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-21-1б Носов Д.О.

Проверил: ассистент каф. ВММБ Нетбай Г.В.

Пермь, 2022

**Содержание**

Оглавление

[Задание 1 3](#_Toc121434513)

[1.1. Постановка задачи 3](#_Toc121434514)

[1.2. Решение задачи, код программы 3](#_Toc121434515)

[1.3. Тестирование работы программы с проверкой 3](#_Toc121434516)

[Задание 2 5](#_Toc121434517)

[2.1. Постановка задачи 5](#_Toc121434518)

[2.2. Решение задачи, код программы 5](#_Toc121434519)

[2.3. Тестирование работы программы с проверкой 5](#_Toc121434520)

[Задание 3 7](#_Toc121434521)

[3.1. Постановка задачи 7](#_Toc121434522)

[3.2. Решение задачи, код программы 7](#_Toc121434523)

[3.3.1 Тестирование работы программы с проверкой 7](#_Toc121434524)

[3.3.2 Тестирование работы программы с проверкой 9](#_Toc121434525)

[3.3.3 Тестирование работы программы с проверкой 10](#_Toc121434526)

[Задание 4 12](#_Toc121434527)

[4.1. Постановка задачи 12](#_Toc121434528)

[4.2. Решение задачи, код программы 12](#_Toc121434529)

[4.3. Тестирование работы программы с проверкой 13](#_Toc121434530)

[Задание 6 15](#_Toc121434531)

[6.1. Постановка задачи 15](#_Toc121434532)

[6.2. Решение задачи, код программы 15](#_Toc121434533)

[6.3. Тестирование работы программы с проверкой 16](#_Toc121434534)

[Задание 7 19](#_Toc121434535)

[7.1. Постановка задачи 19](#_Toc121434536)

[7.2. Решение задачи, код программы 19](#_Toc121434537)

[7.3. Тестирование работы программы с проверкой 19](#_Toc121434538)

[Задание 8 21](#_Toc121434539)

[8.1. Постановка задачи 21](#_Toc121434540)

[8.2. Решение задачи, код программы 21](#_Toc121434541)

[8.3. Тестирование работы программы с проверкой 22](#_Toc121434542)

[Задание 9 23](#_Toc121434543)

[9.1. Постановка задачи 23](#_Toc121434544)

[9.2. Решение задачи, код программы 23](#_Toc121434545)

[9.3. Тестирование работы программы с проверкой 23](#_Toc121434546)

[Задание 10 27](#_Toc121434547)

[10.1. Постановка задачи 27](#_Toc121434548)

[10.2. Решение задачи, код программы 27](#_Toc121434549)

[10.3. Тестирование работы программы с проверкой 28](#_Toc121434550)

# Задание 1

## 1.1. Постановка задачи

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

**.

## 1.2. Решение задачи, код программы

i import java.util.Scanner;  
import static java.lang.Math.\*;  
  
public class zad1 {  
 public void zad1() {  
 Scanner scn = new Scanner(System.*in*);  
 System.*out*.println("Введите кол-во первых членов ряда:");  
 int N = scn.nextInt();  
 System.*out*.println("Введите сравниваемое число M:");  
 double M = scn.nextDouble();  
 System.*out*.println("Сумма первых "+N+" членов ряда равна "+*SumPervN*(N));  
 System.*out*.println("Сумма членов ряда < "+M+" равна "+*SumDoM*(M));  
 }  
 public static double SumPervN(int N){  
 double count = *cos*(1) / *sin*(2);  
 double sum = count;  
 for (int f = 2; f <= N; f++) {  
 count = *cos*(f) / *sin*(2 \* f);  
 sum += count;  
 }  
 if (1 <= N) {  
 return sum;  
 } else {  
 return 0;  
 }  
 }  
  
 public static double SumDoM(double M){  
 double count = *cos*(1) / *sin*(2);  
 double sum = 0;  
 for (int f = 2; count < M; f++) {  
 sum += count;  
 count = *cos*(f) / *sin*(2 \* f);  
 }  
 return sum;  
 }  
}

## 1.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в ячейки А:А записаны кол-во первых членов N, в ячейках B:В – n членов рядов, в ячейках C:С – члены ряда, в ячейку E1 – заданное с клавиатуры число M. В ячейку С1 записана формула для вычисления члена ряда и растянута вниз, в Е2 – сумма первых членов SumPervN, Е3 – сумма членов до M SumDoM.

