Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 4

тема «Линейные алгоритмы в Java»

по дисциплине «Информатика»

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

Проверил: Нетбай Г.В.

Пермь, 2022

**Содержание**

[Задание 1 3](#_Toc119768185)

[1.1. Постановка задачи 3](#_Toc119768186)

[1.2. Решение задачи, код программы 3](#_Toc119768187)

[1.3. Тестирование работы программы с проверкой 3](#_Toc119768188)

[Задание 2 5](#_Toc119768189)

[2.1. Постановка задачи 5](#_Toc119768190)

[2.2. Решение задачи, код программы 5](#_Toc119768191)

[2.3. Тестирование работы программы с проверкой 6](#_Toc119768192)

[Задание 3 8](#_Toc119768193)

[3.1. Постановка задачи 8](#_Toc119768194)

[3.2. Решение задачи, код программы 8](#_Toc119768195)

[3.3. Тестирование работы программы с проверкой 8](#_Toc119768196)

[Задание 4 10](#_Toc119768197)

[4.1. Постановка задачи 10](#_Toc119768198)

[4.2. Решение задачи, код программы 10](#_Toc119768199)

[4.3. Тестирование работы программы с проверкой 10](#_Toc119768200)

[Задание 5 12](#_Toc119768201)

[5.1. Постановка задачи 12](#_Toc119768202)

[5.2. Решение задачи, код программы 12](#_Toc119768203)

[5.3. Тестирование работы программы с проверкой 15](#_Toc119768204)

[Задание 6 18](#_Toc119768205)

[6.1. Постановка задачи 18](#_Toc119768206)

[6.2. Решение задачи, код программы 18](#_Toc119768207)

[6.3. Тестирование работы программы с проверкой 19](#_Toc119768208)

[Задание 7 22](#_Toc119768209)

[7.1. Постановка задачи 22](#_Toc119768210)

[7.2. Решение задачи, код программы 22](#_Toc119768211)

[7.3. Тестирование работы программы с проверкой 23](#_Toc119768212)

# Задание 1

## 1.1. Постановка задачи

1. Написать программу, которая находит наименьшее и наибольшее среди трех заданных чисел.

## 1.2. Решение задачи, код программы

import java.util.\*;  
public class Task1 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 double a, b, c;  
 double min = 0, max = 0;  
 System.*out*.println("Enter a-digit");  
 a = scan.nextDouble();  
 System.*out*.println("Enter b-digit");  
 b = scan.nextDouble();  
 System.*out*.println("Enter c-digit");  
 c = scan.nextDouble();  
 if (a > b) {  
 max = a;  
 min = b;  
 if (b > c) {  
 min = c;  
 } else if (a < c) {  
 max = c;  
 }  
 }  
 else if (a < b) {  
 max = b;  
 min = a;  
 if (a > c) {  
 min = c;  
 } else if (b < c) {  
 max = c;  
 }  
 }  
 System.*out*.println("Min = "+min);  
 System.*out*.println("Max = "+max);  
 }  
}

## 1.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных.

Через функции МИН и МАКС вычислены соотв. числа из области оценивания.

На рис. 1 представлен вид решения в MS Excel.



Рис. 1. Решение задачи в MS Excel

Далее в таблице 1 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Таблица 1

Тестирование работы программы и проверка результатов решения

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

Сравнение решения задачи с использованием двух прикладных пакетов показала, что решения задачи в Java и MS Excel совпадает. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи не выявлена.

# Задание 2

## 2.1. Постановка задачи

Написать программу, выводящую на экран те числа, которые больше  и сумма цифр которых больше . Изначально задается три трехзначных числа.

