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 1.

Выполнил: студент группы ИСТ-23-1Б Шустов А.В.

Проверил: Нетбай Георгий Владимирович.

Пермь, 2023

Оглавление

[Задание 1 4](#_Toc162974312)

[1.1 Постановка задачи 4](#_Toc162974313)

[1.2 Решение на Java 4](#_Toc162974314)

[1.3 Проверка результатов 4](#_Toc162974315)

[Задание 2 4](#_Toc162974316)

[2.1 Постановка задачи 4](#_Toc162974317)

[2.2 Решение на Java 4](#_Toc162974318)

[2.3 Проверка результатов 5](#_Toc162974319)

[Задание 3 5](#_Toc162974320)

[3.1 Постановка задачи 5](#_Toc162974321)

[3.2 Решение на Java 5](#_Toc162974322)

[3.3 Проверка результатов 6](#_Toc162974323)

[Задание 4 6](#_Toc162974324)

[4.1 Постановка задачи 6](#_Toc162974325)

[4.2 Решение на Java 6](#_Toc162974326)

[4.3 Проверка результатов 7](#_Toc162974327)

[Задание 5 7](#_Toc162974328)

[5.1 Постановка задачи 7](#_Toc162974329)

[5.2 Решение на Java 7](#_Toc162974330)

[5.3 Проверка результатов 8](#_Toc162974331)

[Задание 6 8](#_Toc162974332)

[6.1 Постановка задачи 8](#_Toc162974333)

[6.2 Решение на Java 8](#_Toc162974334)

[6.3 Проверка решения 9](#_Toc162974335)

[Задание 7 9](#_Toc162974336)

[7.1 Постановка задачи 9](#_Toc162974337)

[7.2 Решение на Java 10](#_Toc162974338)

[7.3 Проверка результатов 10](#_Toc162974339)

[Задание 8 10](#_Toc162974340)

[8.1 Постановка задачи 10](#_Toc162974341)

[8.2 Решение на Java. 10](#_Toc162974342)

[8.3 Проверка результатов 11](#_Toc162974343)

[Задание 9 11](#_Toc162974344)

[9.1 Постановка задачи 11](#_Toc162974345)

[9.2 Решение на Java 11](#_Toc162974346)

[9.3 Проверка результатов 12](#_Toc162974347)

# 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 на Java

import java.util.Scanner;  
  
public class Task1 {  
 public static void main(String[] args) {  
 var scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Введите значение x: ");  
 double x = scanner.nextDouble();  
  
 System.*out*.print("Введите количество членов ряда (N): ");  
 int N = scanner.nextInt();  
  
 System.*out*.print("Введите значение M: ");  
 double M = scanner.nextDouble();  
  
 double sum = 0.0;  
 double sumAnother = 0.0;  
  
 for (int n = 0; n < N; n++) {  
 double item = 1.0 / Math.*pow*(x, 2 \* n - 2);  
 sum += item;  
 if (item < M) {  
 sumAnother += item;  
 }  
 }  
  
 System.*out*.println("Сумма первых " + N + " членов ряда: " + sum);  
 System.*out*.println("Сумма членов ряда, меньших " + M + ": " + sumAnother);  
 }  
}

## 1.3 Проверка результатов

Для проверки результатов, заполним столбец A значениями n, а в столбец B введём формулу =1/СТЕПЕНЬ($E$1; 2 \* $A2 - 2). Для подсчёта суммы первых N членов ряда и суммы членов меньше M используем формулы =СУММЕСЛИ(A2:A1000;"<" & $E$2;B2:B1000) и =СУММЕСЛИМН(B2:B1000;A2:A1000;"<" & $E$2;B2:B1000;"<" & $E$3) соответственно.

|  |  |
| --- | --- |
| Java | Excel |
|  |  |

# Задание 2

## 2.1 Постановка задачи

Подсчитать количество “счастливых” билетов, у которых сумма трех цифр равна 13.

