Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчёт по лабораторной работе № 5

тема «Циклы в Java»

по дисциплине «Информатика»

Выполнил: студент группы ИСТ-22-1б Золотарев И.О.

Пермь, 2022

**Содержание**

[Задание 1 3](#_Toc119395672)

[1.1. Постановка задачи 3](#_Toc119395673)

[1.2. Решение задачи, код программы 3](#_Toc119395674)

[1.3. Тестирование работы программы с проверкой 3](#_Toc119395675)

[Задание 2 4](#_Toc119395676)

[2.1. Постановка задачи 4](#_Toc119395677)

[2.2. Решение задачи, код программы 4](#_Toc119395678)

[2.3. Тестирование работы программы с проверкой 4](#_Toc119395679)

[Задание 3 5](#_Toc119395680)

[3.1. Постановка задачи 5](#_Toc119395681)

[3.2. Решение задачи, код программы 5](#_Toc119395682)

[3.3. Тестирование работы программы с проверкой 5](#_Toc119395683)

[Задание 4 6](#_Toc119395684)

[4.1. Постановка задачи 6](#_Toc119395685)

[4.2. Решение задачи, код программы 6](#_Toc119395686)

[4.3. Тестирование работы программы с проверкой 6](#_Toc119395687)

[Задание 5 7](#_Toc119395688)

[5.1. Постановка задачи 7](#_Toc119395689)

[5.2. Решение задачи, код программы 7](#_Toc119395690)

[5.3. Тестирование работы программы с проверкой 7](#_Toc119395691)

[Задание 6 8](#_Toc119395692)

[6.1. Постановка задачи 8](#_Toc119395693)

[6.2. Решение задачи, код программы 8](#_Toc119395694)

[6.3. Тестирование работы программы с проверкой 8](#_Toc119395695)

[Задание 7 9](#_Toc119395696)

[7.1. Постановка задачи 9](#_Toc119395697)

[7.2. Решение задачи, код программы 9](#_Toc119395698)

[7.3. Тестирование работы программы с проверкой 9](#_Toc119395699)

[Задание 8 10](#_Toc119395700)

[8.1. Постановка задачи 10](#_Toc119395701)

[8.2. Решение задачи, код программы 10](#_Toc119395702)

[8.3. Тестирование работы программы с проверкой 10](#_Toc119395703)

[Задание 9 11](#_Toc119395704)

[9.1. Постановка задачи 11](#_Toc119395705)

[9.2. Решение задачи, код программы 11](#_Toc119395706)

[9.3. Тестирование работы программы с проверкой 11](#_Toc119395707)

[Задание 10 12](#_Toc119395708)

[10.1. Постановка задачи 12](#_Toc119395709)

[10.2. Решение задачи, код программы 12](#_Toc119395710)

[10.3. Тестирование работы программы с проверкой 12](#_Toc119395711)

# 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

import java.util.Scanner;  
import static java.lang.Math.\*;  
  
public class laba61 {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Введите m");  
 double m = sc.nextDouble();  
 System.*out*.println("Введите n");  
 double n = sc.nextDouble();  
 System.*out*.println("Введите x"); //переменная//  
 double x = sc.nextDouble();  
 double sum1 = 0; //сумма первых n членов ряда//  
 double sum2 = 0; //сумма членов ряда, которые меньше m//  
 double i = 0; //счётчик количества итераций//  
 while (i < n){  
 sum1 += *pow*(-1,n-1)\*(*pow*(x-1,2\*n-1))/*getFactorial*(2\*n-1);  
 i += 2;  
 }  
 while (n > 1){  
 if (*pow*(-1,n-1)\*(*pow*(x-1,2\*n-1))/*getFactorial*(2\*n-1) < m)  
 sum2 += (*pow*(-1,n-1)\*(*pow*(x-1,2\*n-1))/*getFactorial*(2\*n-1));  
 n -= 2;  
 }  
 System.*out*.println(sum1);  
 System.*out*.println(sum2);  
 }  
  
 public static double getFactorial(double f) {  
 if (f <= 1) {  
 return 1;  
 }  
 else {  
 return f \* *getFactorial*(f - 1);  
 }  
 }  
}

## 1.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в ячейку А1

записана переменная m, в ячейку B1 – n, в ячейку C1 – x и в ячейку F1 – i. В ячейки D2, E2, G2 и H2 записаны формулы.

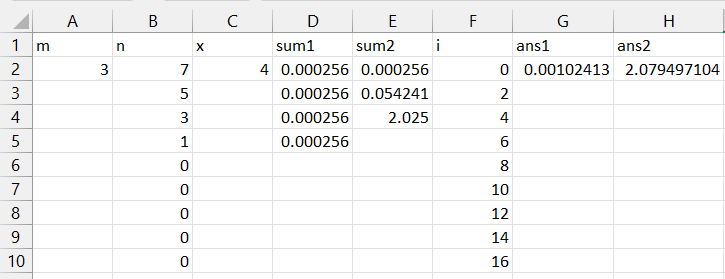
Формулы для вычисления:

D2 = ЕСЛИ(F2<$B$2, (-1^($B$2-1)\*($C$2-1)^(2\*$B$2-1))/ФАКТР(2\*$B$2-1), "")

E2 = ЕСЛИ(И(B2>1, (-1^(B2-1)\*($C$2-1)^(2\*B2-1))/ФАКТР(2\*B2-1)<$A$2), (-1^(B2-1)\*($C$2-1)^(2\*B2-1))/ФАКТР(2\*B2-1), "")

G2 = СУММ(D2:D10)

H2 = СУММ(E2:E10)



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 (например, для числа 132 233 ответ равен 3, для числа 46 336 – 2, для числа 12 345 – 1).

## 2.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class laba62 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите натуральное число");  
 int n = in.nextInt();  
 int p, cnt = 0, maxdigit = 0;  
 while (n != 0){  
 p = n % 10;  
 n /= 10;  
 if (p > maxdigit){  
 maxdigit = p;  
 cnt = 0;  
 }  
 if (p == maxdigit) cnt++;  
 }  
 System.*out*.println(cnt);  
 }  
}

## 2.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в ячейку А1

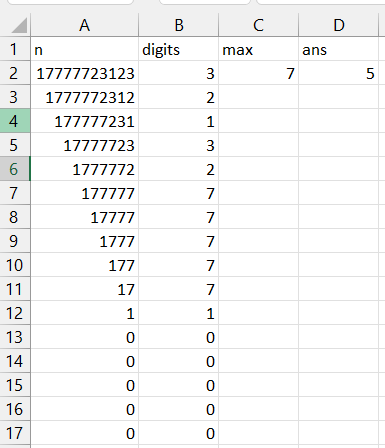
записана переменная n. В ячейки B2, C1, D2 записаны формулы для нахождения остатка от деления на число, для нахождения наибольшей цифры в числе и для нахождения количества этой цифры в числе.

Формулы для вычисления:

B2 = ОСТАТ(A2,10)

C2 = НАИБОЛЬШИЙ(B2:B7,1)

D2 = СЧЁТЕСЛИ(B2:B17,C2)



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

Написать программы, которые вычисляют выражения:

, , 

## 3.2. Решение задачи, код программы

a)

import static java.lang.Math.\*;  
  
public class laba63a {  
 public static void main(String[] args) {  
 double a = 0;  
 for (int i = 1; i <= 8; i++){  
 for (int j = i; j <= 8; j++){  
 a += *pow*(j + 2\*i - 1,2);  
 }  
 }  
 System.*out*.println(a);  
 }  
}

b)

import static java.lang.Math.\*;  
  
public class laba63a {  
 public static void main(String[] args) {  
 double a = 0;  
 for (int i = 1; i <= 8; i++){  
 for (int j = i; j <= 8; j++){  
 a += *pow*(j + 2\*i - 1,2);  
 }  
 }  
 System.*out*.println(a);  
 }  
}

c)

import static java.lang.Math.\*;  
  
public class laba63c {  
 public static void main(String[] args) {  
 double i, j, k, a = 0, b = 0, c = 1;  
 for (i = 1; i <= 8; i++){  
 for(j = i; j<= 2 \* i; j++){  
 for (k = 1; k <= j ; k++){  
 a += (i + *pow*(j - k, i + j));  
 }  
 b += a;  
 }  
 c \*= b;  
 }  
 System.*out*.println(c);  
 }  
}

## 3.3. Тестирование работы программы с проверкой

a) Для проверки задачи в MS Excel создана таблица данных в которой в ячейку

записаны переменные i, j. В ячейки C2:J2 записаны значения для j, в B3:B10 записаны значения для i, в ячейке C3 записана формула, которую мы растягиваем на диапазон C3:J10. В ячейке K2 записана формула для получения ответа.

Формулы:

C3 = ЕСЛИ(C$2>=$B3, (C$2+2\*$B3-1)^2, 0)

K2 = СУММ(C3:J10)

b) Для проверки задачи в MS Excel создана таблица данных в которой в ячейку

записаны переменные i, j. В ячейки C2:G2 записаны значения для j, в B3:B7 записаны значения для i, в ячейке C3 записана формула, которую мы растягиваем на диапазон C3:G7. В ячейке H2 записана формула для получения ответа.

Формулы:

C3 = ЕСЛИ(C$2<=$B3, SIN($B3+C$2), 1)

H2 = ПРОИЗВЕД(C3:G7)

c)

В таблице 3 представлено решение задания на языке Java с проверкой в MS Excel.

Таблица 3

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Решение MS Excel |
| a |  |  |
| b |  |  |
| c |  |  |

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 laba64 {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Введите число");  
 int number = sc.nextInt();  
 boolean ans = *getAns*();  
 while (ans) {  
 int ans1 = Integer.*parseInt*(*getInBase*(number));  
 System.*out*.println(ans1);  
 ans = *getAns*();  
 }  
 }  
 public static String getInBase(int number) {  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Введите основание");  
 int base = sc.nextInt();  
 if (base < 2 || base >= 11 || number < 0) {  
 throw new IllegalArgumentException();  
 }  
 StringBuilder num = new StringBuilder();  
 while (number > 0) {  
 num.insert(0, number % base);  
 number = number / base;  
 }  
 return num.toString();  
 }  
 public static boolean getAns(){  
 System.*out*.println("Изменить систему счисления? Напишите - True или False");  
 Scanner sc = new Scanner(System.*in*);  
 return sc.nextBoolean();  
 }  
}

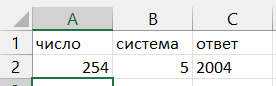
## 4.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в ячейку А1, B2

записаны переменные «число», «система». В ячейку С2 записана формула для нахождения числа в указанной системе счисления.

Формула для вычисления:

С2 = ОСНОВАНИЕ(A2, B2, 1)



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:

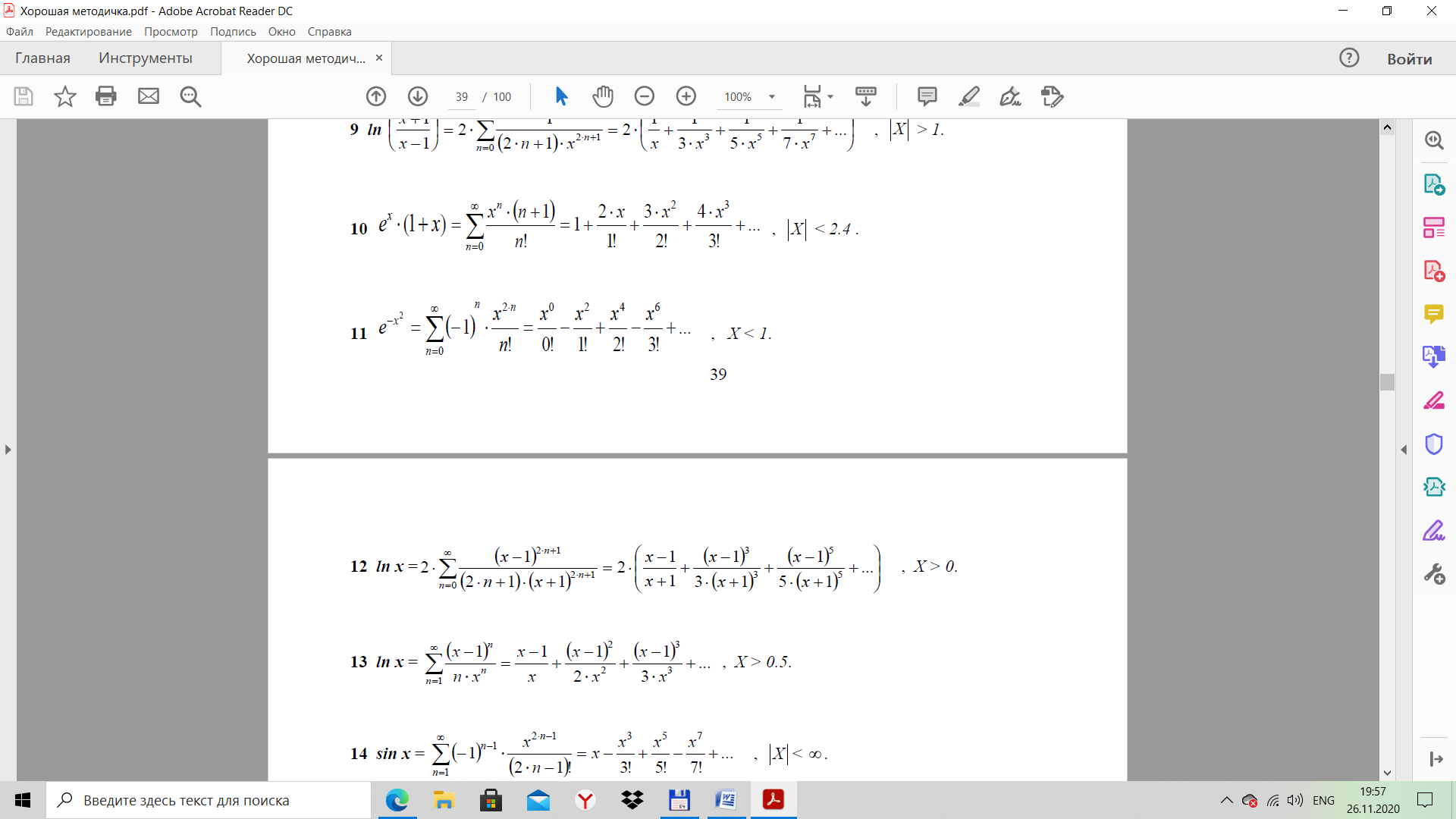
|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Решение MS Excel |
| 1 |  |  |
| 2 |  |  |

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 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 static java.lang.Math.\*;  
import java.util.Scanner;  
  
public class laba65 {  
 public static void main(String[] args) {  
 double ep1 = 0.0001;  
 double ep2 = 0.001;  
 double ep3 = 0.01;  
 Scanner sc = new Scanner(System.*in*);  
 double x = sc.nextDouble();  
 System.*out*.println("ep1 = " + *summa*(ep1,x));  
 System.*out*.println("ep2 = " + *summa*(ep2,x));  
 System.*out*.println("ep3 = " + *summa*(ep3,x));  
 }  
 public static double summa(double ep,double x){  
 double s = 0;  
 int n = 1;  
 while (*abs*(*log*(x) - 2 \* s) > ep){  
 s += *pow*(x - 1, n)/(n \* *pow*(x + 1, n));  
 n += 2;  
 }  
 return 2 \* s;  
 }  
}

## 5.3. Тестирование работы программы с проверкой

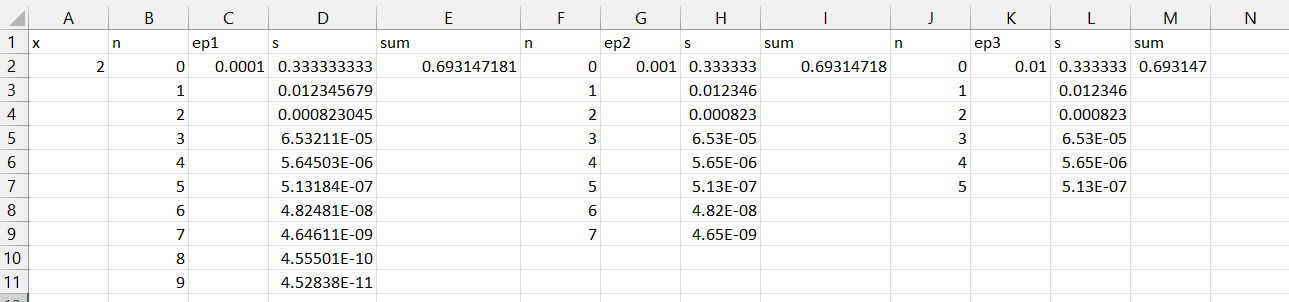
Для проверки задачи в MS Excel создана таблица, где в ячейку А1 записана переменная x, в диапазонах B:B, F:F, J:J – значения переменной n. В ячейки C3, G3, K3 записаны погрешности от 0,0001 до 0, 01. В диапазонах D:D, H:H, L:L записаны формулы, по которым вычисляется данное выражение. В ячейках E2, I2, M2 записаны формулы, по которым вычисляются суммы при трех погрешностях.

D2 = ЕСЛИ(ABS(LN($A$2) - 2 \* (($A$2-1)^(2\*B2+1))/((2\*B2+1)\*($A$2+1)^(2\*B2+1))) > C2,(($A$2-1)^(2\*B2+1))/((2\*B2+1)\*($A$2+1)^(2\*B2+1)),0)

E2 = 2 \* СУММ(D2:D21)

I2 = 2 \* СУММ(H2:H21)

M2 = 2 \* СУММ(L2:L7)



В таблице 1 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 | 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.\*;  
import java.util.Scanner;  
  
public class laba66 {  
  
 public static class Oblast{  
 public static boolean CalculateHIt(double x, double y)  
 {  
 return ((x >= 0 && x \* x + y \* y <= 36) && ((*pow*(x - 2, 2) + *pow*(y - 3, 2) >= 4) || (*pow*(x - 2, 2) + *pow*(y - 4, 2) <= 1))  
 && (((*pow*(x - 2, 2) + *pow*(y + 1, 2) <= 1) || (*pow*(x - 2, 2) + *pow*(y + 3, 2) <= 1))||((!(y < -0.2 \* x + 1) || !(x < 5)) || (!(y > 0.4 \* x - 5)))));  
 }  
 }  
 public static boolean getAns(){  
 System.*out*.println("Проверить точки? 1 - да , иначе - нет");  
 Scanner sc = new Scanner(System.*in*);  
 int ans = sc.nextInt();  
 return ans == 1;  
 }  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 boolean ans = *getAns*();  
 while (ans){  
 System.*out*.println("Введите х и у");  
 double x = sc.nextDouble();  
 double y = sc.nextDouble();  
 System.*out*.println(Oblast.*CalculateHIt*(x,y));  
 ans = *getAns*();  
 }  
 }  
}

## 6.2. Решение задачи, код программы

## 

# Задание 7

## 7.1. Постановка задачи

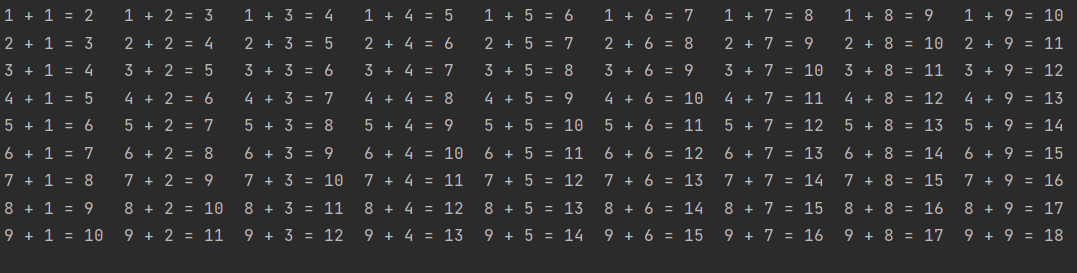
Напечатать полную таблицу сложения в виде:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 + 1 = 2 | 2 + 1 = 3 | ... | 9 + 1 = 9 |
| 1 + 2 = 3 | 2 + 2 = 4 | ... | 9 + 2 = 11 |
| ... | ... | ... | ... |
| 1 + 9 = 10 | 2 + 9 = 11 | ... | 9 + 9 = 18 |

## 7.2. Решение задачи, код программы

import static java.lang.Math.\*;  
  
public class laba67 {  
 public static void main(String[] args){  
 String[][] table = *tableArray*(9, 9);  
 StringBuilder tableF = new StringBuilder();  
  
 for (int line = 0; line < table.length; line++){  
 for (int column = 0; column < table[line].length; column++){  
 tableF.append(line + 1).append(" + ").append(column + 1).append(" = ").append(table[line][column]).append("\t");  
 }  
 System.*out*.println(tableF);  
 tableF = new StringBuilder();  
 }  
 }  
  
 public static String[][] tableArray(int lines, int columns){  
 String[][] tableCount = new String[lines][columns];  
 for (int line = 0; line < tableCount.length; line++){  
 for (int column = 0; column < tableCount[line].length; column++){  
 tableCount[line][column] = String.*valueOf*((line + 1)+(column + 1));  
 }  
 }  
 return tableCount;  
 }  
}

## 7.3. Тестирование кода программы



# Задание 8

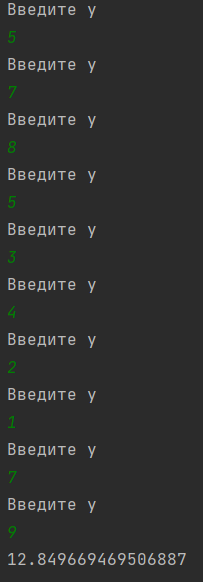
## 8.1. Постановка задачи

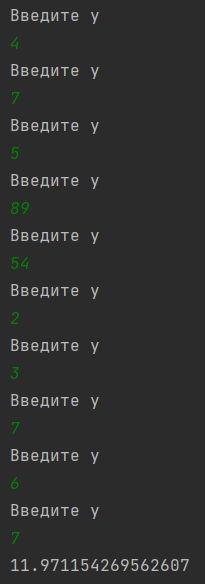
Переделать программу (класс) задания 1 лабораторной работы 2 о нахождении значений 2-х функций в класс без метода main с названием FunctionMy. Метод main заменить на метод FunctionMy c входными данными. Создать программу, взаимодействующую с классом FunctionMy, в которой пользователь в цикле находит 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 laba68 {  
 public static class FunctionMy {  
 public static double FunctionMy1(double y){  
 double m = 1;  
 return ((*pow*(m,2)+2.8\*m+0.355)/(*cos*(2\*y) + 3.6));  
 }  
 public static double FunctionMy2(double y){  
 double x = 1, d = 1, a = 1,c = 1;  
 return ((*pow*(*cos*(y + *sqrt*(x)),y)+2.4\*d\**cbrt*(y-x))/(*pow*(*exp*(1),y)+*pow*(*exp*(1),*pow*(y,x))\**sqrt*(*abs*(a+x\**pow*(c,4)))));  
 }  
 }  
  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 int n = 0;  
 double sum = 0;  
 while (n < 10){  
 System.*out*.println("Введите y");  
 double y = sc.nextDouble();  
 sum += FunctionMy.*FunctionMy1*(y) + FunctionMy.*FunctionMy2*(y);  
 n++;  
 }  
 System.*out*.println(sum);  
 }  
}

## 8.2. Тестирование кода программы





# 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 static java.lang.Math.\*;  
import java.util.Scanner;  
  
public class laba69 {  
 public static void main(String[] args) {  
 double ep1 = 0.01; double ep2 = 0.001; double ep3 = 0.0001; double ep4 = 0.00001; double ep5 = 0.000001; int n = 0;  
 Scanner sc = new Scanner(System.*in*);  
 double x = sc.nextDouble();  
 double p = sc.nextDouble();  
 double y0 = *exp*(*log*(x\*(p+1))/p)\*0.9;  
 *CalculateSquare*(ep1, x, y0, p, n);  
 *CalculateSquare*(ep2, x, y0, p, n);  
 *CalculateSquare*(ep3, x, y0, p, n);  
 *CalculateSquare*(ep4, x, y0, p, n);  
 *CalculateSquare*(ep5, x, y0, p, n);  
 }  
 public static void CalculateSquare(double ep,double x,double y,double p,double n){  
 double y\_i = 1/p\*((p - 1)\*y+x/*pow*(y, p - 1));  
 n++;  
 if (*abs*(y\_i - y) > ep) *CalculateSquare*(ep, x, y\_i, p, n);  
 else System.*out*.println(ep + ", " + (y\_i) + ", " + n);  
 }  
}

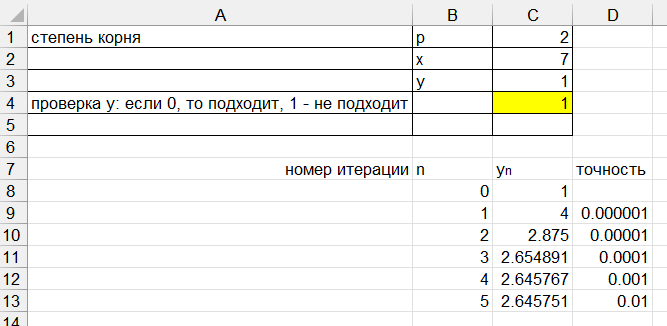
## 9.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных, в которой в ячейку C1 записано значение степени корня p, в C2 – значение переменной x, в C3 –начальное значение y0. В C4 записана формула для проверки y0. В диапазоне B8:B13 записано число итераций n. В диапазоне C8:C13 записаны значения корня p-й степени. В диапазоне D9:D13 записаны погрешности.

Формулы:

C4 =ЕСЛИ(C3<EXP(LN(C2\*(C1+1)));1;0)

C9 =1/$C$1\*(($C$1-1)\*C8+$C$2/C8^($C$1-1))



В таблице 6 представлено тестирование работы программы с проверкой решения

задачи на языке Java с решением задачи в MS Excel.

Таблица 6

|  |  |  |
| --- | --- | --- |
| № 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 laba610 {  
 public static void main(String[] args) {  
 double sum;  
 double h;  
 for(double n = 10;n <= 10000; n \*= 10){  
 h = 1/n;  
 sum = 0;  
 for (int i = 0; i < n; i++){  
 sum += (*cos*(*pow*(n-h\*i,2) + 5 \* (n-h\*i) + 6)+ 5) \* h;  
 }  
 System.*out*.println(n + " " + h + " " + sum + " 5.18" + " "+ *abs*(sum - 5.18));  
 }  
  
 }  
}

## 10.3. Тестирование работы программы с проверкой

В таблице 1 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 на калькуляторе |
| 1 |  |  |

Сравнение решения задачи с использованием двух прикладных пакетов показала, что решения задачи в Java и калькуляторе совпадает. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.