Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 6

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. 1. Найти сумму первых N членов ряда и найти сумму членов ряда, которые меньше заданного с клавиатуры числа M:

**

## 1.2. Решение задачи, код программы

import java.util.\*;  
import static java.lang.Math.\*;  
public class Tsk1Lab6 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Pls, enter the required number of elements");  
 double N = scan.nextDouble();  
 System.*out*.println("Pls, enter the required value of M-number");  
 double M = scan.nextDouble();  
 double x = *sin*(1), sumM=0, S = 0;  
 int i;  
 for (i=1;i<=N;i++) {  
 System.*out*.println("The number of the iteration is " +i);  
 S = S + 1/x;  
 System.*out*.println("The sum of the first elements ="+S);  
 if (1/x<M) {  
 sumM = sumM + 1 / x;  
 System.*out*.println("The sum of the M-elements = " + sumM);  
 }  
 x = *sin*(x);  
 }  
 }  
}

## 1.3. Тестирование работы программы с проверкой

A2=СУММЕСЛИ(B2:B3; "<10")

C2 = СУММ (B2:B3)

ITERATION1=1/(SIN(F1))  
ITERATION2=1/(SIN(F2))

Для проверки задачи в MS Excel создана таблица данных.

На рис. 1 представлен вид решения в MS Excel.