## 2.2 Решение на Java

public class Task2 {  
 public static void main(String[] args) {  
 int count = 0;  
 for (int i = 0; i <= 999; i++) {  
 for (int j = 0; j <= 999; j++) {  
 int firstPart = i;  
 int secondPart = j;  
  
 int firstSum = 0;  
 int secondSum = 0;  
  
 while (firstPart > 0) {  
 firstSum += firstPart % 10;  
 firstPart /= 10;  
 }  
  
 while (secondPart > 0) {  
 secondSum += secondPart % 10;  
 secondPart /= 10;  
 }  
   
 if (firstSum == 13 && secondSum == 13) {  
 count++;  
 }  
 }  
 }  
  
 System.*out*.println("Количество счастливых билетов: " + count);  
 }  
}

## 2.3 Проверка результатов

|  |  |
| --- | --- |
| Java | Ожидание |
| 5625 | 5625 |

# Задание 3

## 3.1 Постановка задачи

Написать программу, вычисляющую следующие выражения (рис. 1).



Рисунок 1

## 3.2 Решение на Java

public class Task3 {  
 public static void main(String[] args) {  
 System.*out*.println("Sum1: " + *Sum1*());  
 System.*out*.println("Sum2: " + *Sum2*());  
 System.*out*.println("Sum3: " + *Sum3*());  
 }  
 public static double Sum1() {  
 var sum = 0f;  
 for (var i = 1; i <= 8; i ++) {  
 for (var j = 1; j <= i; j++) {  
 sum += j \* j;  
 }  
 }  
 return sum;  
 }  
 public static double Sum2() {  
 var sum = 0d;  
 for (var i = 1; i <= 8; i++) {  
 var localProduct = 1d;  
 for (var j = 1; j <= i \* 2; j++) {  
 localProduct \*= (Math.*pow*(j, 3) + Math.*pow*(i, 2));  
 }  
 sum += localProduct;  
 }  
 return sum;  
 }  
 public static double Sum3() {  
 var sum = 0;  
 for (var i = 1; i <= 8; i++) {  
 for (var j = 1; j <= i; j++) {  
 for (var k = 1; k <= 3; k++) {  
 sum += (j \* j + i + k);  
 }  
 }  
 }  
 return sum;  
 }  
}

## 3.3 Проверка результатов

|  |  |
| --- | --- |
| Java | Расчёты |
|  | Да |

# Задание 4

## 4.1 Постановка задачи

Написать программу перевода целого десятичного числа из десятичной системы счисления в систему счисления с основанием, которое пользователь вводит с клавиатуры. 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 на Java

import java.util.Scanner;  
  
public class Task4 {  
 public static void main(String[] args) {  
 var scanner = new Scanner(System.*in*);  
  
 while (true) {  
 System.*out*.print("Enter decimal number: ");  
 var number = scanner.nextInt();  
  
 var base = 1;  
 while (true) {  
 System.*out*.print("Enter base: ");  
 base = scanner.nextInt();  
 if (base >= 2 && base <= 35) { // 9 numbers + 26 letters  
 break;  
 }  
 System.*out*.println("Error!");  
 }  
 System.*out*.println(*DecimalToBase*(number, base));  
  
 System.*out*.println("Do you want to continue? (true/false)");  
 if (!scanner.nextBoolean()) {  
 break;  
 }  
 }  
 }  
 public static String DecimalToBase(int decimalNumber, int base) {  
 var result = "";  
 while (decimalNumber != 0) {  
 var x = (decimalNumber % base);  
 result = (x < 10 ? Integer.*toString*(x) : ((char)((int)'A' + x - 10)) ) + result;  
 decimalNumber /= base;  
 }  
 return result;  
 }  
}

