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»

по дисциплине «Информатика»

Вариант 2

Выполнил: студент группу ИСТ-22-1б Афонина А.М

Проверил: доцент каф. ВММБ Каменских А.А

Пермь, 2022

Задача 1

1.1. Постановка задачи

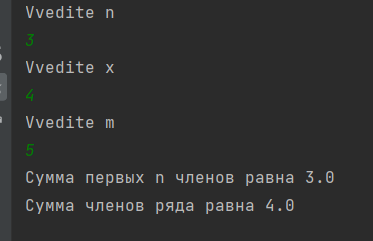
Найти сумму первых N членов ряда и найти сумму членов ряда, которые меньше заданного с клавиатуры числа M:

.

1.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class main1\_6 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Vvedite n");  
 double n = in.nextDouble();  
 System.*out*.println("Vvedite x");  
 double x = in.nextDouble();  
 System.*out*.println("Vvedite m");  
 double m = in.nextDouble();  
 double Sn = (n+1)\*Math.*pow*(x,0);  
 double y = 0;  
 double k = 0;  
 while (Sn < m){  
 y = y+Sn;  
 k = k+n;  
 n++;  
 Sn = (n+1)\*Math.*pow*(x,0);  
 }  
 System.*out*.println("Сумма первых n членов равна " + k);  
 System.*out*.println("Сумма членов ряда равна " + y);  
 }  
}

1.3. Тестирование работы программы



Задача 2

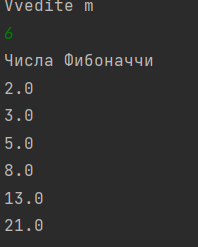
2.1 Постановка задачи

В 1202г итальянский математик Леонард Пизанский (Фибоначчи) предложил такую задачу: пара кроликов каждый месяц дает приплод – двух кроликов (самца и самку), от которых через два месяца уже получается новый приплод. Сколько кроликов будет через год, если в начале года имелась одна пара? Согласно условию задачи числа, соответствующие количеству кроликов, которые появляются через каждый месяц, составляют последовательность 1, 1,2, 3, 5, 8, 13, 21, 37, ... Составьте программу, позволяющую найти все числа Фибоначчи, меньшие заданного числа N.

2.2 Решение задачи, код программы

import java.util.Scanner;  
  
public class main6\_2 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Vvedite m");  
 double m = in.nextDouble();;  
 double a = 1;  
 double b = 1;  
 double c;  
 double i = 0;  
 System.*out*.println("Числа Фибоначчи");  
 while (i<m){  
 c = a+b;  
 b = a;  
 a = c;  
 i++;  
 System.*out*.println(c);  
 }  
 }  
}

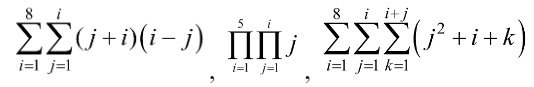
2.3 Тестирование работы программы



Задача 3

3.1 Постановка задачи

Написать программы, которые вычисляют выражения:



3.2.1 Решение задачи, код программы

public class main6\_3 {  
 public static void main(String[] args) {  
 double i;  
 double j;  
 double Sj = 0;  
 double Si = 0;  
 for ( i = 1;i<=8;i++){  
 j = 1;  
 while (j<=i){  
 Sj = Sj + (i+j)\*(i-j);  
 j++;  
  
 }  
 Si = Si+Sj;  
 Sj = 0;  
 }  
 System.*out*.println("Сумма сумм равна = "+ Si);  
}  
}

3.3.1 Тестирование работы программы

# 

3.2.1 Решение задачи, код программы

import java.util.Scanner;  
  
public class main6\_3\_2 {  
 public static void main(String args[]) {  
 Scanner in = new Scanner(System.*in*);  
 long 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);  
 }  
 }  
 }  
}