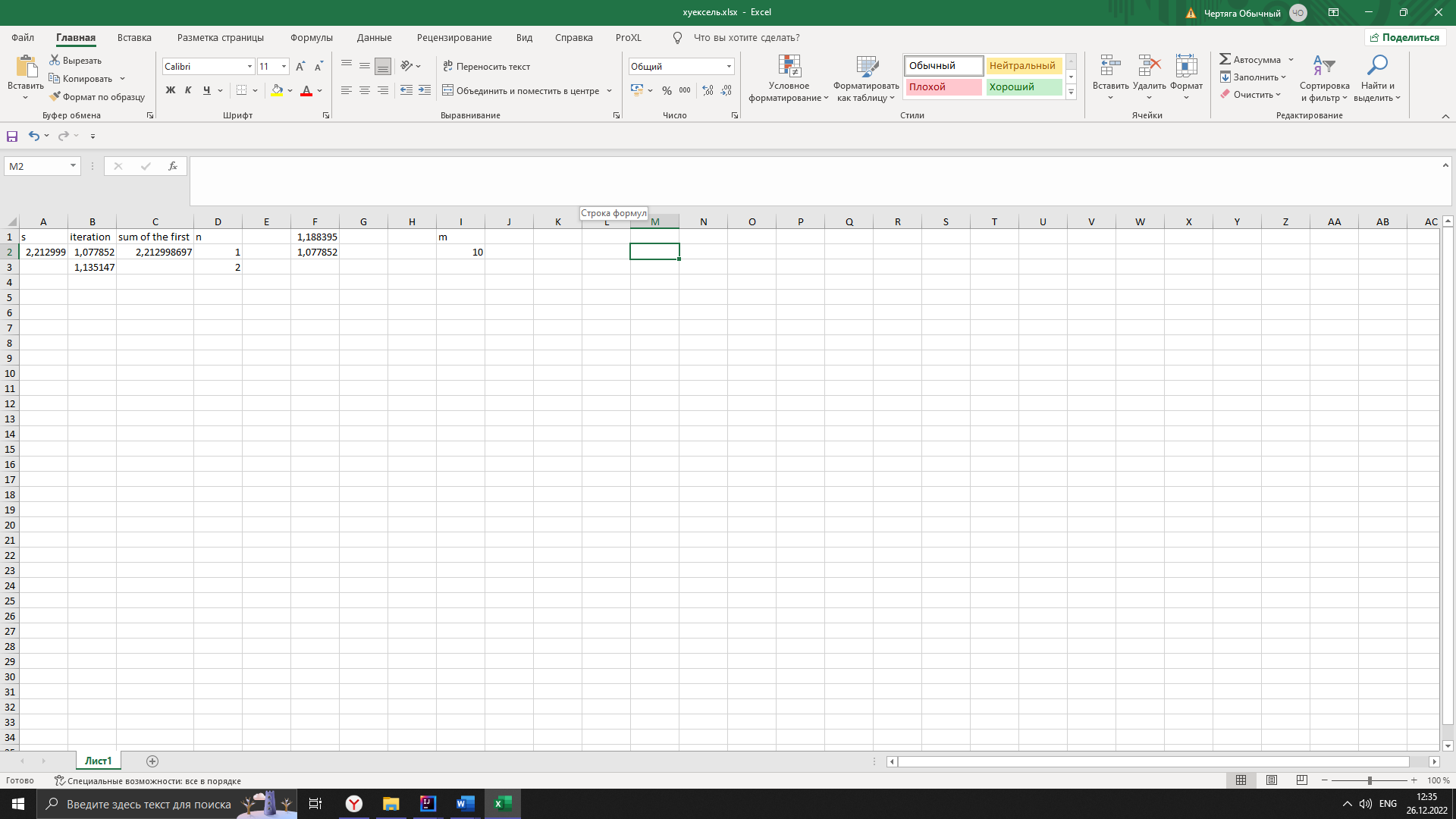


Рис. 1. Решение задачи в MS Excel

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

Ежемесячная стипендия студента составляет A руб., а расходы на проживание превышают ее и составляют В руб. в месяц. Рост цен ежемесячно увеличивает расходы на 3% по нечетным месяцам и на 5% по четным. Определить, какую нужно иметь сумму денег, чтобы прожить N месяцев и сколько студенту необходимо получать дополнительных денег от родителей.

## 2.2. Решение задачи, код программы

import java.util.\*;  
public class Tsk2Lab6 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 double A,B,S=0,D;  
 int N;  
 System.*out*.println("Pls,enter your bursary");  
 A = scan.nextDouble();  
 System.*out*.println("Pls,enter your annual expenses(That number must be bigger than bursary)");  
 B = scan.nextDouble();  
 System.*out*.println("Pls,enter the quantity of months");  
 N = scan.nextInt();  
 double Sa = A;  
 int i;  
 for (i=1;i<=N;i++) {  
 if (i==1) {}  
 else if ( i%2 ==0 ) {  
 double per5 = (B\*5)/100;  
 B = B+per5;  
 Sa=Sa+A;  
 }  
 else {  
 double per3 = (B \* 3) / 100;  
 B = B + per3;  
 Sa=Sa+A;  
 }  
 S = S+B;  
 }  
 D = S-Sa;  
 System.*out*.printf("Full sum for " +N+ " months is %.3f \n", S);  
 System.*out*.printf("Additional money needed = %.3f \n", D);  
 }  
}

