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 «Циклы в Java»

по дисциплине «Информатика»

Выполнил: студент группы ИСТ-22-1б Васин М.А.

Проверил: ассистент каф. ВММБ Нетбай Г.В.

Пермь, 2021

**Содержание**

[Задание 1 3](#_Toc122568645)

[1.1. Постановка задачи 3](#_Toc122568646)

[1.2. Решение задачи, код программы 3](#_Toc122568647)

[1.3. Тестирование работы программы с проверкой 3](#_Toc122568648)

[Задание 2 5](#_Toc122568649)

[2.1. Постановка задачи 5](#_Toc122568650)

[2.2. Решение задачи, код программы 5](#_Toc122568651)

[2.3. Тестирование работы программы 5](#_Toc122568652)

[Задание 3 5](#_Toc122568653)

[3.1. Постановка задачи 5](#_Toc122568654)

[3.2.1 Решение задачи, код программы 5](#_Toc122568655)

[3.2.2 Решение задачи, код программы 6](#_Toc122568656)

[3.2.3 Решение задачи, код программы 6](#_Toc122568657)

[3.3. Тестирование работы программы с проверкой 6](#_Toc122568658)

[Задание 4 8](#_Toc122568659)

[4.1. Постановка задачи 8](#_Toc122568660)

[4.2. Решение задачи, код программы 8](#_Toc122568661)

[4.3. Тестирование работы программы с проверкой 9](#_Toc122568662)

[Задание 6 10](#_Toc122568663)

[6.1. Постановка задачи 10](#_Toc122568664)

[6.2. Решение задачи, код программы 11](#_Toc122568665)

[6.3. Тестирование работы программы 11](#_Toc122568666)

[Задание 7 12](#_Toc122568667)

[7.1. Постановка задачи 12](#_Toc122568668)

[7.2.1 Решение задачи, код программы 12](#_Toc122568669)

[7.3. Тестирование работы программы 12](#_Toc122568670)

[Задание 8 12](#_Toc122568671)

[8.1. Постановка задачи 12](#_Toc122568672)

[8.2. Решение задачи, код программы 12](#_Toc122568673)

[8.3. Тестирование работы программы 13](#_Toc122568674)

[Задание 9 14](#_Toc122568675)

[9.1. Постановка задачи 14](#_Toc122568676)

[9.2. Решение задачи, код программы 14](#_Toc122568677)

[9.3. Тестирование работы программы 15](#_Toc122568678)

[Задание 10 15](#_Toc122568679)

[10.1. Постановка задачи 15](#_Toc122568680)

[10.2. Решение задачи, код программы 16](#_Toc122568681)

[10.3. Тестирование работы программы 16](#_Toc122568682)

# 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.\*;  
import static java.lang.System.*out*;  
public class z1 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 *out*.println("Введите m");  
 double m = in.nextDouble();  
 *out*.println("Введите n");  
 double n = in.nextDouble();  
 *out*.println("Введите x");  
 double x = in.nextDouble();  
 double sum1 = 0;  
 double sum2 = 0;  
 double i = 0;  
 while (i<n) {  
 sum1 += *pow*(x, n)/ *getFactorial*(n);  
 i += 1;  
 }  
 while (n>1){  
 if ((*pow*(x, n)/ *getFactorial*(n))<m)  
 sum2 += *pow*(x, n)/ *getFactorial*(n);  
 n -= 1;  
 }  
 *out*.println("Сумма первых N членов ряда - " + sum1);  
 *out*.println("Сумма членов ряда, которые меньше заданного числа M - " + 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 А2 записана переменная m, в ячейку B2 – n, в ячейку C2 – x, в ячейку F2 – i. В ячейки D2, E2, G2 и H2 записаны формулы для вычисления значения функций sum1, sum2, sum n и sum<m.

Формулы для вычисления функций s, u, answer и u:

D2) = ЕСЛИ(F2<$B$2; (($C$2^$B$2)/(ФАКТР($B$2))); "");

E2) =ЕСЛИ(И(B2>1;(($C$2^B2)/(ФАКТР(B2)))<$A$2);(($C$2^B2)/(ФАКТР(B2)));"");

G2) = СУММ(D2:D10);

H2) = СУММ(E2:E10).