Формулы для вычисления члена ряда, SumPervN, SumDoM:

С1) =COS(B1)/SIN(2\*B1);

Е2) =СУММЕСЛИ(A:A; ">= 1";C:C);

Е3) =СУММЕСЛИ(C:C; "< 4").

На рис. 1 представлен вид решения в MS Excel.

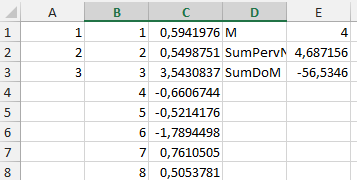


Рис. 1 Решение задачи в MS Excel

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 |  |  |
| 2 |  |  |

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 выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# Задание 2

## 2.1. Постановка задачи

Дано натуральное число. Определить, сколько раз в нем встречается максимальная цифра числа и вычислить факториал от этого количества (например, для числа 132 23 цифра 3 – максимальная, встречается 2 раза, 2! = 2, для числа 46 636 – ответ 6, для числа 12 345 – ответ 1).

## 2.2. Решение задачи, код программы

import java.util.Random;  
  
public class zad2 {  
 public void zad2() {  
 Random rnd = new Random();  
 int x = Math.*abs*(rnd.nextInt())+1;  
 int z = x;  
 int max = 0;  
 int num = 1;  
 while (z >= 1) {  
 int i = 0;  
 int c = z;  
 while (c >= 1) {  
 if (z % 10 == c % 10) {  
 i++;  
 }  
 c /= 10;  
 }  
 if (z % 10 > num) {  
 num = z % 10;  
 max = i;  
 }  
 z /= 10;  
 }  
 System.*out*.println("В числе " + x + " максимальная цифра " + num);  
 System.*out*.println("Цифра " + num + " встречается " + max + " раз(а)");  
 System.*out*.println("Факториал от " + max + " равен " + *Fact*(max));  
 }  
 public static int Fact(int b){  
 int i = 1;  
 for (int f = 2; f <= b; f++){  
 i \*= f;  
 }  
 return i;  
 }  
}

## 2.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в ячейку А2 записана данная переменная x, в ячейку B2 – наибольшая цифра данного числа, в ячейку C2 – кол-во наибольших цифр. В ячейки D2 записана формула для вычислений факториала количества наибольшей цифры.

Формулы для вычислений факториала количества наибольшей цифры:

D2) =ФАКТР(C2).

На рис. 2 представлен вид решения в MS Excel.

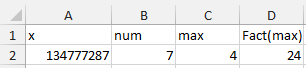


Рис. 2. Решение задачи в MS Excel

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 |  |  |
| 2 |  |  |

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 выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# Задание 3

## 3.1. Постановка задачи

Найти , , .

## 3.2. Решение задачи, код программы

import static java.lang.Math.\*;  
  
public class zad3 {  
 public static void SumSum() {  
 double sum1 = 0;  
 for (int i = 1; i <= 8; i++){  
 double sum2 = 0;  
 for (int j = 1; j<=i; j++){  
 sum2 += *pow*(*pow*(j,*cos*(i)) + *pow*(i,*cos*(j)),2);  
 }  
 sum1 += sum2;  
 }  
 System.*out*.println("Первое выражение равно "+sum1);  
 }  
 public static void ProdProd() {  
 double prod1 = 1;  
 for (int i = 1; i <= 5; i++){  
 double prod2 = 1;  
 for (int j = 1; j<=i; j++){  
 prod2 \*= *sin*(*pow*(j,*cos*(i)));  
 }  
 prod1 \*= prod2;  
 }  
 System.*out*.println("Второе выражение равно "+prod1);  
 }  
 public static void ProdSumSum() {  
 double prod = 1;  
 for (int i = 1; i <= 8; i++){  
 double sum1 = 0;  
 for (int j = i; j <= 2\*i - 1; j++){  
 double sum2 = 0;  
 for (int k = i+j; k<=2\*(i+j); k++){  
 sum2 += 2\**cos*(j) - 3\**log*(i + 0.5\*k);  
 }  
 sum1 += sum2;  
 }  
 prod \*= sum1;  
 }  
 System.*out*.println("Третье выражение равно "+prod);  
 }  
 public void zad3() {  
 *SumSum*();  
 *ProdProd*();  
 *ProdSumSum*();  
 }  
}