## 2.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных.

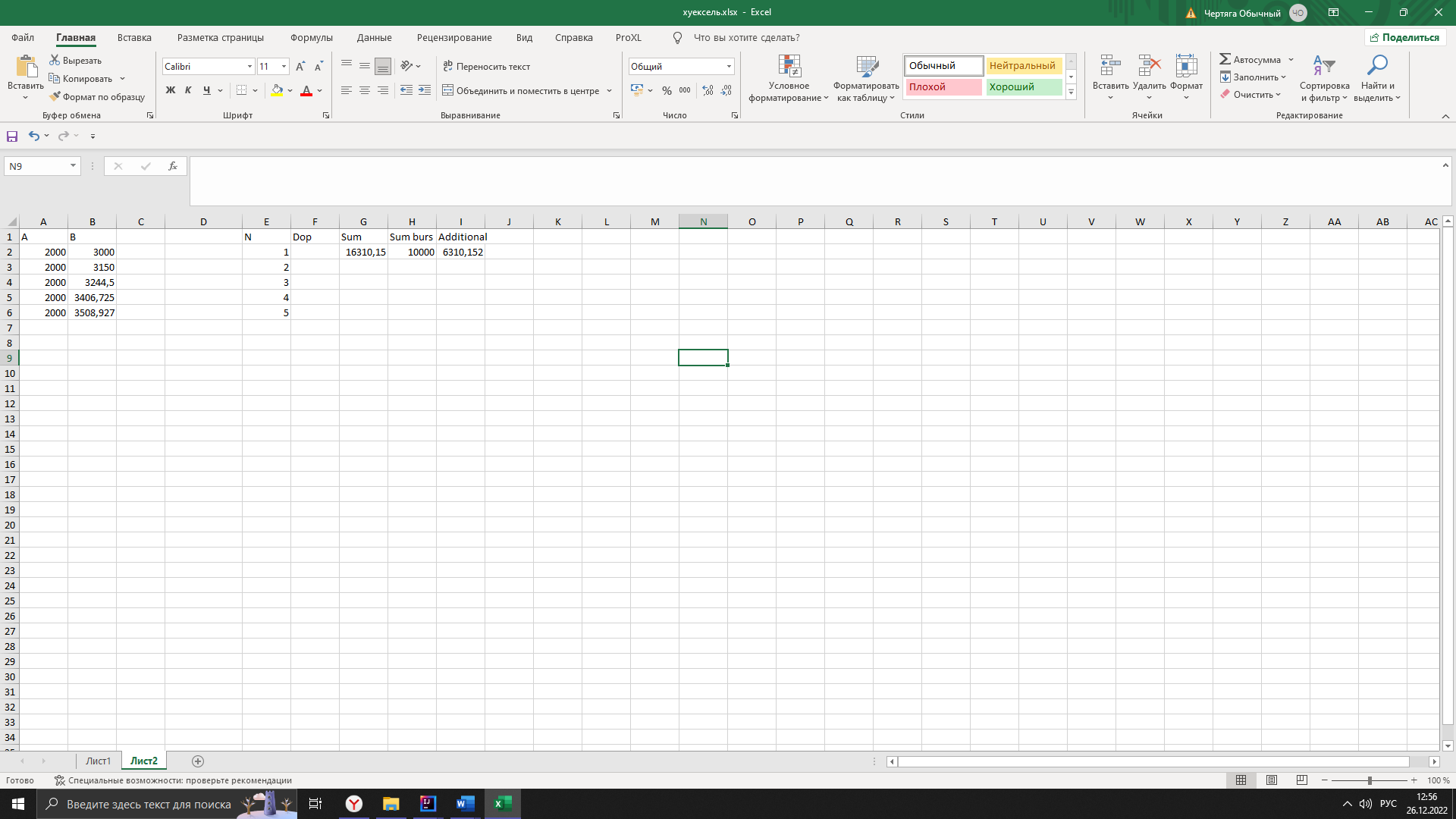


Рис. 2. Решение задачи в MS Excel

В3=ЕСЛИ(ОСТАТ(E3;2)=0;B2+(B2\*5/100);B2+(B2\*3/100))

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. |

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

3. Написать программы, которые вычисляют выражения:

, , 

## 3.2. Решение задачи, код программы

import java.util.\*;  
import static java.lang.Math.\*;  
public class Tsk3Lab6 {  
 public static void main(String[] args) {  
 System.*out*.println("Pick a number for the required part of the task, from 1 to 3");  
 Scanner scan = new Scanner(System.*in*);  
 int a =scan.nextInt();  
 if (a==1) {  
 System.*out*.println(*v1*());  
 }  
 else if (a==2) {  
 System.*out*.println(*v2*());  
 }  
 else if (a==3) {  
 System.*out*.println(*v3*());  
 }  