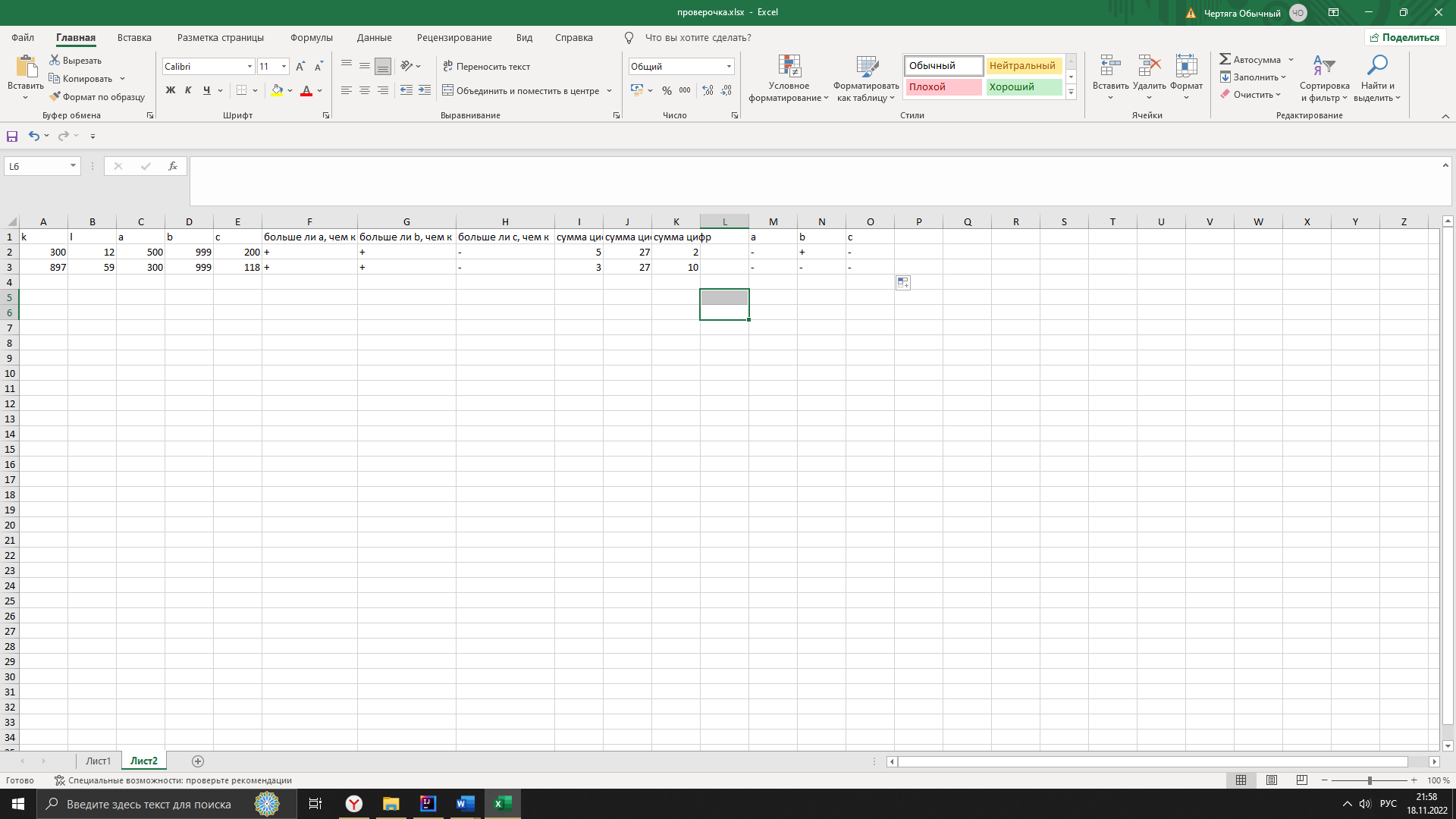
## 2.2. Решение задачи, код программы

import java.util.\*;  
  
public class Task2 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 int a, b, c, k, l;  
 System.*out*.println("Enter k");  
 k = scan.nextInt();  
 System.*out*.println("Enter l");  
 l = scan.nextInt();  
 System.*out*.println("Enter a");  
 a = scan.nextInt();  
 System.*out*.println("Enter b");  
 b = scan.nextInt();  
 System.*out*.println("Enter c");  
 c = scan.nextInt();  
 int suma = 0;  
 int sumb = 0;  
 int sumc = 0;  
 if (a > k) {  
 int anew = a;  
 while (anew > 0) {  
 int chisloa = anew % 10;  
 anew = anew / 10;  
 suma = suma + chisloa;  
 }  
 }  
 if (b > k) {  
 int bnew = b;  
 while (bnew > 0) {  
 int chislb = bnew % 10;  
 bnew = bnew / 10;  
 sumb = sumb + chislb;  
 }  
 }  
 if (c > k) {  
 int cnew = c;  
 while (cnew > 0) {  
 int chislc = cnew % 10;  
 cnew = cnew / 10;  
 sumc = sumc + chislc;  
 }  
 }  
 if (suma > l) {  
 System.*out*.println (a);  
 }  
 if (suma < l) {  
 System.*out*.println("Number 'a' is too small");  
 }  
 if (sumb > l) {  
 System.*out*.println (b);  
 }  
 if (sumb < l) {  
 System.*out*.println("Number 'b' is too small");  
 }  
 if (sumc > l) {  
 System.*out*.println (c);}  
 if (sumc < l) {  
 System.*out*.println("Number 'c' is too small");  
 }  
 }  
}