3.3.1 Тестирование работы программы



3.2.1 Тестирование работы программы

public class main\_6\_3\_3 {  
 public static void main(String[] args) {  
 long Sss = 0;  
 int j;  
 int i;  
 int k ;  
 for ( i = 1; i<=8;i++){  
 for (j=1 ; j<=i; j++){  
 for (k=1;k<=(i+j);k++){  
 Sss += (j\*j+i+k);  
  
 }  
 }  
 }  
 System.*out*.println("Сумма сумм сумм " + Sss);  
  
 }  
}

3.3.1 Тестирование работы программы



Задача 4

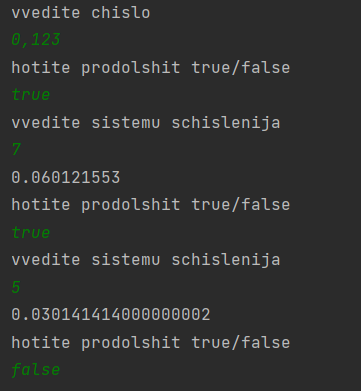
4.1 Постановка задачи

Пользователь вводит десятичную дробь (например, 0,2345 – у дроби нет целой части, если пользователь введет число, где есть целая часть, то должно выплыть сообщение об ошибке). Написать программу перевода десятичной дроби 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, определить погрешность, если она есть. 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 main6\_4 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("vvedite chislo");  
 double number = in.nextDouble();  
 if (number >= 1 || number <0) {  
 System.*out*.println("Oshibka");  
 System.*exit*(0);  
 }  
 boolean otv = *datotvet*();  
 while (otv) {  
 double otvet = Double.*parseDouble*(*basa*(number));  
 System.*out*.println(otvet);  
 otv = *datotvet*();  
 }  
 }  
 public static String basa(double number) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("vvedite sistemu schislenija");  
 int base = in.nextInt();  
 if (base < 2 || base >= 11 || number < 0) {  
 System.*out*.println("Oshibka");  
 System.*exit*(0);  
 }  
 int k = -1;  
 double x=1,ans=0,bv=0;  
 while (number % 2 != 0 && Math.*abs*(k) < 10) {  
 number \*= base;  
 bv = (number - (number % 1));  
 ans += bv \* Math.*pow*(10, k);  
 number -= bv;  
 k--;  
 }  
  
 return String.*valueOf*((ans));  
 }  
 public static boolean datotvet() {  
 System.*out*.println("hotite prodolshit true/false");  
 Scanner sc = new Scanner(System.*in*);  
 return sc.nextBoolean();  
 }  
}

4.3 Тестирование работы программы



Задача 6

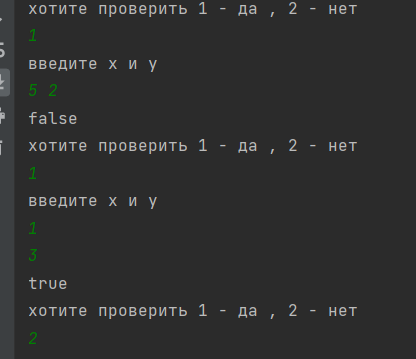
6.1 Постановка задачи

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 main6\_6 {  
 public static class Oblast {  
 public static boolean Vichislenie(double x, double y) {  
 return (x <= Math.*sqrt*(36 - y \* y) & x >= 0) &&(((Math.*pow*(x - 4, 2) + Math.*pow*(y, 2) > 1 && Math.*pow*(x - 4, 2) + Math.*pow*(y, 2) < 4) || (y < 4 && y > 1 && y > x + 1 && x < 3) || (y > x + 1 && y > (-4 / 3) \* x + 1 && y < 4 && y > -3 && x > 0 && x < 3) || (y < 1 && y > -4 && x > 0 && x < 3 && y > (-4 / 3) \* x + 1 && y > (1 / 3) \* x - 4)));  
 }  
 }  
  
 public static boolean getAns() {  
 System.*out*.println("хотите проверить 1 - да , 2 - 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 x и y");  
 double x = sc.nextDouble();  
 double y = sc.nextDouble();  
 System.*out*.println(Oblast.*Vichislenie*(x, y));  
 ans = *getAns*();  
 }  
 }  
}