## 3.3.1 Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в ячейку А2:A9 записаны все значения i, в ячейки B2:В9 – все значения j. В ячейки C2 записана формулы для суммы с адаптацией цикла for в MS Exel. В ячейки C2:J9 записаны формулы суммы функций, С10:J10 – суммs значений в цикле и растянута до ячейки J10, J11 – итог первого выражения SumSum.

Формулы для вычисления выражения SumSum:

C2) =(B2^COS($A$2) + $A$2^COS(B2))^2;

J9) =(B9^COS($A$9) + $A$9^COS(B9))^2;

C10) =СУММ(C2:C9);

J11) =СУММ(C10:J10).

На рис. 3.1 представлен вид решения в MS Excel.

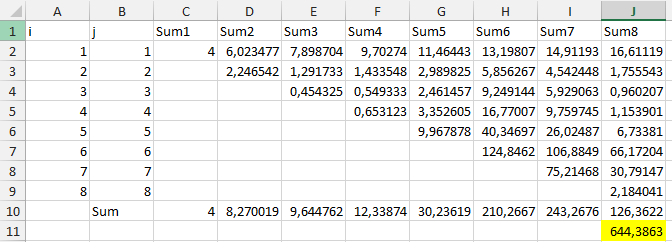


Рис. 3.1. Решение задачи в MS Excel

Далее в таблице 3.1 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Таблица 3.1

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 выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

## 3.3.2 Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в ячейку А2:A6 записаны все значения i, в ячейки B2:В6 – все значения j. В ячейки C2 записана формулы для произведения с адаптацией цикла for в MS Exel. В ячейки C2:G6 записаны формулы произведения функций, С7:G7 – произведения значений в цикле и растянуты до ячейки G7, G8 – итог первого выражения ProdProd.

Формулы для вычисления выражения ProdProd:

C2) =SIN(B2^COS($A$2));

G6) =SIN(B6^COS($A$6));

C7) =ПРОИЗВЕД(C2:C6);

G8) =ПРОИЗВЕД(C7:G7).

На рис. 3.2 представлен вид решения в MS Excel.

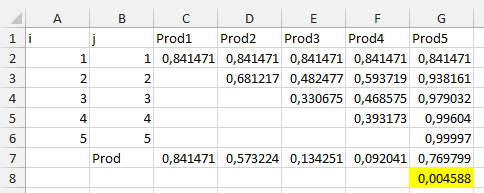


Рис. 3.2. Решение задачи в MS Excel

Далее в таблице 3.2 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Таблица 3.2

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 выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

## 3.3.3 Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в столбец F записаны все значения i, относительно которых вычислялась сумма на разных итерациях цикла for, в столбец G – все значения j. Относительно значений j записываются значения k(как в рис. 3.3) и формируется таблица значений функции и их сумма сумм(на рис. 3.3 выделены желтым). Произведение суммы сумм записаны в ячейке подписанный Prod. В крайней верхней левой ячейке таблиц записана формула для вычисления функции(1) и растянута по таблице, суммы сумм(2) и произведения суммы сумм(3).