## 2.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных.

Рис. 2. Решение задачи в MS Excel



Пример вычисления суммы цифр на литере «а» :

=ОТБР(ОСТАТ(C2;10))+ОТБР(ОСТАТ(C2;100)/10)+ОТБР(C2/100)

Пример сравнения числа l и заданного:

=ЕСЛИ(J2>$B$2;"+";"-")

Далее в таблице 2 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Таблица 2

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Решение MS Excel |
| 1 |  | Проверка на рис. 2 |
| 2 |  | Проверка на рис. 2 |

Сравнение решения задачи с использованием двух прикладных пакетов показала, что решения задачи в Java и MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи не выявлена.

# Задание 3

## 3.1. Постановка задачи

Найти **,  с шагом 0,5.

## 3.2. Решение задачи, код программы

import static java.lang.Math.\*;  
  
public class Task3Lab5 {  
 public static void main(String[] args) {  
 double shag = 0.5;  
 double x = -3.5;  
 while (x < 3) {  
 x = x+shag;  
 double f;  
 if (x >= 0) {  
 f = *pow*(x, 2) - 5;  
 }  
 else {  
 f = 5 - (x\*x);  
 }  
 System.*out*.println("f=" +f);  
 }  
 }  
 }

## 3.3. Тестирование работы программы с проверкой

Для проверки подзадачи «Правильный треугольник» в задаче 3 в MS Excel создана таблица данных (рисунок 3).

Рис. 3. Решение задачи в MS Excel

Функция вычисления в эксель:

=ЕСЛИ(D2>=0; (D2^2-5); (5-D2^2))

Далее в таблице 3 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Таблица 3

Тестирование работы программы и проверка результатов решения

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

Сравнение решения задачи с использованием двух прикладных пакетов показала, что решения задачи в Java и MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи не выявлена.