 else {  
 System.*out*.println("Pls restart");  
 }  
 }  
 public static String v1() {  
 int j, i;  
 double Sj=0, Si=0;  
 for (i=1;i<=8;i++) {  
 for (j=1;j<=8;j++) {  
 if (j!=i) {  
 Sj += (j \* j);  
 }  
 Sj=Sj+0;  
 }  
 Si+=Sj;  
 Sj=0;  
 System.*out*.println(Si);  
 }  
 return "end";  
 }  
 public static String v2() {  
 int j, i;  
 double Pj = 1, Si = 0;  
 for (i = 1; i <= 8; i++) {  
 for (j = 1; j <= 3; j++) {  
 Pj\*= ((*pow*(j, 3)) + (i \* i));  
 }  
 Si += Pj;  
 Pj=1;  
 System.*out*.println(Si);  
 }  
 return "end";  
 }  
 public static String v3() {  
 int j, i, k;  
 double Pk = 1, Si = 0, Sj=0;  
 for (i = 1; i <= 8; i++) {  
 for (j = 1; j <= i; j++) {  
 for (k=1;k<=2\*i;k++) {  
 Pk \*= ((2\*j)+(3\*i)-k);  
 }  
 Sj+=Pk;  
 Pk=1;  
 }  
 Si+=Sj;  
 Sj=0;  
 System.*out*.println(Si);  
 }  
 return "end";  
 }  
}