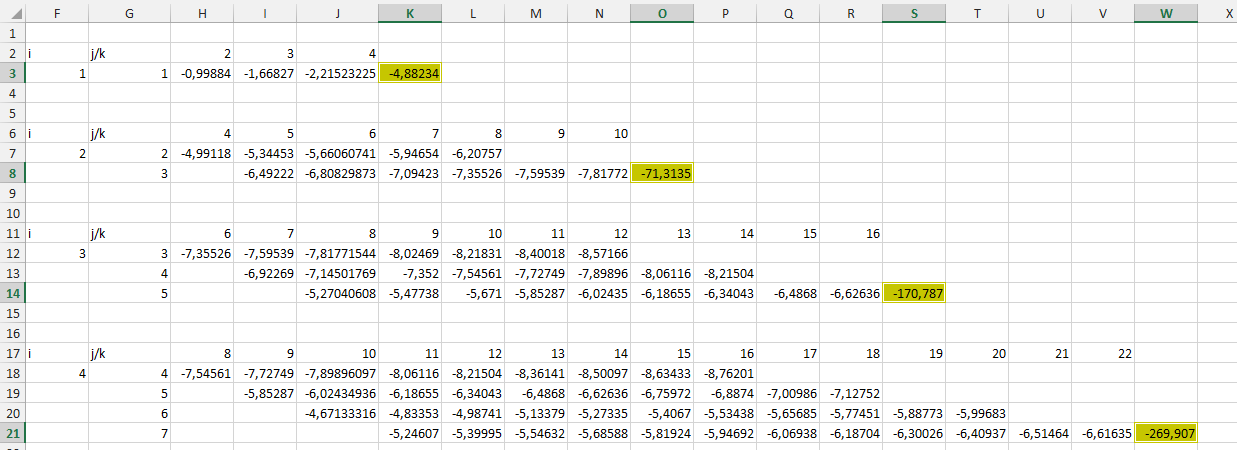
Формулы для вычисления функции, суммы сумм и их произведения:

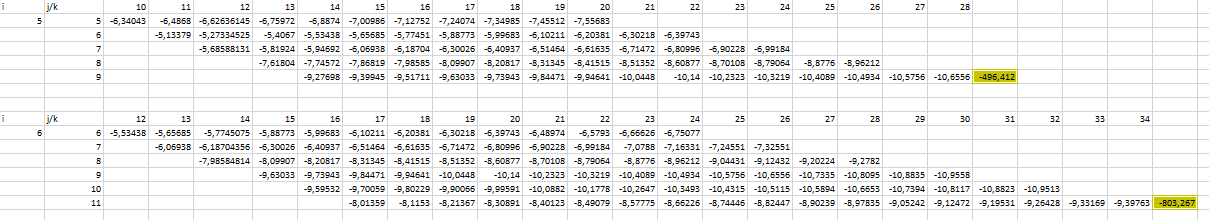
1) =2\*COS($G12)-3\*LN($F$12+0,5\*H$11);

2) =СУММ(H12:R14);

3) =ПРОИЗВЕД(AM59;AI48;AE38;AA29;W21;S14;O8;K3).

На рис. 3.3 представлен вид решения в MS Excel.





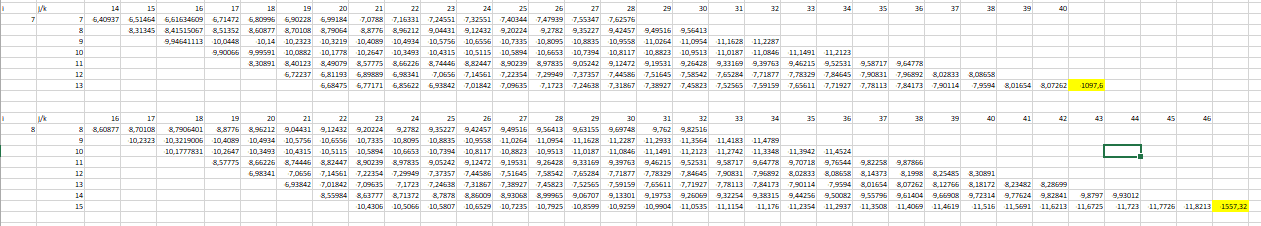


Рис. 3.3. Решение задачи в MS Excel

Далее в таблице 3.3 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Таблица 3.3

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 совпадает. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# 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.Scanner;  
  
public class zad4 {  
 public static int Perevod(int x,int y) {  
 int c = Math.*abs*(x);  
 String num = "";  
 while (c > 0) {  
 num = c % y + num;  
 c /= y;  
 }  
 if (x<0){  
 c = Integer.*parseInt*(num)\*-1;  
 } else {  
 c = Integer.*parseInt*(num);  
 }  
 return c;  
 }  
 public static boolean Proverka(int x, int y, int z){  
 int i = -1;  
 int c = Math.*abs*(x);  
 int num = 0;  
 while(c>0){  
 i++;  
 num += (c%10)\*Math.*pow*(y,i);  
 c /= 10;  
 }  
 if (x<0){  
 num = num\*-1;  
 }  
 return num==z;  
 }  
  