## 4.3 Проверка результатов

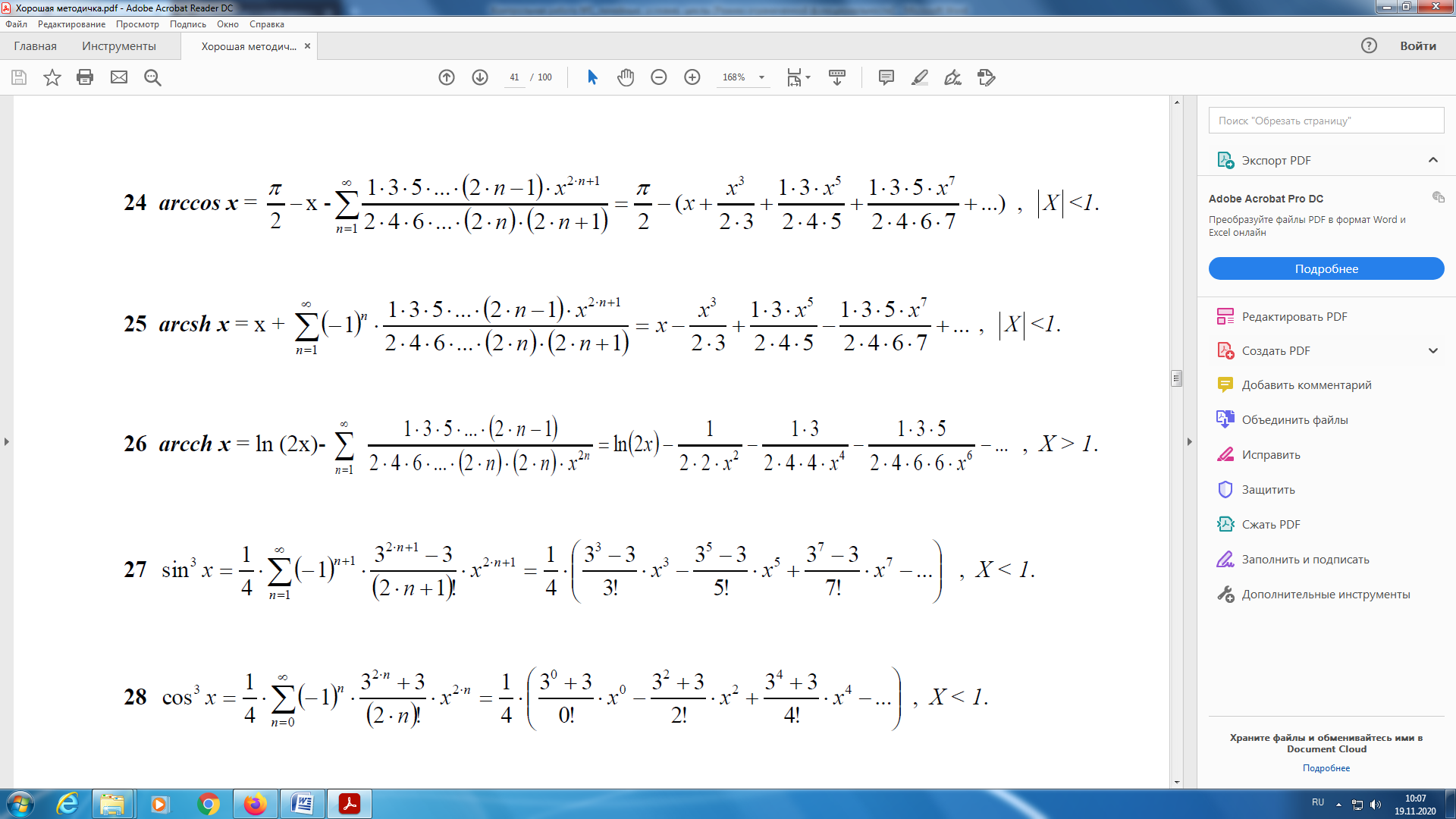
Проверим результаты используя Excel и формулы =ДЕС.В.ДВ(A2) и =ДЕС.В.ШЕСТН(A3).

|  |  |
| --- | --- |
| Java | Ожидание |
|  |  |
|  |
|  |

# Задание 5

## 5.1 Постановка задачи

Написать программу для вычисления  через ряд Маклорена с погрешностью 10-2 – 10-4, остановка итерационной процедуры .