## 3.3. Тестирование работы программы с проверкой

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

Функция вычисления в эксель:

1)B2=ЕСЛИ(B$1<>$A2;B$1\*B$1;0)

J10=СУММ(J2:J9)

2)B14=(B$13^3+$A14^2)

3)N17=(2\*$M17+3\*$L$17-N$16)

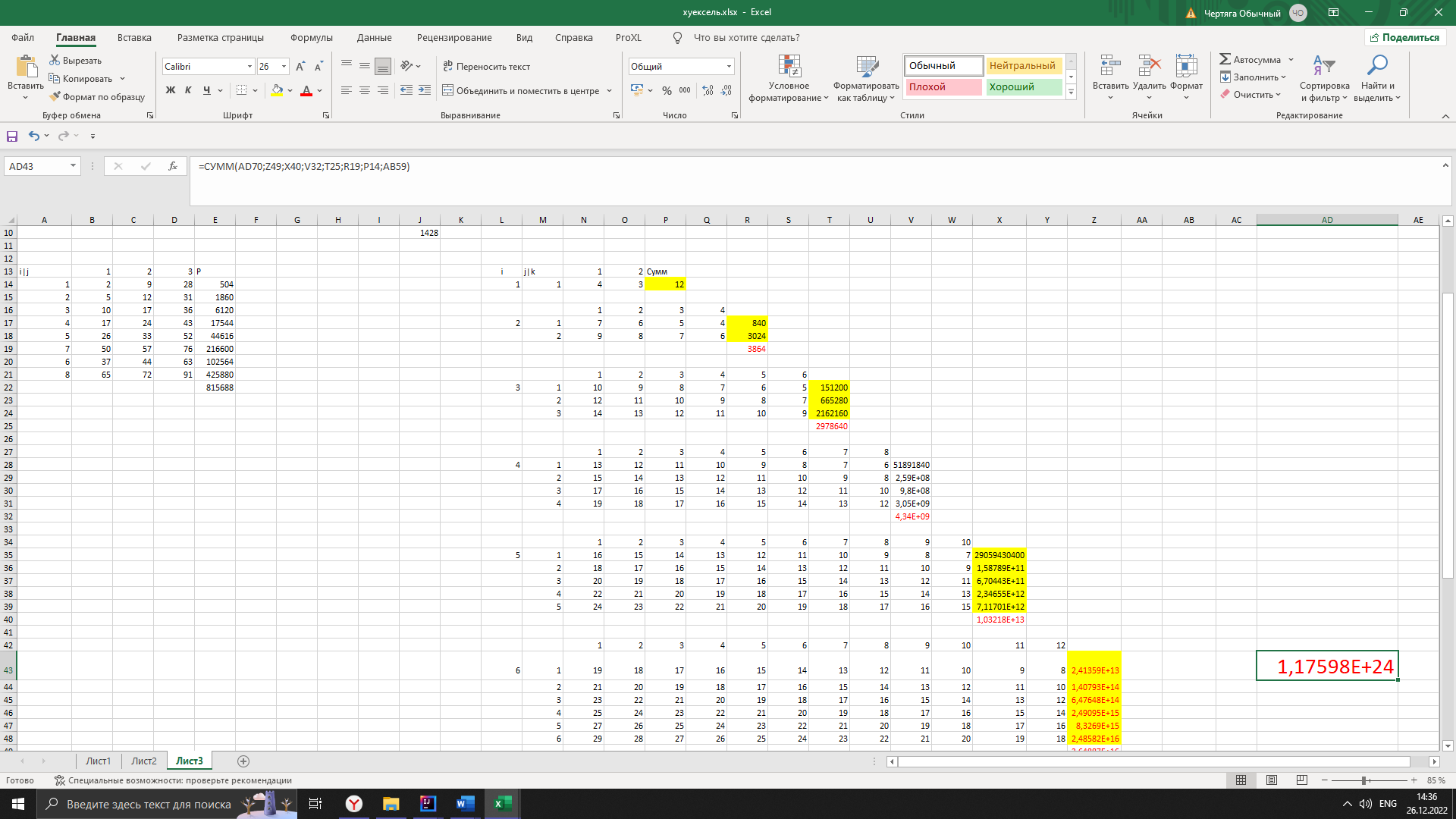


Рис 3 – Решение 3 подзадачи

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 |  |  |
| 2 |  |  |
| 3 |  |  |

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

Пользователь вводит целое десятичное число. Написать программу перевода целого десятичного числа из десятичной системы счисления в систему счисления с основанием, которое пользователь вводит с клавиатуры (вводим ограничения на системы счисления, в которых есть буквенное обозначение цифр, так же не может быть введено 0 и 1 как основание системы счисления). После перевода сделать проверку. Пользователь может выбирать систему счисления до бесконечности, т.е. необходимо предусмотреть внешний цикл с вопросом к пользователю о необходимости продолжать перевод из одной системы счисления в другую.

## 4.2. Решение задачи, код программы

import java.util.\*;  
public class Tsk4Lab6 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Welcome to number system converter");  
 for (; ; ) {  
 System.*out*.println("Do you want to proceed? Type 'y' or 'n'");  
 String yes = "y";  
 String no = "n";  
 String ans = scan.nextLine();  
 if (ans.equals(yes)) {  
 System.*out*.println("Pls enter decimal integer for transfer");  
 int a = scan.nextInt();  
 System.*out*.println("Pls enter number system, that is <=9 and isn't 0 or 1");  
 int b = scan.nextInt();  
 if (b <= 9 & b != 0 & b != 1) {  
 int i=0;  
 int[] ss = new int[1000];  
 while (a>0) {  
 ss[i]= a%b;  
 i++;  
 a/=b;  
 }  
 for (int j =i-1; j>=0;j--) {  
 System.*out*.print(ss[j]);  
 }  
 System.*out*.println();  
 }  
 else {  
 System.*out*.println("Something went wrong, pls follow the instructions");  
 }  
 }  
 else if (ans.equals(no)) {  
 break;  
 }  
 }  
 }  
}