# Задание 4

## 4.1. Постановка задачи

Найти 

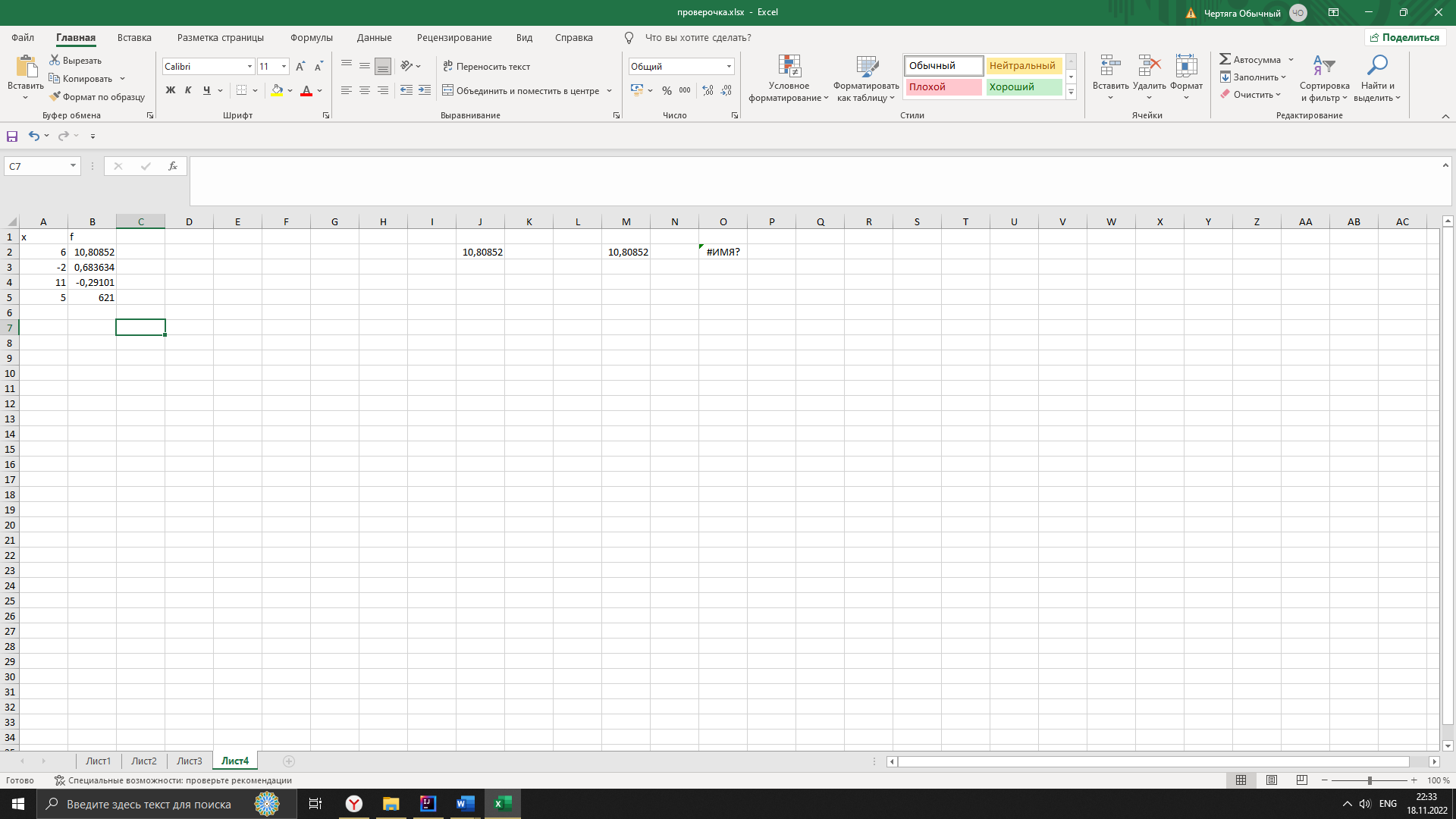
## 4.2. Решение задачи, код программы

import java.util.\*;  
import static java.lang.Math.\*;  
public class Task4Lab5 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Pls,enter an 'x-number'");  
 double x = scan.nextDouble();  
 double y=0;  
 if (x<5) {  
 y= *pow*(*sin*(2),*pow*(x,2));  
 }  
 else if (x>5 && x<10) {  
 y = (*cos*(2\*x))/*sin*(x);  
 }  
 else if (x>10) {  
 y = *tan*(x+5\**cos*(*PI*));  
 }  
 else if (x==5 || x==10) {  
 y = 1-x+5\**pow*(x,3);  
 }  
 System.*out*.println("f="+y);  
 }  
}

## 4.3. Тестирование работы программы с проверкой

Для проверки задачи 4 в MS Excel создана таблица данных (рисунок 4).

Рис. 4. Решение задачи в MS Excel



=ЕСЛИ(A2<5;(SIN(2))^(A2^2);ЕСЛИ(И(A2>5;A2<10);(COS(2\*A2))/((SIN(A2))^2);ЕСЛИ(A2>10;TAN(A2+5\*COS(ПИ()));ЕСЛИ(ИЛИ(A2=5; A2=10 );1-A2+5\*(x^3)))))

Далее в таблице 4 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Таблица 4

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Решение MS Excel |
| 1 |  | Результат на рисунке 4 |
| 2 |  | Результат на рисунке 4 |
| 3 |  | Результат на рисунке 4 |
| 4 |  | Результат на рисунке 4 |

Сравнение решения задачи с использованием двух прикладных пакетов показала, что решения задачи в Java и MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи не выявлена.

# Задание 5

## 5.1. Постановка задачи

Даны две фигуры (см. рис. 1), организовать программу, которая дает ответ, попала ли произвольно введенная точка в одну из фигур и если попала, то в какую.



## 5.2. Решение задачи, код программы

import java.util.\*;  
import static java.lang.Math.\*;  
public class Task5Lab5 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Enter the x-coordinate");  
 double x = scan.nextDouble();  
 System.*out*.println("Enter the y-coordinate");  
 double y = scan.nextDouble();  
 if (*figure1*(x, y) || *figure2*(x, y) || *figure3*(x, y) || *figure4*(x, y) || *figure5*(x, y) || *figure6*(x, y) ||  
 *figure7*(x, y) || *figure8*(x, y) || *figure10*(x, y)) {  
 if (*figure9*(x, y)) {  
 System.*out*.println("Miss");  
 }  
 else {System.*out*.println("Hit 1");  
 }  
 }  
 else if (*figure2\_4*(x, y) || *figure2\_5*(x, y) || *figure2\_6*(x, y) || *figure2\_7*(x, y) || *figure2\_7*(x, y) ||  
 *figure2\_8*(x, y) || *figure2\_9*(x, y) ||*figure2\_10*(x, y)) {  
 if (*figure2\_1*(x, y) || *figure2\_2*(x, y) || *figure2\_3*(x, y)) {  
 System.*out*.println("Miss");  
 } else System.*out*.println("Hit 2");  
 }  
 else {  
 System.*out*.println("Miss");  
 }  
}  
  