 public void zad4() {  
 Scanner scn = new Scanner(System.*in*);  
 int fl = 1;  
 while (fl == 1) {  
 System.*out*.println("Введите целое десятичное число для перевода:");  
 int x = scn.nextInt();  
 int y = 0;  
 while (y < 2 || y > 10) {  
 System.*out*.println("Введите систему исчисления для перевода в диапазоне [2,10]:");  
 y = scn.nextInt();  
 }  
 System.*out*.println("Результат: "+*Perevod*(x,y));  
 if (*Proverka*(*Perevod*(x,y),y,x)) {  
 System.*out*.println("В десятичной = "+ x+" - верно");  
 } else {  
 System.*out*.println("В десятичной != "+ x+" - неверно");  
 }  
 System.*out*.println("Хотите продолжить перевод? 1 - Да, 2 - Нет");  
 fl = scn.nextInt();  
 }  
 }  
}

## 4.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в столбец А записаны остатки от деления заданного числа х на заданное число y(система счисления в которую преобразуется x), в столбец В с ячейки B2 – целочисленное деления х на у, С1 – у. В ячейки А2, B2 записаны формулы для вычисления остатка и целого значения, растянутые вниз.

Формулы для вычисления функций y(x):

А2) =ОСТАТ(B1;$C$1);

В2) =ЦЕЛОЕ(B1/$C$1).

На рис. 4 представлен вид решения в MS Excel.

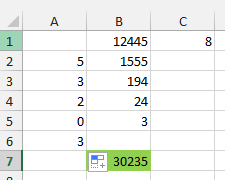


Рис. 4. Решение задачи в MS Excel

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 |  |  |
| 2 |  |  |

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 выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# 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 static java.lang.Math.*pow*;  
  
public class Oblast {  
 private double x, y;  
  
 public void setX(double x) {  
 this.x = x;  
 }  
  
 public void setY(double y) {  
 this.y = y;  
 }  
  
 public boolean Oblast(){  
 if (x \* x + y \* y <= 36 && y >= 0) {  
 if (*Trap*(x, y) == false) {  
 if (*pow*(y - 2, 2) + *pow*(x - 3, 2) >= 4) {  
 return true;  
 } else if (*Okr*(x, y)) {  
 return true;  
 } else {  
 return false;  
 }  
 } else if (*Okr*(x, y)) {  
 return true;  
 } else {  
 return false;  
 }  
 } else {  
 return false;  
 }  
 }  
  
 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 Trap(double x, double y) {  
 if (y < x \* *k*(-5, 0, -3, 5) + *b*(-5, 0, -3, 5) &&  
 y < 5 && y < x \* *k*(1, 0, 0, 5) + *b*(1, 0, 0, 5)) {  
 return true;  
 }  
 return false;  
 }  
  
 public static boolean Okr(double x, double y) {  
 if (*pow*(y - 1, 2) + *pow*(x + 2, 2) <= 1 || *pow*(y - 3, 2) + *pow*(x + 2, 2) <= 1 ||  
 *pow*(y - 3, 2) + *pow*(x - 3, 2) <= 1) {  
 return true;  
 }  
 return false;  
 }  
  
}

import java.util.Scanner;  
  
public class zad6 {  
 public void zad6() {  
 Scanner scn = new Scanner(System.*in*);  
 Oblast oblast = new Oblast();  
 int i = 1;  
 while (i == 1) {  
 System.*out*.println("Задайте Х:");  
 double x = scn.nextDouble();  
 System.*out*.println("Задайте Y:");  
 double y = scn.nextDouble();  
 oblast.setX(x);  
 oblast.setY(y);  
 if (oblast.Oblast()) {  
 System.*out*.println("Попадание в закрашенную область");  
 } else {  
 System.*out*.println("Точка не попала");  
 }  
 System.*out*.println("Хотите проверить другую точку?");  
 i = -1;  
 while (i != 1 && i != 0) {  
 System.*out*.println("Если да - 1, иначе - 0");  
 i = scn.nextInt();  
 }  
 }  
 }  
}

public static boolean Okr(double x, double y) {  
 if (*pow*(y - 1, 2) + *pow*(x + 2, 2) <= 1 || *pow*(y - 3, 2) + *pow*(x + 2, 2) <= 1 ||  
 *pow*(y - 3, 2) + *pow*(x - 3, 2) <= 1) {  
 return true;  
 }  
 return false;  
 }  
}