## 

## 4.3. Тестирование работы программы

Таблица 4

Тестирование работы программы и проверка результатов решения

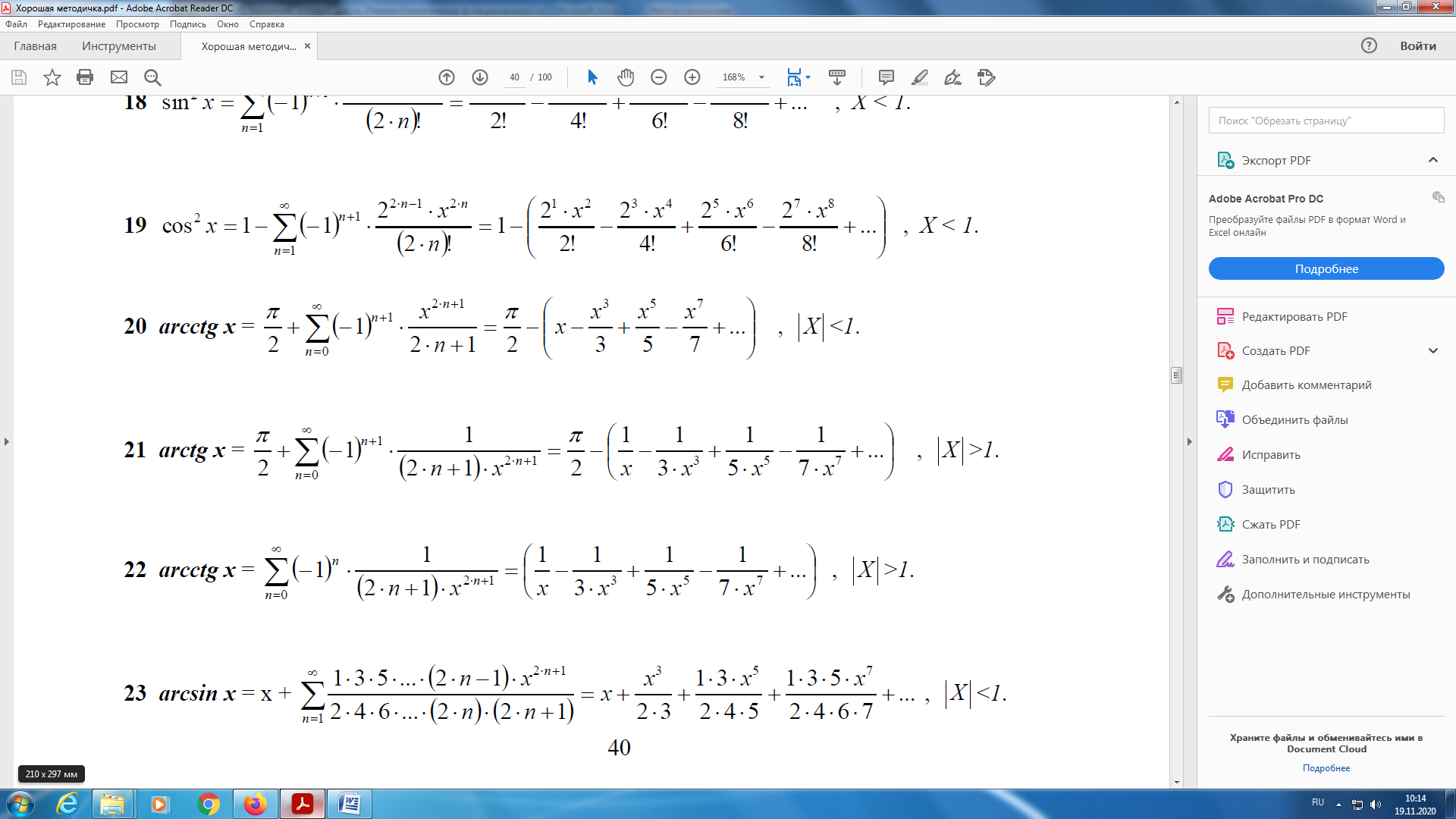
|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java |  |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

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

# Задание 5

## 5.1. Постановка задачи

Организовать цикл для нахождения функции , через ряд Маклорена с погрешностью 10-2 – 10-4, остановка итерационной процедуры , где  – это погрешность. Вывести значение суммы ряда и итоговую погрешность для всех вариантов остановки итерационной процедуры.



## 5.2. Решение задачи, код программы

import java.util.\*;  
import static java.lang.Math.\*;  
public class Tsk5Lab6 {  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 double e1=0.01, e2=0.001, e3=0.0001;  
 System.*out*.println("Pls,enter the |x|>1");  
 double x =scan.nextDouble();  
 if (*abs*(x)>1) {  
 System.*out*.println(*summ*(e1, x));  
 System.*out*.println(*summ*(e2, x));  
 System.*out*.println(*summ*(e3, x));  
 }  
 else {  
 System.*out*.println("Something went wrong,pls restart");  
 }  
  
 }  
 public static double summ(double e,double x) {  
 System.*out*.println("Inaccuracy = "+e);  
 int n=0;  
 double sum=0;  
 while ((*abs*(*atan*(x)-((*PI*/2)+sum)))>e) {  
 sum += (*pow*(-1,n+1))\*(1d/(((2\*n)+1)\*(*pow*(x,((2\*n)+1)))));  
 n++;  
 }  
 return *PI*/2+sum;  
 }  
}

## 5.3. Тестирование работы программы с проверкой

Таблица 5

Тестирование работы программы и проверка результатов решения

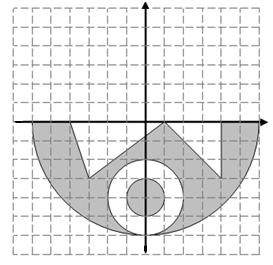
|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |

Неопределенностей при решении задачи не выявлена.