  
 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 figure1(double x, double y) {  
 if (y >= *k*(-3, -1, 0, -3) \* x + *b*(-3, -1, 0, -3)  
 && y <= *k*(-3, -1, -1, 0) \* x + *b*(-3, -1, -1, 0)  
 && y <= *k*(0, -3, -1, 0) \* x + *b*(0, -3, -1, 0)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean figure2(double x, double y) {  
 if (y >= *k*(-3, -1, -1, 0) \* x + *b*(-3, -1, -1, 0)  
 && y <= *k*(-3, -1, -2, 0) \* x + *b*(-3, -1, -2, 0)  
 && y <= *k*(-2, 0, -1, 0) \* x + *b*(-2, 0, -1, 0)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean figure3(double x, double y) {  
 if (y <= *k*(-2, 0, -1, 1) \* x + *b*(-2, 0, -1, 1)  
 && y >= *k*(-2, 0, -1, 0) \* x + *b*(-2, 0, -1, 0)  
 && y >= *k*(-1, 1, -1, 0) \* x + *b*(-1, 1, -1, 0)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean figure4(double x, double y) {  
 if (y <= *k*(-1, 0, -1, 1) \* x + *b*(-1, 0, -1, 1)  
 && y <= *k*(-1, 1, 1.5, 1.25) \* x + *b*(-1, 1, 1.5, 1.25)  
 && y >= *k*(-1, 0, 1.5, 1.25) \* x + *b*(-1, 0, 1.5, 1.25)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean figure5(double x, double y) {  
 if (y <= *k*(-1, 1, 1, 3) \* x + *b*(-1, 1, 1, 3)  
 && y >= *k*(-1, 1, 1.5, 1.25) \* x + *b*(-1, 1, 1.5, 1.25)  
 && y <= *k*(1, 3, 1.5, 1.25) \* x + *b*(1, 3, 1.5, 1.25)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean figure6(double x, double y) {  
 if (y <= *k*(1, 3, 3, 4) \* x + *b*(1, 3, 3, 4)  
 && y >= *k*(1, 3, 1.5, 1.25) \* x + *b*(1, 3, 1.5, 1.25)  
 && y >= *k*(1.5, 1.25, 3, 4) \* x + *b*(1.5, 1.25, 3, 4)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure7(double x, double y) {  
 if (y <= *k*(-1, 1, 0, 5) \* x + *b*(-1, 1, 0, 5)  
 && y >= *k*(-1, 1, 1, 3) \* x + *b*(-1, 1, 1, 3)  
 && y <= *k*(0, 5, 1, 3) \* x + *b*(0, 5, 1, 3)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure8(double x, double y) {  
 if (y >= *k*(-1, 1, -2, 3) \* x + *b*(-1, 1, -2, 3)  
 && y >= *k*(-1, 1, 0, 5) \* x + *b*(-1, 1, 0, 5)  
 && y <= *k*(-2, 3, 0, 5) \* x + *b*(-2, 3, 0, 5)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure9(double x, double y) {  
 if (y <= *k*(-2, 3, -1, 5) \* x + *b*(-2, 3, -1, 5)  
 && y >= *k*(-2, 3, 0, 5) \* x + *b*(-2, 3, 0, 5)  
 && y <= *k*(-1, 5, 0, 5) \* x + *b*(-1, 5, 0, 5)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure10(double x, double y) {  
 if (*pow*((y-3),2)+*pow*((x+1),2)<=*pow*(2,2)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure2\_1(double x, double y) {  
 if (y >= *k*(4, -2, 2, -1) \* x + *b*(4, -2, 2, -1)  
 && y >= *k*(2, -1, 2, 1) \* x + *b*(2, -1, 2, 1)  
 && y <= *k*(4, -2, 2, 1) \* x + *b*(4, -2, 2, 1)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure2\_2(double x, double y) {  
 if (y >= *k*(5, -2, 2, 1) \* x + *b*(5, -2, 2, 1)  
 && y <= *k*(2, 1, 4, 3) \* x + *b*(2, 1, 4, 3)  
 && y <= *k*(5, -2, 4, 3) \* x + *b*(5, -2, 4, 3)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure2\_3(double x, double y) {  
 if (y <= *k*(4, 3, 5, 5) \* x + *b*(4, 3, 5, 5)  
 && y >= *k*(6, 0, 4, 3) \* x + *b*(6, 0, 4, 3)  
 && y <= *k*(6, 0, 5, 5) \* x + *b*(6, 0, 5, 5)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure2\_4(double x, double y) {  
 if (y >= *k*(5, -2, 4, 3) \* x + *b*(5, -2, 4, 3)  
 && y <= *k*(6, 0, 4, 3) \* x + *b*(6, 0, 4, 3)  
 && y >= *k*(5, -2, 6, 0) \* x + *b*(6, 0, 5, 5)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure2\_5(double x, double y) {  
 if (y >= *k*(6, 0, 7, 0) \* x + *b*(6, 0, 7, 0)  
 && y >= *k*(6, 0, 5, 5) \* x + *b*(6, 0, 5, 5)  
 && y <= *k*(7, 0, 5, 5) \* x + *b*(7, 0, 5, 5)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure2\_6(double x, double y) {  
 if (y <= *k*(5, -2, 6, 0) \* x + *b*(6, -1, 6, 0)  
 && y >= *k*(5, -2, 7, 0) \* x + *b*(6, -1, 7, 0)  
 && y <= *k*(6, 0, 7, 0) \* x + *b*(6, 0, 7, 0)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure2\_7(double x, double y) {  
 if (y <= *k*(4, -3, 6, -1) \* x + *b*(4, -3, 6, -1)  
 && y >= *k*(4, -3, 7, -4) \* x + *b*(4, -3, 7, -4)  
 && y <= *k*(6, -1, 7, -4) \* x + *b*(6, -1, 7, -4)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure2\_8(double x, double y) {  
 if (y <= *k*(2, -4, 4, -3) \* x + *b*(2, -4, 4, -3)  
 && y >= *k*(2, -4, 7, -4) \* x + *b*(2, -4, 7, -4)  
 && y <= *k*(4, -3, 7, -4) \* x + *b*(4, -3, 7, -4)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure2\_9(double x, double y) {  
 if (y >= *k*(4, -2, 5, -2) \* x + *b*(4, -2, 5, -2)  
 && y >= *k*(4, -2, 2, 1) \* x + *b*(4, -2, 2, 1)  
 && y <= *k*(2, 1, 5, -2) \* x + *b*(2, 1, 5, -2)) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean figure2\_10(double x, double y) {  
 if (*pow*((y+1),2)+*pow*((x-4),2)<=*pow*(2,2)) {  
 return true;  
 }  
 return false;  
 }  
}