## 5.2 Решение на Java

import java.util.Scanner;  
  
public class Task5 {  
 public static void main(String[] args) {  
 var scanner = new Scanner(System.*in*);  
 var x = scanner.nextDouble();  
  
 var target = Math.*pow*(Math.*cos*(x), 3);  
 System.*out*.println("Target: " + target);  
 for (double n = Math.*pow*(10, 2); n <= Math.*pow*(10, 4); n++) {  
 var result = *Calc*(x, 1/n);  
 System.*out*.println("При погрешности " + (1/n) + "; \n F = " + result + "; \n Итоговая погрешность: " + (target - result));  
 }  
 System.*out*.println("Target: " + target);  
 }  
 public static long Factorial(long n) {  
 long result = 1;  
 for (long i = 1; i <= n; i++) {  
 result \*= i;  
 }  
 return result;  
 }  
 public static double Calc(double x, double e) {  
 var target = Math.*pow*(Math.*cos*(x), 3);  
  
 var sum = 0d;  
 for (int n = 0; Math.*abs*(target - (sum/4d)) > e; n++) {  
 sum += Math.*pow*(-1, n) \* ((Math.*pow*(3, 2\*n) + 3) / *Factorial*(2 \* n)) \* Math.*pow*(x, 2\*n);  
 }  
 return sum / 4d;  
 }  
}

## 5.3 Проверка результатов

Для проверки посчитаем функцию cos^3(x) в Excel и сравним результаты.

|  |  |
| --- | --- |
| Java | Ожидание |
|  |  |
|  |
|  |

# Задание 6

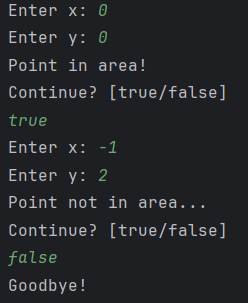
## 6.1 Постановка задачи

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 на Java

import java.util.Scanner;  
  
public class Task6 {  
 public static void main(String[] args) {  
 var scaner = new Scanner(System.*in*);  
  
 while (true) {  
 System.*out*.print("Enter x: ");  
 var x = scaner.nextDouble();  
 System.*out*.print("Enter y: ");  
 var y = scaner.nextDouble();  
 var result = Oblast.*Oblast*(x, y);  
 if (result) {  
 System.*out*.println("Point in area!");  
 } else {  
 System.*out*.println("Point not in area...");  
 }  
  
 System.*out*.println("Continue? [true/false]");  
 var c = scaner.nextBoolean();  
 if (!c) {  
 System.*out*.println("Goodbye!");  
 break;  
 }  
 }  
 }  
  
 public static class Oblast {  
 public static boolean Oblast(double x, double y) {  
 var inArea = false;  
  
 if (y > 0) {  
 return false;  
 }  
 if (*inCircle*(0, 0, 6, x, y)) { // main  
 inArea = true;  
 }  
 if (*inCircle*(2, -4, 1, x, y)) { // small empty  
 inArea = false;  
 }  
 if (*inCircle*(-1, -3, 2, x, y)) { // 2, big empty  
 inArea = *inCircle*(-1, -2, 1, x, y); // 3, in 2  
 }  
 if ((x >= -4) && (x <= -1) && (y > (x + 1)) && (y < 0) && (y > -3)) { // left tr  
 inArea = false;  
 }  
 if ((x >= -1) && (x <= 4) && (y > (x \* -0.75d - 0.75d)) && (y > (x \* 3 - 12)) && (y < 0) && (y > -3)) {  
 inArea = false;  
 }  
  
 return inArea;  
 }  
 private static boolean inCircle(double circleX, double circleY, double r, double pointX, double pointY) {  
 return Math.*sqrt*(Math.*pow*(circleX - pointX, 2) + Math.*pow*(circleY - pointY, 2)) <= r;  
 }  
 }  
}