# Задание 6

## 6.1. Постановка задачи

Переделать программу (класс) задания 6 лабораторной работы 5 о попадании точки в область в класс без метода main с названием Oblast. Метод main заменить на метод Oblast c входными данными в виде координат произвольной точки пространства и выходными данными типа boolean (true – если точка попала в область, false – если точка не попала в область). Создать программу, взаимодействующую с классом Oblast (без использования наследования), в которой пользователь в цикле проверяет попадание точек в область до бесконечности, т.е. необходимо предусмотреть цикл с вопросом к пользователю о необходимости проверки точки.



## 6.2. Решение задачи, код программы

import java.util.Scanner;  
import static java.lang.Math.*pow*;  
public class Tsk6Lab6 {  
 public static class Oblast {  
 public static boolean dots(double x, double y) {  
 if ((*tri1*(x, y) || *tri2*(x, y)) ||  
 (4 > ((*pow*((x - 0), 2) + *pow*((y + 4), 2))) & (1 < (*pow*((x - 0), 2) + *pow*((y + 4), 2)))) ||  
 (36 >= (*pow*((x - 0), 2) + *pow*((y - 0), 2)) & y >= 0)) {  
 return false;  
 }  
 else return 36 >= (*pow*((x - 0), 2) + *pow*((y - 0), 2)) & y <= 0;  
 }  
 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) {  
 return 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)));  
 }  
  
 public static boolean tri2(double x, double y) {  
 return y > *k*(4, -3, 1, 0) \* x + *b*(4, -3, 1, 0) &  
 (x < 4) & (x > 1);  
 }  
 }  
 public static void main(String[] args) {  
 Scanner scan = new Scanner(System.*in*);  
 boolean ans = *ans*();  
 while (ans) {  
 System.*out*.println("Pls enter x-coordinate");  
 double x = scan.nextDouble();  
 System.*out*.println("Pls enter y-coordinate");  
 double y = scan.nextDouble();  
 System.*out*.println(Oblast.*dots*(x,y));  
 ans = *ans*();  
 }  
 }  
 public static boolean ans() {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Do you want to proceed checking? 'y' or 'n'");  
 String ans= scan.next();  
 String yes = "y";  
 return ans.equals(yes);  
 }  
}

## 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. Напечатать полную таблицу вычитания в виде:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 – 1 = 0 | 2 – 1 = 1 | ... | 9 – 1 = 8 |
| 1 – 2 = –1 | 2 – 2 = 0 | ... | 9 – 2 = 7 |
| ... | ... | ... | ... |
| 1 – 9 = –8 | 2 – 9 = –7 | ... | 9 – 9 = 0 |

## 7.2. Решение задачи, код программы

Таблица 7

Тестирование работы программы и проверка результатов решения

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

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.

# Задание 8

## 8.1. Постановка задачи

8. Переделать программу (класс) задания 1 лабораторной работы 4 о нахождении значений 2-х функций в класс без метода main с названием FunctionMy. Метод main заменить на метод FunctionMy c входными данными. Создать программу, взаимодействующую с классом FunctionMy, в которой пользователь в цикле находит сумму 10 значений функции изменяя только один параметр функции в цикле, остальные параметры, которые входя в формулу, считаются константами. Взаимодействие с классом FunctionMy сделать в виде наследования.