## 5.3. Тестирование работы программы с проверкой

Для проверки задачи 5 в MS Excel создана таблица данных (рисунок 5).

Рис. 5. Решение задачи в MS Excel

Далее в таблице 5 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Таблица 5

Тестирование работы программы и проверка результатов решения

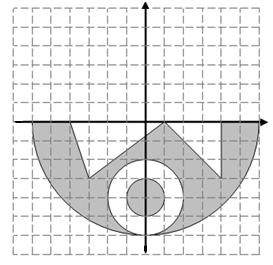
|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Графическое отображение |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |

Сравнение решения задачи с использованием двух прикладных пакетов показала, что решения задачи в Java и MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи не выявлена.

# Задание 6

## 6.1. Постановка задачи

6. Пользователь вводит координаты точки с клавиатуры, определить попала ли точка в закрашенную часть фигуры, вывести сообщение на экран (см. рис. 2). Одно деление сетки соответствует 1.



## 6.2. Решение задачи, код программы

import java.util.\*;  
import static java.lang.Math.\*;  
  
public class Task6Lab5 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Pls enter x-coordinate");  
 Double x = scan.nextDouble();  
 System.*out*.println("Pls enter y-coordinate");  
 Double y = scan.nextDouble();  
 if (6 <= (*pow*((x - 0), 2) + *pow*((y - 0), 2)) & y <= 0) {  
 if (*tri1*(x, y) || *tri2*(x, y)) {  
 System.*out*.println("Miss");  
 }  
 else if (4 <= (*pow*((x - 0), 2) + *pow*((y + 4), 2))) {  
 if (1 >= (*pow*((x - 0), 2) + *pow*((y + 4), 2))) {  
 System.*out*.println("Mel");  
 System.*out*.println("Hit");  
 }  
 else {  
 System.*out*.println("Miss");  
 System.*out*.println("Krup");  
 }  
 }  
 else {  
 System.*out*.println("Hit");  
 System.*out*.println("Main");  
 }  
 }  
 else {  
 System.*out*.println("Miss");  
 }  
 }  
 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 tri1 (double x, double y) {  
 if ((y >= *k*(-3, -3, -4, 0) \* x + *b*(-3, -3, -4, 0) &  
 (y >= *k*(1, 0, -3, -3) \* x + *b*(1, 0, -3, -3) &  
 (y <= *k*(1, 0, -4, 0) \* x + *b*(1, 0, -4, 0))))) {  
 return true;  
 }  
 return false;  
 }  
 public static boolean tri2 (double x, double y) {  
 if ((y <= *k*(1, 0, 4, 0) \* x + *b*(1, 0, 4, 0) &  
 (y >= *k*(4, 0, 4, -3) \* x + *b*(4, 0, 4, -3) &  
 (y >= *k*(1, 0, 4, -3) \* x + *b*(1, 0, 4, -3))))) {  
 return true;  
 }  
 return false;  
 }  
}