## 6.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в ячейку А2 записана переменная x, в ячейку B2 – y, в ячейку С2 – проверка попадания точки в закрашенную область, в ячейку А4 – проверка попадания в трапецию Trap, в ячейку B4 – проверка попадания в окружности с радиусом 1 Okr. В ячейки С2, А4 и В2 записаны формула для проверки попадания в область, проверка попадания в трапецию с учетом условия в С2 где у >= 0, проверка попадания в любую из вписанных окружностей с радиусом 1.

Формулы для вычисления функций Trap, Okr и проверки попадания:

С2) =ЕСЛИ(И(A2^2+B2^2 <= 36; B2 >= 0); ЕСЛИ(A4=0; ЕСЛИ((B2-2)^2 + (A2-3)^2 >= 4;"Попадание в закрашенную область";ЕСЛИ(B4 = 1;"Попадание в закрашенную область";"Точка не попала")); ЕСЛИ(B4 = 1;"Попадание в закрашенную область";"Точка не попала"));"Точка не попала");

А4) =ЕСЛИ(И(B2 < A2 \*(2,5) + 12,5;B2 < A2 \*(-5) +5 );1;0);

B4) =ЕСЛИ(ИЛИ(1 >= (B2-1)^2 + (A2+2)^2;1 >= (B2-3)^2 + (A2+2)^2;1 >= (B2-3)^2 + (A2-3)^2);1;0).

На рис. 6 представлен вид решения в MS Excel.

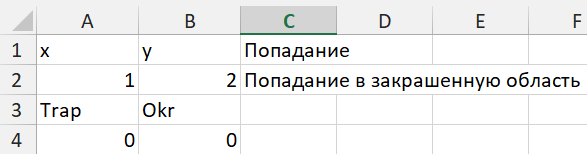


Рис. 6. Решение задачи в MS Excel

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 | Решение MS Excel |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |

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 выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# Задание 7

## 7.1. Постановка задачи

Напечатать полную таблицу целочисленного деления в виде:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 / 1 = 1 | 2 / 1 = 2 | ... | 9 / 1 = 9 |
| 1 / 2 = 0 | 2 / 2 = 1 | ... | 9 / 2 = 4 |
| ... | ... | ... | ... |
| 1 / 9 = 0 | 2 / 9 = 0 | ... | 9 / 9 = 1 |

## 7.2. Решение задачи, код программы

public class zad7 {  
 public static void Ceil(int k){  
 for (int i = 1; i <= k; i++){  
 for (int j = 1; j <= k; j++){  
 System.*out*.print(j+" / "+i+" = "+(j/i)+"\t");  
 }  
 System.*out*.println();  
 }  
 }  
  
 public void zad7() {  
 *Ceil*(9);  
 }  
}

## 7.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в столбец А записаны знаменатели выражения и строку В записаны числители, в соответствии с данными сформирована таблица итоговых значений. В ячейку В2 записана формула для левого верхнего значения таблицы и растянута по всей таблице.

Формула для вычисления выражения:

B2) =ЦЕЛОЕ(B$1/$A2).

На рис. 7. представлен вид решения в MS Excel.

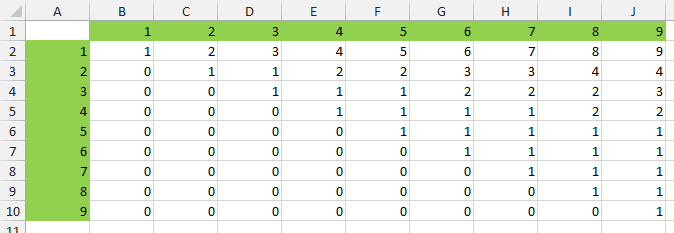


Рис. 7.1. Решение задачи в MS Excel

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 |  |  |

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 выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# Задание 8