## 6.3 Проверка решения



# Задание 7

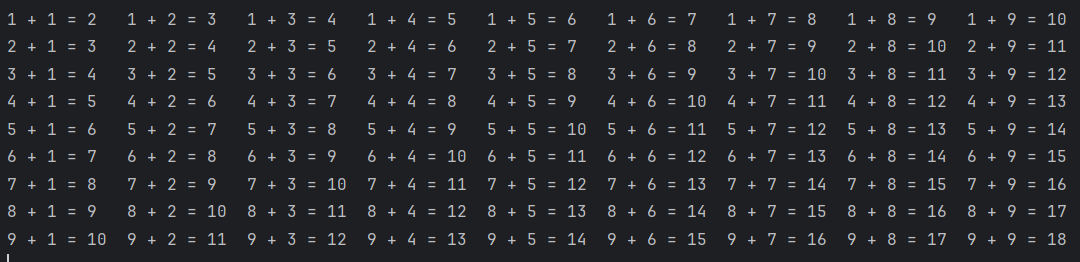
## 7.1 Постановка задачи

Напечатать полную таблицу сложения.

## 7.2 Решение на Java

public class Task7 {  
 public static void main(String[] args) {  
 for (int i = 1; i <= 9; i++) {  
 for (int j = 1; j <= 9; j++) {  
 int sum = i + j;  
 System.*out*.print(i + " + " + j + " = " + sum + "\t");  
 }  
 System.*out*.println();  
 }  
 }  
}

## 7.3 Проверка результатов



# 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 на Java.

import java.util.Scanner;  
  
public class Task8 extends MyFunction {  
 public static void main(String[] args) {  
 var scanner = new Scanner(System.*in*);  
 var task = new Task8();  
  
 final var r = 1d;  
 final var y = 1d;  
  
 var sum = 0d;  
 for (var i = 0; i < 10; i++) {  
 System.*out*.print("["+(i+1)+"] Enter value: ");  
 var t = scanner.nextDouble();  
 var result = task.FunctionMy(t, r, y);  
 sum += result;  
 }  
 System.*out*.println("Sum: " + sum);  
 }  
}

import java.util.Scanner;  
  
public class MyFunction {  
 public double FunctionMy(double t, double r, double y) {  
 return (4 \* Math.*pow*(t, 3) + Math.*log*(r)) / (Math.*pow*(Math.*E*, y + r) + 7.2d \* Math.*sin*(r));  
 }  
}

## 8.3 Проверка результатов

Проверим при помощи Excel и формулы =(4 \* A4^3 + LN($B$2)) / (СТЕПЕНЬ(EXP(1); $C$2 + $B$2) + 7,2 \* SIN($B$2))

|  |  |
| --- | --- |
| Java | Excel |
|  |  |

# Задание 9

## 9.1 Постановка задачи

Написать программу вычисление корня р-й степени в рамках итерационной процедуры . Для определения используется итерационная процедура на основе формулы Ньютона.

## 9.2 Решение на Java

import java.util.Scanner;  
  
public class Task9 {  
 public static void main(String[] args) {  
 var scanner = new Scanner(System.*in*);  
  
 System.*out*.print("Enter x: ");  
 var x = scanner.nextInt();  
 System.*out*.print("Enter p: ");  
 var p = scanner.nextInt();  
  
 System.*out*.println("Точность; Результат; Итераций;");  
 for (double n = -2; n >= -6; n--) {  
 var e = Math.*pow*(10, n);  
  
 var y = Math.*exp*(Math.*log*(x \* (p + 1)) / p); // y0  
 var yLast = 0d + e \* 2;  
 var iterCount = 0;  
 while (Math.*abs*(y - yLast) > e) {  
 yLast = y;  
 y = *Func*(p, x, y);  
 iterCount++;  
 }  
 System.*out*.println(e + "; " + y + "; " + iterCount);  
 }  
 }  
 private static double Func(double p, double x, double y) {  
 return (1/p) \* ((p - 1) \* y + (x / Math.*pow*(y, p - 1)));  
 }  
}

## 9.3 Проверка результатов

Воспользуемся Excel и формулой =A2^(1/B2), где A2 – это X, а B2 – это P.

|  |  |
| --- | --- |
| Java | Excel |
|  |  |
|  |