## 6.3. Тестирование работы программы с проверкой

Тестирование не требуется

Далее в таблице 6 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Таблица 6

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Графическое отображение |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |

Сравнение решения задачи с использованием двух прикладных пакетов показала, что решения задачи в Java и MS Excel совпадают. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи не выявлена.

# Задание 7

## 7.1. Постановка задачи

7. Составить две программы для определения функций с точками разрыва. Программа должна учитывать, что пользователь может ввести переменную, которая не удовлетворяет интервалу определения функции . Если, введенный аргумент из области определения функции не подходит из-за особенностей аргументов математических функций для корректного определения функции, то программа должна сообщить об этом.

|  |  |
| --- | --- |
|  |  |

## 7.2. Решение задачи, код программы

import java.util.\*;  
import static java.lang.Math.\*;  
  
public class Task7Lab5 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Pls, enter any digit from the array of [-4;4]");  
 double x = scan.nextDouble();  
 if (x >= -4 & x <= 0) {  
 do {  
 System.*out*.println("x=" + x);  
 System.*out*.println((*cos*(*sin*(*cos*((x \* x) + 2 \* x))) + *PI* \* *exp*(1)));  
 x = x + 1;  
 } while (x <= 0);  
 }  
 else if (x > 0 & x<=4) {  
 do {  
 System.*out*.println("x=" + x);  
 System.*out*.println(*pow*(*exp*(1), (*pow*(1 + ((2 \* x) / (1 + (x \* x))), 1 / 2d))));  
 x = x + 1;  
 } while (x<=4);  
 }  
 else {  
 System.*out*.println("Pls, enter the meaning from the array");  
 }  
 }  
}

Part2

import java.util.Scanner;  
import static java.lang.Math.\*;  
  
public class Task7Lab5prt2 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Pls, enter any digit from the array of [-4;4]");  
 double x = scan.nextDouble();  
 if (x >= -4 & x < 0) {  
 do {  
 System.*out*.println("x=" + x);  
 System.*out*.println(*tan*(x) + *cos*(*pow*((1 + x \* x), 1 / 2d)));  
 x = x + 1;  
 } while (x < 0);  
 }  
  
 else if (x>=0 & x<=1) {  
 do {  
 System.*out*.println("x=" + x);  
 System.*out*.println(2 \* *pow*(x, 6) + *pow*(x, 4) - (2 \* *cos*(x) \* *pow*(*exp*(1), (-2 \* x))));  
 x = x + 1;  
 } while (x <= 1);  
 }  
 else if (x > 1) {  
 do {  
 System.*out*.println("x=" + x);  
 System.*out*.println((2 \* *sin*(*cos*(x))) - ((x \* x \* x + x) / (2 \* *cos*(x + 1) + 2)));  
 x= x +1;  
 } while (x<=4);  
 }  
 else {  
 System.*out*.println("Pls, enter the meaning from the array");  
 }  
 }  
}

## 7.3. Тестирование работы программы с проверкой

Для проверки задачи 7 в MS Excel создана таблица данных (рисунок 4).