## 8.1. Постановка задачи

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.\*;  
  
public class FunctionMy {  
 public static String K(double t, double l, double y) {  
 if (y <= 0 || *log*(y) + *exp*(2\*l) == 0) {  
 return "Ошибка";  
 } else {  
 double Chislitel = 2 \* *pow*(t, 2) + 3 \* l + 7.2;  
 double Znamenatel = *log*(y) + *exp*(2\*l);  
 return String.*valueOf*(Chislitel / Znamenatel);  
 }  
 }  
 public static String L(double a, double x, double y, double z) {  
 if (y \* z < -1 || y \* z > 1 || *abs*(x - *sin*(y \* z)) > 1 || *abs*(x \* *pow*(y, 3) - z) == 0 || *abs*(*log*(*abs*(x \* *pow*(y, 3) - z))) > 1) {  
 return "Ошибка";  
 } else {  
 double first = a - *exp*(2 \* *cos*(*abs*(x - *sin*(y \* z))));  
 double sec = *E* \* *cos*(*abs*(*log*(*abs*(x \* *pow*(y, 3) - z))));  
 return String.*valueOf*(first + sec);  
 }  
 }  
 public static void FunctionMy(double x, double y) {  
 String K = *K*(x,1,1);  
 String L = *L*(y,1,0,1);  
 if (K.equals("Ошибка")) {  
 System.*out*.println("Ошибка в данных 1-го уравнения");  
 } else {  
 System.*out*.println("K = " + K);  
 }  
 if (L.equals("Ошибка")) {  
 System.*out*.println("Ошибка в данных 2-го уравнения");  
 } else {  
 System.*out*.println("L = " + L);  
 }  
 }  
}

import java.util.Scanner;  
  
public class zad8 extends FunctionMy {  
 public void zad8() {  
 Scanner scn = new Scanner(System.*in*);  
 for (int i = 1; i <= 10; i++){  
 *FunctionMy*(scn.nextDouble(), scn.nextDouble());  
 }  
 }  
}

## 8.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в ячейку А2 записана переменная t, в ячейку B2 – l и в ячейку C2 – y. В ячейку D2 записана формула для вычисления значения функции K.

Формула для вычисления функции K:

D2) =ЕСЛИ(C2<=0; "Ошибка данных"; ЕСЛИ(LN(C2) + EXP(2\*B2)=0; "Ошибка данных"; (2\*A2^2+3\*B2+7,2) / (LN(C2) + EXP(2\*B2)))).

На рис. 8 представлен вид решения в MS Excel.

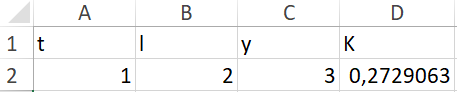


Рис. 8. Решение задачи в MS Excel

Далее в таблице 8 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Таблица 8

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Решение MS Excel |
| 1 |  |  |
| 2 |  |  |

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 выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# Задание 9

## 9.1. Постановка задачи

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 итраций, 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;  
import static java.lang.Math.\*;  
  
public class zad9 {  
 public static double Newton(double y, double p, double x) {  
 return ((p-1)\*y + x/*pow*(y,p-1))/p;  
 }  
 public void zad9() {  
 Scanner scn = new Scanner(System.*in*);  
 System.*out*.println("Введите подкоренное число больше нуля: ");  
 double x = scn.nextDouble();  
 if (x<0){  
 System.*out*.println("Под корнем не может быть отрицательного числа");  
 System.*out*.println("Взят модуль от данного числа");  
 x = *abs*(x);  
 }  
 System.*out*.println("Введите степень корня: ");  
 double p = scn.nextDouble();  
 System.*out*.println("Точность:\t Корень:\t Число итераций:\t");  
 for (int i = 2; i <=6; i++) {  
 double y = *exp*(*log*(x \* (p + 1)) / p) - 0.1;  
 double e = *pow*(10, -i);  
 int c = 0;  
 while(*abs*(y-*Newton*(y,p,x)) > e) {  
 c++;  
 y = *Newton*(y, p, x);  
 }  
 System.*out*.printf(" %s\t %s\t %s\t", e, y, c);  
 System.*out*.println();  
 }  
 }  
}

## 9.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в столбец А записаны все значения точности вычислений е, в столбец B – все итерации вычисления корня с помощью уравнения Ньютона в зависимости от е, С1 – показатель корня, С2 – подкоренное число, С3 – начальное значение уравнения Ньютона у1. В ячейках D3, B1, B2(где формула одинакова для каждой итерации исключая е) записаны формулы для вычисления у1, у при второй итерации, у при последующих итерациях.