На рис. 1 представлен вид решения в MS Excel.

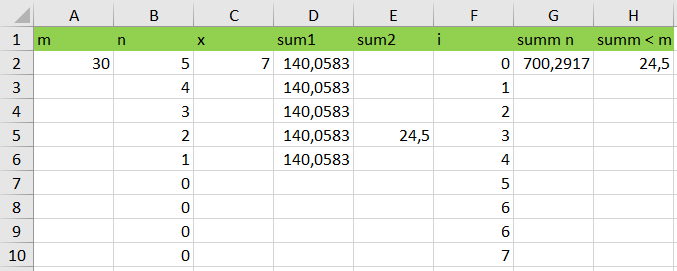


Рис. 1. Решение задачи в MS Excel

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

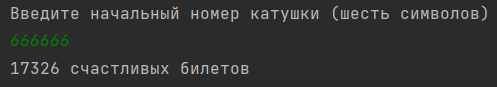
## 2.1. Постановка задачи

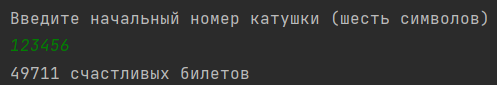
Билет называют «счастливым», если в его номере сумма первых трех цифр равна сумме последних трех (цифры шестизначные). Подсчитать число «счастливых» билетов, которые есть в катушке у кондуктора, если на начало рабочего дня начальный номер катушки N (минимальный номер билета 000000, максимальный номер билета 999999).

## 2.2. Решение задачи, код программы

import java.util.Scanner;  
public class z2 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите начальный номер катушки (шесть символов)");  
 int n = in.nextInt();  
 int sum = 0;  
 for(int i = n; i <= 999999; i++) {  
 int number1 = i/100000;  
 int number2 = (i/10000)%10;  
 int number3 = (i/1000)%10%10;  
 int number4 = (i/100)%10%10%10;  
 int number5 = (i/10)%10%10%10%10;  
 int number6 = i%10%10%10%10;  
 if (number1 + number2 + number3 == number4 + number5 + number6) {  
 sum++;  
 }  
 }  
 System.*out*.println(sum + " счастливых билетов");  
 }  
}

## 2.3. Тестирование работы программы





# Задание 3

## 3.1. Постановка задачи

Написать программы, которые вычисляют выражения:

, , 

## 3.2.1 Решение задачи, код программы

import static java.lang.Math.\*;  
public class z31 {  
 public static void main(String[] args) {  
 double S=0, Sj=0;  
 int j;  
 for (int i=1; i<=8; i++) {  
 j=1;  
 while (j <= i) {  
 Sj += *pow*((j + i), 2);  
 j++;  
 }  
 S += Sj;  
 Sj = 0;  
 }  
 System.*out*.println("Сумма сумм равна " + S);  
 }  
}

## 3.2.2 Решение задачи, код программы

public class z32 {  
 public static void main(String[] args) {  
 double Pr=1;  
 for (int i=1; i<=5; i++) {  
 for (int j = 1; j <= i; j++) {  
 Pr \*= j;  
 }  
 }  
 System.*out*.println("Произведение произведений равно " + Pr);  
 }  
}

## 3.2.3 Решение задачи, код программы

public class z33 {  
 public static void main(String[] args) {  
 double S=0, Prj=1, Prk=1;  
 for (int i=1; i<=8; i++) {  
 for (int j = i; j <= 8; j++) {  
 for (int k = 1; k <= 2\*i; k++) {  
 Prk \*= (2\*j\*i - k);  
 }  
 Prj \*= Prk;  
 Prk = 1;  
 }  
 S += Prj;  
 Prj=1;  
 }  
 System.*out*.println("Сумма произведения произведений - " + S);  
 }  
}

## 3.3. Тестирование работы программы с проверкой

1) Для проверки задачи в MS Excel создана таблица данных в которой в ячейку записаны переменные i, j. В ячейки B1:I1 записаны значения для j, в A2:A9 записаны значения для i, в ячейке B2 записана формула, которую мы растягиваем на диапазон B2:I9. В ячейке J2 записана формула для получения ответа.