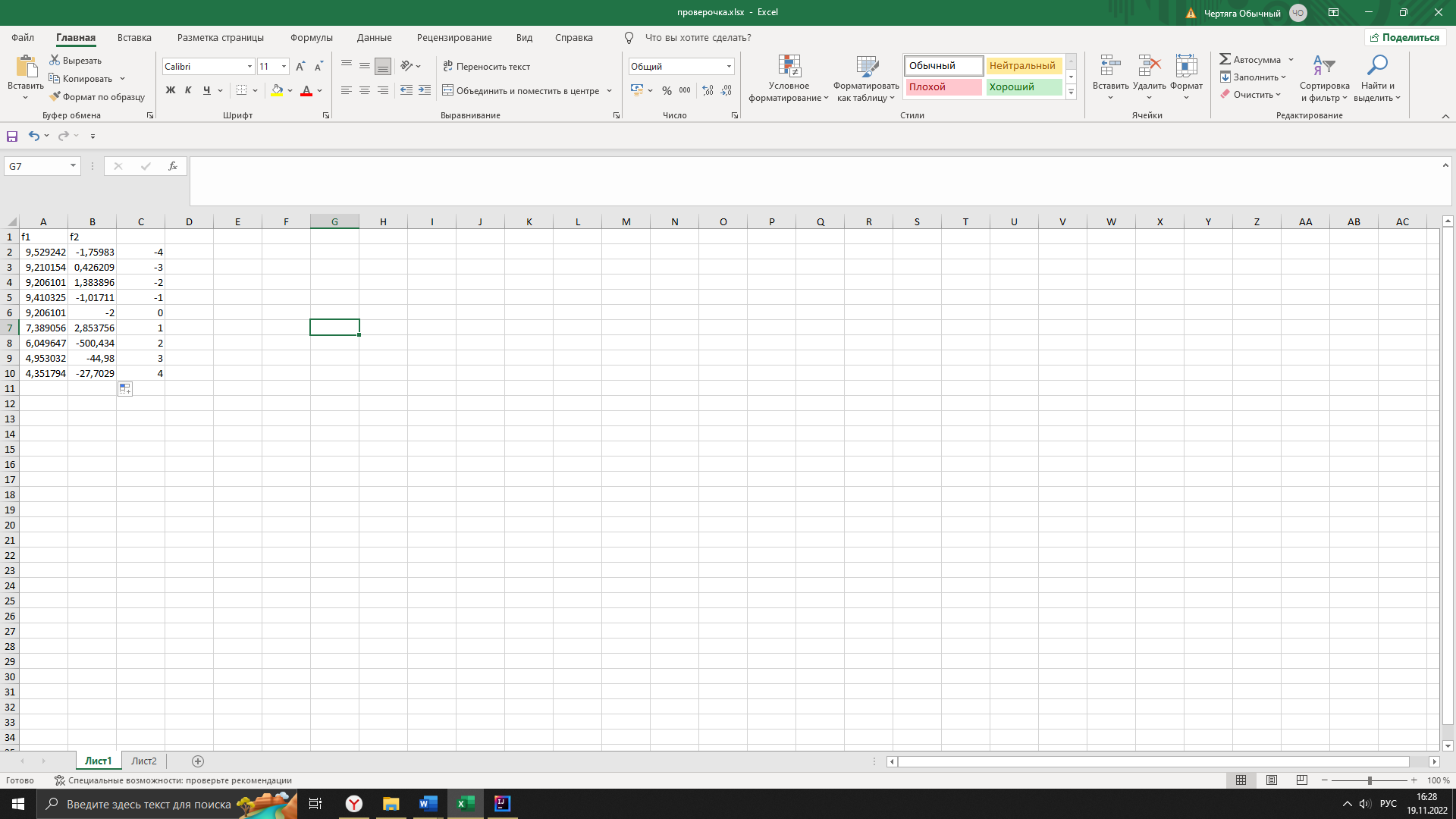


Рис. 7. Решение задачи в MS Excel

1 часть

=ЕСЛИ(C2<=0;COS(SIN(COS(C2^2+2\*C2)))+ПИ()\*EXP(1);ЕСЛИ(C2>0; EXP(1)^(1+((2\*C2)/(1+C2^2)))))

2 часть

=ЕСЛИ(C2<0; TAN(C2)+COS((1+C2^2)^1/2);ЕСЛИ(И(C2>=0;C2<=1);2\*C2^6+C2^4-2\*COS(C2)\*EXP(1)^(-2\*C2);ЕСЛИ(C2>1;2\*SIN(COS(C2))-((C2^3+C2)/(2\*COS(C2+1)+2)))))\

Далее в таблице 7 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Таблица 7

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Решение MS Excel |
| 1 |  | Результат на рисунке 7 |
| 2 |  | Результат на рисунке 7 |
| 3 |  | Результат на рисунке 7 |
| 4 |  | Результат на рисунке 7 |
| 5 |  | Результат на рисунке 7 |

Сравнение решения задачи с использованием двух прикладных пакетов показала, что решения задачи в Java и MS Excel совпадают во всех вариантах, кроме 3. Имеются расхождения в ответах, вероятно это связано с расхождением способов вычисления функций в двух разных программах. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлена.