & x <= -1 & x >= -1 \* (Math.*sqrt*(4 - Math.*pow*(y + 3, 2))) - 1 & y < 5 & y >= 1) || (y <= Math.*sqrt*(4 - Math.*pow*(x + 1, 2)) + 3 & x >= -3 & x <= -1 & y <= -(1 / 2) \* x + (5 / 2) & y <= 5) || (x >= -2 & x <= 2 & y <= 4 & y <= -(1 / 2) \* x + (5 / 2) & y <= 2 \* x) & (x >= 1 & x <= 2 & y <= 4 & y >= -3 & y <= 2 \* x & y >= (7 / 2) \* x - 3)) {  
 return "попало на границу 1 фигуры";}  
 if ((y > - Math.*sqrt*(4-Math.*pow*(x-4,2))-1 & x>2 & x<7 & y>-3 & y<0 & y> x-7) || (y > - Math.*sqrt*(4-Math.*pow*(x-4,2))-1 & x>2 & y>-3 & y<0 & x<6 & y<x-3)|| (y<x-3 & y<-x+3 & x>2 & y <1 & y >-1 & x<6)||(y<-x+3 & x>1 & x<5 & y>0 & y<1 & y<-((1)/(3))\*x + (5/3))||(y<-((1)/(3))\*x + (5/3) & x>2 & x< 7 & y<5 & y>-1)||( y<5 & y>0 & x>5 & x<7 & y<-((5)/(2))\*x+((35)/(2)))||(y<-((5)/(2))\*x+((35)/(2)) & y<5 & y>-1 & x>5 & x<7 & y>x-7 )){  
 return "попало в область 2 фигуры";  
 }  
 else if ((y >= - Math.*sqrt*(4-Math.*pow*(x-4,2))-1 & x>=2 & x<=7 & y>=-3 & y<=0 & y>= x-7) || (y >= - Math.*sqrt*(4-Math.*pow*(x-4,2))-1 & x>=2 & y>=-3 & y<=0 & x<=6 & y<=x-3)|| (y<=x-3 & y<=-x+3 & x>=2 & y <=1 & y >=-1 & x<=6)||(y<=-x+3 & x>=1 & x<=5 & y>=0 & y<=1 & y<=-((1)/(3))\*x + (5/3))||(y<=-((1)/(3))\*x + (5/3) & x>=2 & x<= 7 & y<=5 & y>=-1)||( y<=5 & y>=0 & x>=5 & x<=7 & y<=-((5)/(2))\*x+((35)/(2)))||(y<=-((5)/(2))\*x+((35)/(2)) & y<=5 & y>=-1 & x>=5 & x<=7 & y>=x-7 )){  
 return "попало на границу 2 фигуры";  
 }  
 else {  
 return "Никуда не попало";  
 }  
 }  
  
 public static void main(String[] args) throws IOException{  
 File file = new File("10\_8.txt");  
 FileWriter writer = new FileWriter(file,true);  
 writer.write("Фигура 1\n"+  
 "1 область\n"+  
 "(x >= -3 & x<=2 & y >=-(2/3)\*x-3 & y>= (7/2)\*x-3 & y <=4 & y>=-3) || (x>-3 & x <=0 & y >=-(2/3)\*x-3 & y<=x+2 & y<=1 & y>=-3)\n"+  
 "2 область\n"+" (y<=x+2 & x>=-3 & x<=-1 & x >= -1\*(Math.sqrt(4-Math.pow(y+3,2)))-1 & y<5 & y>=1) || (y <= Math.sqrt(4-Math.pow(x+1,2))+3 & x>=-3 & x<=-1& y<=-(1/2)\*x+(5/2) & y<=5)"  
 +"3 область\n"+  
 "(x>=-2 & x<=2 & y <=4 & y<=-(1/2)\*x+(5/2) & y<=2\*x) &(x>=1 & x<=2 & y <=4 & y>=-3 & y<=2\*x & y>=(7/2)\*x -3))\n"+  
 "Фигура 2\n"+  
 "1 область\n"+"(y >= - Math.sqrt(4-Math.pow(x-4,2))-1 & x>=2 & x<=7 & y>=-3 & y<=0 & y>= x-7) || (y >= - Math.sqrt(4-Math.pow(x-4,2))-1 & x>=2 & y>=-3 & y<=0 & x<=6 & y<=x-3)\n"+  
 "2 область\n"+"(y<=x-3 & y<=-x+3 & x>=2 & y <=1 & y >=-1 & x<=6)||(y<=-x+3 & x>=1 & x<=5 & y>=0 & y<=1 & y<=-((1)/(3))\*x + (5/3))\n"+  
 "3 область\n"+"(y<=-((1)/(3))\*x + (5/3) & x>=2 & x<= 7 & y<=5 & y>=-1)||( y<=5 & y>=0 & x>=5 & x<=7 & y<=-((5)/(2))\*x+((35)/(2)))\n"+  
 "4 область\n"+ "(y<=-((5)/(2))\*x+((35)/(2)) & y<=5 & y>=-1 & x>=5 & x<=7 & y>=x-7 )\n"+  
 "Точки ( -3; 2) , ( 0,1), (4;-1),(1000;-3)\n"  
 +"Вывод\n");  
 writer.write("Для 1 точки ( -3; 2) ");  
 writer.write(*hittt*(-3, 2)+ "\n");  
 writer.write("Для 2 точки ( 0; 1) ");  
 writer.write(*hittt*(0, 1)+ "\n");  
 writer.write("Для 3 точки ( 4; -1) ");  
 writer.write(*hittt*(4, -1)+ "\n");  
 writer.write("Для 4 точки ( 1000; -3) ");  
 writer.write(*hittt*(1000, -3)+ "\n");  
 writer.flush();  
 writer.close();  
  
 }  
  
}

## 1.3. Тестирование работы программы