6.3 Тестирование работы программы



Задача 7

7.1 Постановка задачи

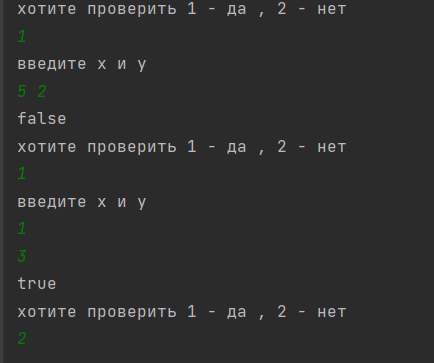
Напечатать полную таблицу сложения в виде:

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

7.2 Решение задачи, код программы

public class main6\_7 {  
 public static void main(String[] args) {  
 int[][] m;  
 m = new int[9][9];  
 int a;  
 int b;  
 int c ;  
 for (a = 1; a<=9;a++){  
 for (b =1; b<=9;b++) {  
 c = a + b;  
 System.*out*.print(a + "+" + b + "=" + c);  
 System.*out*.print(" ");  
 }  
 System.*out*.println(" ");  
 }  
 }  
}

7.3 Тестирование работы программы



Задача 8

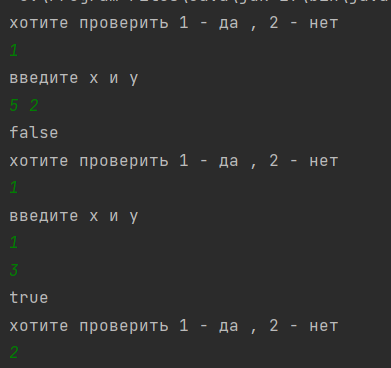
8.1 Постановка задачи

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

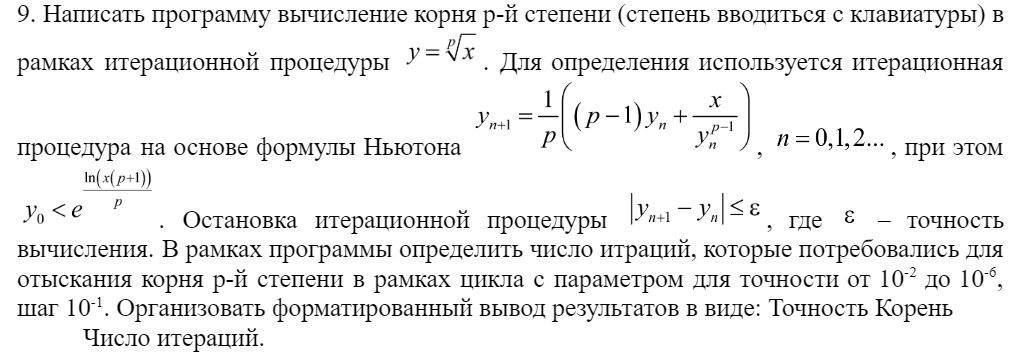
import java.util.Scanner;  
  
public class main6\_8 {  
 public static class FunctionMy {  
 public static double funct1(double y) {  
 final double p = 1;  
 return (Math.*pow*(Math.*sin*(p + 0.4), 2) / (y \* y + 7.325 \* p));  
 }  
  
 public static double funct2(double y) {  
 final double a = 0, x = 0, c = 1;  
 return ((Math.*pow*(a, 5) + Math.*acos*(a + Math.*pow*(x, 3)) - Math.*pow*(Math.*sin*(y - c), 4))) / (Math.*pow*(Math.*sin*(x + y), 3) + Math.*abs*(x - y));  
 }  
 }  
  
 public static void main(String[] args) {  
 Scanner in = new Scanner (System.*in*);  
 double sum = 0;  
 double m = 0;  
 while (m<10){  
 System.*out*.println("Vvedite y");  
 double y = in.nextDouble();  
 sum += FunctionMy.*funct1*(y) + FunctionMy.*funct2*(y);  
 m++;  
 }  
 System.*out*.println("Итоговая сумма 2 функций равна " + sum);  
 }  
}

8.3 Тестирование работы программы



Задача 9

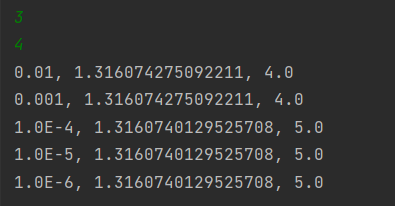
9.1 Постановка задачи



9.2 Решение задачи, код программы

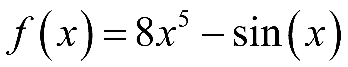
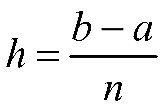
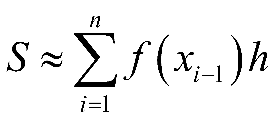
mport static java.lang.Math.\*;  
import java.util.Scanner;  
  
public class main6\_9 {  
 public static void main(String[] args) {  
 double a1 = 0.01; double a2 = 0.001; double a3 = 0.0001; double a4 = 0.00001; double a5 = 0.000001; int n = 0;  
 Scanner in = new Scanner(System.*in*);  
 double x = in.nextDouble();  
 double p = in.nextDouble();  
 double y0 = *exp*(*log*(x\*(p+1))/p)\*0.9;  
 *m*(a1, x, y0, p, n);  
 *m*(a2, x, y0, p, n);  
 *m*(a3, x, y0, p, n);  
 *m*(a4, x, y0, p, n);  
 *m*(a5, x, y0, p, n);  
 }  
 public static void m(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) *m*(ep, x, y\_i, p, n);  
 else System.*out*.println(ep + ", " + (y\_i) + ", " + n);  
 }  
}

9.3 Тестирование работы программы



Задача 10

10.1 Постановка задачи

Разработать алгоритм приближённого вычисления площади криволинейной фигуры, ограниченной осью абсцисс, графиком заданной функции , и вертикальными прямыми, т.е. https://lh6.googleusercontent.com/h19S-zSZWGT0-XA2cS_L72ZkI4A1q7tJJi9yJIXIkazNZI_fVU0AWJ2PIrOwKNYLHQfhOuZ5I3xpRlODU6sYz7VVPHJAzFDBJFTTX13VR-gwZ5SIcHd6CpDCS_AXRsp1WMVuWPboTnM0B3qm3-OpZyzEKsQX3t76ERyypmV-kNy7JMmNL7_oTalpUJnhwWCsdhRPgZYMmg. Каждый отрезок функции представляется в виде прямоугольника c длиной отрезка (шагом)  (см. рис. ниже). Затем площадь под кривой вычисляется по формуле  – левый прямоугольник, т.к. высота прямоугольника берется как значение функции в крайней левой 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

10.2 Решение задачи, код программы

import static java.lang.Math.\*;  
import java.util.Scanner;  
  
public class main6\_10 {  
 public static void main(String[] args) {  
 double h;  
 double sum;  
 double n;  
 for (n = 10; n <= 10000; n \*= 10) {  
 h = 1 / n;  
 sum = 0;  
 for (int i = 1; i < n; i++) {  
 sum += (8 \* Math.*pow*(n - h \* (i-1), 5) - Math.*sin*(n - h \* (i - 1))) \* h;  
 }  
 System.*out*.println("n " + n + " h " + h + " sum " + sum );  
  
 }  
 }  
}

10.3 Тестирование работы программы