Формулы для вычисления:

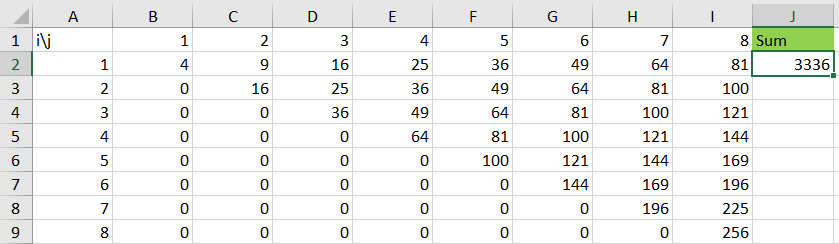
B2) = ЕСЛИ(B$1>=$A2; (B$1+$A2)^2; 0);

B2) =СУММ(B2:I9).

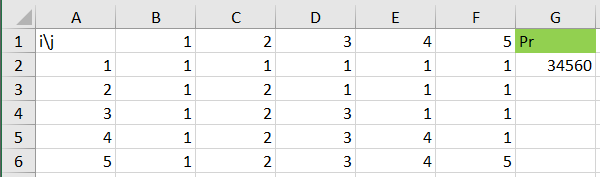
Аналогично для 2 и 3.

На рис. 3 представлен вид решения в MS Excel.

1)



2)



3)

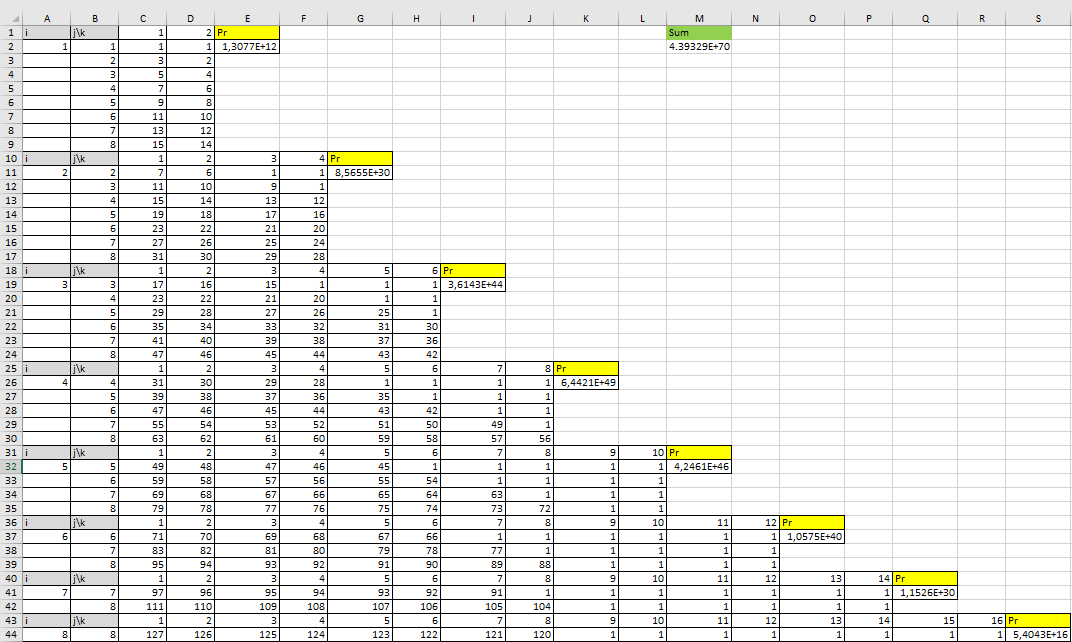


Рис. 3. Решение задачи в MS Excel

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 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 z4 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите целое десятичное число");  
 int x = in.nextInt();  
 boolean otv = *getOtv*();  
 while (otv) {  
 int otv1 = Integer.*parseInt*(*getInOsn*(x));  
 System.*out*.println(otv1);  
 otv = *getOtv*();  
 }  
 }  
 public static String getInOsn(int x) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите основание системы счисления");  
 int osn = in.nextInt();  
 if (osn < 2 || osn >= 11 || x < 0) {  
 throw new IllegalArgumentException();  
 }  
 StringBuilder num = new StringBuilder();  
 while (x > 0) {  
 num.insert(0, x%osn);  
 x = x/osn;  
 }  
 return num.toString();  
 }  
 public static boolean getOtv() {  
 System.*out*.println("Изменить систему счисления? True - да, False - нет");  
 Scanner in = new Scanner(System.*in*);  
 return in.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.

Формулы для вычисления:

C2 =ОСНОВАНИЕ(A2;B2;1).

На рис. 4 представлен вид решения в MS Excel.

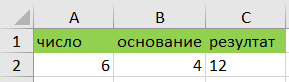


Рис. 4. Решение задачи в MS Excel

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

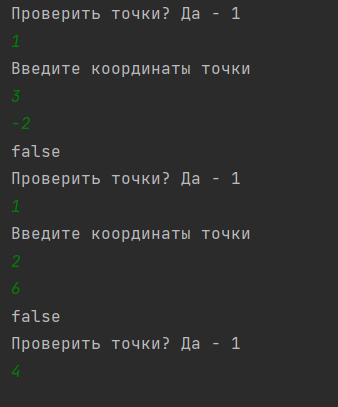
## 6.1. Постановка задачи

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 java.util.Scanner;  
import static java.lang.Math.\*;  
public class z6 {  
 public static class Oblast {  
 public static boolean Koord(double x, double y)  
 {  
 return ((x>=(-5) & (x<=(-2)) & y<=4.0/3\*x+20.0/3 & y>=0) & (x>=(-2) & x<=0 & y<=4 & y>=1.0/2\*x+1) & (!((*pow*((x+2), 2)+*pow*((y-2),2))<=1)) & !(y<=*sqrt*(36-*pow*(x,2)) & y>=-3.0/2\*x+4) & (!(x>=0 & x<=2 & y<=*sqrt*(1-*pow*((x-2),2))+4 & y>=-*sqrt*(1-*pow*((x-2),2))+4)) & (x>=2 & x<=5 & y<=*sqrt*(36-*pow*(x,2)) & y>=1.0/3\*x+1.0/3) & (!(x>=2 & x<=5 & y<=*sqrt*(1-*pow*((x-2),2))+4 & y>=-*sqrt*(1-*pow*((x-2),2))+4)) & (x>=5 & x<=6 & y<=*sqrt*(36-*pow*(x,2)) & y>=0));  
 }  
 }  
 public static boolean getOtv() {  
 System.*out*.println("Проверить точки? Да - 1");  
 Scanner in = new Scanner(System.*in*);  
 int otv = in.nextInt();  
 return otv == 1;  
 }  
  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 boolean otv = *getOtv*();  
 while (otv) {  
 System.*out*.println("Введите координаты точки");  
 double x = in.nextDouble();  
 double y = in.nextDouble();  
 System.*out*.println(Oblast.*Koord*(x, y));  
 otv = *getOtv*();  
 }  
 }  
}

## 6.3. Тестирование работы программы



# Задание 7

## 7.1. Постановка задачи

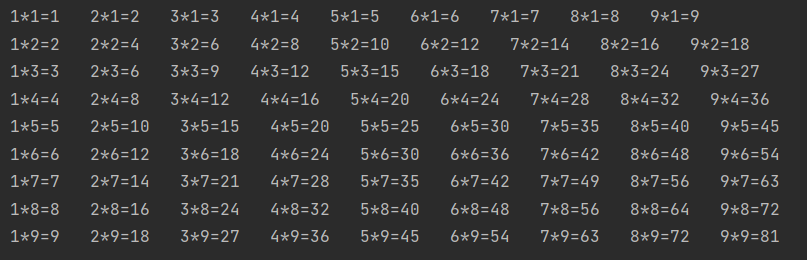
Напечатать полную таблицу умножения в виде:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 х 1 = 1 | 1 х 2 = 2 | ... | 1 х 9 = 9 |
| 2 х 1 = 2 | 2 х 2 = 4 | ... | 2 х 9 = 18 |
| ... | ... | ... | ... |
| 9 х 1 = 9 | 9 х 2 = 18 | ... | 9 х 9 = 81 |

## 7.2.1 Решение задачи, код программы

public class z7 {  
 public static void main(String[] args) {  
 for (int i = 1; i < 10; i++) {  
 for (int j = 1; j < 10; j++) {  
 System.*out*.print(j+"\*"+i+"="+(i\*j)+" ");  
 }  
 System.*out*.println("");  
 }  
 }  
}

## 7.3. Тестирование работы программы



# Задание 8

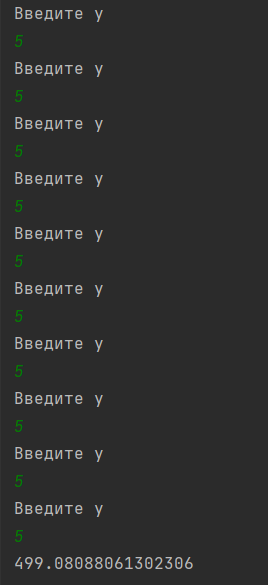
## 8.1. Постановка задачи

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 java.util.Scanner;  
import static java.lang.Math.\*;  
public class z8 {  
 public static class FunctionMy {  
 public static double FunctionMy1(double y){  
 double k = 10;  
 return ((*log*(k-y)+*pow*(y,4))/(*pow*(*exp*(1), y) + 2.355\**pow*(k, 2)));  
 }  
 public static double FunctionMy2(double y){  
 double x = 10, z = 10;  
 return ((*tan*(*pow*(x, 4) - 6)- *pow*(*cos*(z + x\*y), 3))/(*pow*(*cos*(*pow*(x, 3) + *pow*(*sin*(*PI* + x), 3)), 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);  
 }  
}

## 8.3. Тестирование работы программы



# Задание 9

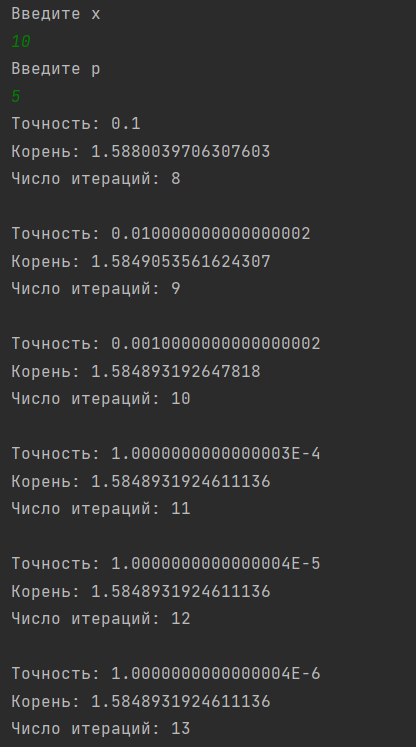
## 9.1. Постановка задачи

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 java.util.\*;  
import static java.lang.Math.\*;  
public class z9 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите x");  
 double x = (in.nextDouble());  
 System.*out*.println("Введите p");  
 double p = (in.nextDouble());  
 double y = *log*(x\*(p+1))/p;  
 int c=0;  
 for (double e=*pow*(10,-1); e>*pow*(10,-6); e\*=0.1) {  
 while (true) {  
 double ly=y;  
 y=1/p\*((p-1)\*y+x/*pow*(y,(p-1))); c++;  
 if (*abs*(y-ly)<e) {  
 System.*out*.println("Точность: "+e+"\nКорень: "+y +"\nЧисло итераций: "+c+"\n");  
 break;  
 }  
 }  
 }  
 }  
}

## 9.3. Тестирование работы программы



# 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 z10 {  
 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*(*cos*(*pow*(n-h\*i,3)+5)) + 5\*(n-h\*i);  
 }  
 System.*out*.println(n + " " + h + " " + sum + " 5.18" + " "+ *abs*(sum - 5.18));  
 }  
  
 }  
}

## 10.3. Тестирование работы программы