Формулы для вычисления у1, у при второй итерации, у при последующих итерациях:

D3) =EXP(LN(D2 \* (D1 + 1)) / D1) - 0,1;

В1) =((D1-1)\*D3 + D2/D3^(D1-1))/D1;

B2) =ЕСЛИ(ABS((($D$1-1)\*B1 + $D$2/B1^($D$1-1))/$D$1 - B1) > $A$1;(($D$1-1)\*B1 + $D$2/B1^($D$1-1))/$D$1;B1).

На рис. 9 представлен вид решения в MS Excel.

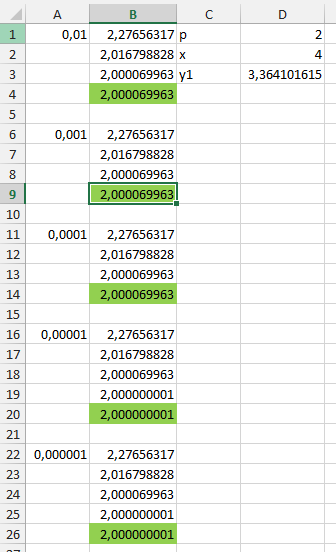


Рис. 9. Решение задачи в MS Excel

Далее в таблице 9 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Таблица 9

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Решение MS Excel |
| 1 |  |  |
| 2 |  |  |

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 выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

# 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  – правый прямоугольник, т.к. высота прямоугольника берется как значение функции в крайней правой координате 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 правый прямоугольник

## 10.2. Решение задачи, код программы

import static java.lang.Math.\*;  
  
public class zad10 {  
 public static double f(double x) {  
 return *exp*(x\*x + 5\*x) + 9;  
 }  
 public void zad10() {  
 double resh = 1.371520897744081\**pow*(10,5);  
 double a = 1, b = 2;  
 System.*out*.println("n: \t\t h:\t\t S:\t\t\t\t Абс. решение:\tПогрешность: ");  
 for (int n = 10; n <= 10000; n\*=10){  
 double h = (b - a)/n;  
 double sum = 0;  
 for (int i = 1; i <= n; i++){  
 sum+=*f*(a+h\*i)\*h;  
 }  
 System.*out*.printf("10^%s\t %s\t %.4f\t %.4f\t",(int) *log10*(n),h,sum,resh);  
 System.*out*.printf("%.4f",*abs*((resh - sum)/resh)\*100);  
 System.*out*.print("%");  
 System.*out*.println();  
 }  
 }  
}

## 10.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в ячейке А2 записано значение переменной а, В2 – значение переменной b, С2 – значение переменной n, D2 – значение переменной h, в столбец А с 3 ячейки записаны все значения i цикла for до n, а в столбец В с 3 ячейки записаны все члены суммы, в ячейку D5 - площадь под графиком. В ячейки D2 записана формула для вычисления h, В3 - формула для вычисления членов суммы растянутая вниз до конечного i, D5 – площадь под графиком.

Формулы для вычисления h, членов суммы, площадь под графиком:

D2) =(B2-A2)/C2;

В3) =EXP((($A$2+$D$2\*A3)^2 + 5\*($A$2+$D$2\*A3))+9)\*$D$2;

D5) =СУММ(B3:B12).

На рис. 10 представлен вид решения в MS Excel.

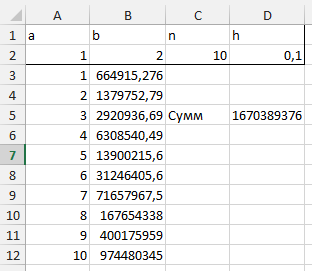


Рис. 10. Решение задачи в MS Excel

Далее в таблице 10 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Таблица 10

Тестирование работы программы и проверка результатов решения

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

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 выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.