## 8.2. Решение задачи, код программы

import static java.lang.Math.\*;  
import java.util.Scanner;  
  
public class Tsk8Lab6 {  
 public static class FunctionMy {  
 public static double FunctionMy1(double y) {  
 double d = 1;  
 return *log*(d) + ((3.5 \* (d \* d) + 1) / (*cos*(2 \* y)));  
 }  
 public static double FunctionMy2(double y) {  
 double x = 1;  
 return (*sqrt*(((2+y)\*(2+y))+(*pow*(*sin*(y+5),1d/7))))  
 /(*log*(x + 1) - *pow*(y, 3));  
 }  
 }  
  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 int n = 0;  
 double sum = 0;  
 while (n < 10){  
 System.*out*.println("input y");  
 double y = sc.nextDouble();  
 sum += FunctionMy.*FunctionMy1*(y) + FunctionMy.*FunctionMy2*(y);  
 n++;  
 }  
 System.*out*.println(sum);  
 }  
}

Таблица 8

Тестирование работы программы и проверка результатов решения

|  |  |
| --- | --- |
| № п.п. | Решение Java |
| 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 выявлена.

# Задание 9

## 9.1. Постановка задачи

9. Написать программу вычисление корня р-й степени (степень вводиться с клавиатуры) в рамках итерационной процедуры . Для определения используется итерационная процедура на основе формулы Ньютона , , при этом . Остановка итерационной процедуры , где  – точность вычисления. В рамках программы определить число итераций, которые потребовались для отыскания корня р-й степени в рамках цикла с параметром для точности от 10-2 до 10-6, шаг 10-1. Организовать форматированный вывод результатов в виде: Точность Корень Число итераций.

## 9.2. Решение задачи, код программы

import java.util.Scanner;  
public class Tsk9Lab6 {  
 public static void main(String[] args) {  
 Scanner in=new Scanner(System.*in*);  
 double y0;  
 System.*out*.println("input p");  
 double p = in.nextDouble();  
 System.*out*.println("input x");  
 double x = in.nextDouble();  
 System.*out*.println("input y0");  
 while ((y0 = in.nextDouble())>= Math.*exp*(Math.*log*((x\*(p+1)))/p)){  
 System.*out*.println("wrong");  
 }  
 int n = 0;  
 double y;  
 for (double ep = 0.000001; ep <= 0.01; ep/=0.1){  
 y = 0;  
 while (Math.*abs*(y - y0) > ep){  
 y = (1/p)\*((p - 1)\*y0+(x/Math.*pow*(y0, p - 1)));  
 y0 = y;  
 n++;  
 }  
 if (Math.*abs*(y - y0)<=ep){  
 System.*out*.println("Точность " + ep + " Корень " + y + " Число итераций " + n);  
 }  
 }  
 }  
}

Тестирование работы программы и проверка результатов решения

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

# Задание 10

## 10.1. Постановка задачи

Разработать алгоритм приближённого вычисления площади криволинейной фигуры, ограниченной осью абсцисс, графиком заданной функции , и вертикальными прямыми, т.е. . Каждый отрезок функции представляется в виде прямоугольника c длиной отрезка (шагом)  (см. рис. ниже). Затем площадь под кривой вычисляется по формуле  – центральный прямоугольник, т.к. высота прямоугольника берется как значение функции в центре каждого прямоугольника. Вычислить значения площади под кривой при n равном 10, 100, 1000, 10000 в рамках цикла по n. Оценить погрешность решения при разных шагах по сравнению с точным аналитическим решением. Организовать форматированный вывод результатов в виде:

n h S Аналитическое решение Погрешность.

|  |  |
| --- | --- |
|  |  |
| а | б |

Рис. Разбиение площади под кривой на прямоугольники с шагом h:

а – общий вид; б – i-й центральный прямоугольник

## 10.2. Решение задачи, код программы

import static java.lang.Math.\*;  
public class Tsk10Lab6 {  
 public static void main(String[] args) {  
 double h;  
 double sum;  
 for (double n = 10; n <= 10000; n \*= 10) {  
 h = 1 / n;  
 sum = 0;  
 double a = 1;  
 for (int i = 1; i <= n; i++) {  
 a += h;  
 sum += (*pow*(a, a) + 10 \* *pow*(a, 2) - (a) \* *sin*(a)) \* h;  
 }  
 System.*out*.println("n: " + n + " h: " + h + " S: " + sum + " 23.94 " + (sum - 23.94));  
 }  
 }  
}

Таблица 10

Тестирование работы программы и проверка результатов решения

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

Неопределенностей при решении задачи выявлена.