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 Вихляев И.С.

6 вариант

# 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 static java.lang.Math.\*;

import java.util.Scanner;

public class Task1 {

    public static void main(String[] args) {

        double s = 0, x = 2;

        Scanner sc = new Scanner(System.in);

        System.out.print("Write number n : ");

        Integer n = sc.nextInt();

        sc.close();

        for (int i = 1; i < n + 1; i++) {

            s = s + pow(x, 2 \* i - 1) / (2 \* i - 1);

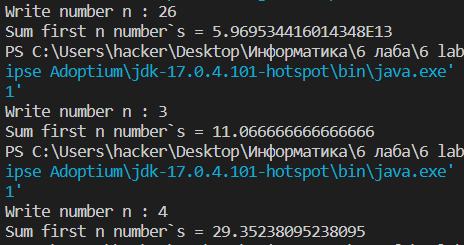
        }

        System.out.println("Sum first n number`s = " + s);

    }

}

## 1.3. Тестирование кода



# Задание 2

## 2.1. Постановка задачи

Каждая бактерия делится на две в течение одной минуты. В начальный момент имеется одна бактерия. Составьте программу, которая при заданном числе бактерий и начальном числе бактерий вычисляет, за сколько по времени в минутах и часах произошел рост популяции бактерий.

## 2.2. Решение задачи, код программы

import java.util.Scanner;

public class Task2 {

    public static void main(String[] args) {

        int hour = 0, minutes = 1;

        Scanner sc = new Scanner(System.in);

        System.out.print("Write quantity bacteries on start : ");

        Integer n = sc.nextInt();

        System.out.print("Write quantity bacteries on before time : ");

        Long s = sc.nextLong();

        sc.close();

        while (s > n) {

            s = s / 2;

            minutes += 1;

        }

        while (minutes > 60) {

            hour += 1;

            minutes = minutes - 60;

        }

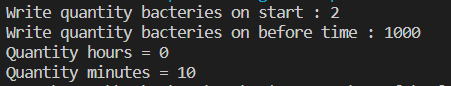
        System.out.println("Quantity hours = " + hour);

        System.out.println("Quantity minutes = " + minutes);

    }

}

## 2.3. Тестирование кода



# Задание 3

## 3.1. Постановка задачи

Написать программы, которые вычисляют выражения:

1. 
2. 
3. 

## 3.2. Решение задачи, код программы для 1.

public class Task31 {

    public static void main(String[] args) {

        int s = 0;

        for (int i = 1; i < 9; i++) {

            for (int j = 1; j < 9; j++) {

                if (i != j) {

                    s += j \* j;

                }

            }

        }

        System.out.println("final answer = " + s);

    }

}

## 3.3. Тестирование кода программы для 1



## 3.2. Решение задачи, код программы для 2.

public class Task32 {

    public static void main(String[] args) {

        int s = 0, s1, answer = 0;

        for (int i = 1; i < 9; i++) {

            s1 = 1;

            for (int j = 1; j < 4; j++) {

                s = (j \* j \* j) + (i \* i);

                s \*= s1;

                s1 = s;

            }

            answer += s;

        }

        System.out.println("final answer = " + answer);

    }

}

## 3.3. Тестирование кода программы для 1



## 3.2. Решение задачи, код программы для 3.

public class App {

    public static void main(String[] args) {

        int s = 0, s1, answer = 0;

        for (int i = 1; i < 9; i++) {

            s1 = 1;

            for (int j = 1; j < 4; j++) {

                s = (j \* j \* j) + (i \* i);

                s \*= s1;

                s1 = s;

            }

            answer += s;

        }

        System.out.println("final answer = " + answer);

    }

}

## 3.3. Тестирование кода программы для 1



# Задание 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.\*;

public class Task4 {

    public static double perevod(double x,int s) {

        int xc = (int)x;

        double xd = x-xc;

        String x1 = "", x2 = "0.";

        while (xc>0) {

            x1+=Integer.toString(xc%s);

            xc/=s;

        }

        for (int a=0; a<10; a++) {

            xd=xd\*s-(int)xd\*s;

            x2+=Integer.toString((int)xd);

        }

        return Integer.parseInt(new StringBuilder(x1).reverse().toString())+Double.parseDouble(new StringBuilder(x2).toString());

    };

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.println("Введите дробное число");

        double a = in.nextDouble();

        int s=0;

        while (s<1 | s>10) {

            System.out.println("Введите систему счисления (1<=s<=10)");

            s=in.nextInt();

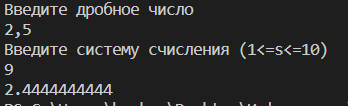
        }

        System.out.println(perevod(a,s));

    }

}

## 4.3. Тестирование кода программы для 1



# 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. Разобьём все фигуры по линиям и напишем уравнение для каждой линии

## 6.2. Решение задачи, код программы

import java.util.\*;

import static java.lang.Math.\*;

public class Task6 {

    public static boolean oblast(double x, double y) {

        if (pow(x,2)+pow(y,2)<=36 & x>=0 & !(y<=4 & y>=x+1) &

                !(y<=-4f/3\*x+1 & y>=1f/3\*x-4) & !(pow(x-4,2) + pow(y+2,2)<=1) &

                !((pow(x-3,2) + pow(y-1,2)<=4) & !(pow(x-2,2)+pow(y-1,2)<=1))) {

            return true;

        }

        else {

            return false;

        }

    }

    public static void main(String[] args) {

        double x=0, y=0;

        Scanner in = new Scanner(System.in);

        boolean stop=false;

        while (!stop) {

            System.out.println("Введите \"stop\", если хотите прекратить поиск координат.");

            String vvod = in.nextLine();

            if (vvod.equals("stop")) {

                stop = true;

            } else {

                System.out.println("Введите координаты х");

                x = Double.parseDouble(in.nextLine());

                System.out.println("Введите координаты y");

                y = Double.parseDouble(in.nextLine());

                if (oblast(x, y)) {

                    System.out.println("Точка попала в область");

                } else {

                    System.out.println("Точка не попала в область");

                }

            }

        }

    }

}

# 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. Решение задачи, код программы

public class Task7 {

    public static void main(String[] args) {

        for (int i = 1; i < 10; i++) {

            for (int n = 1; n < 10; n++) {

                System.out.print(i + "x" + n + "=" + n \* i + ' ');

            }

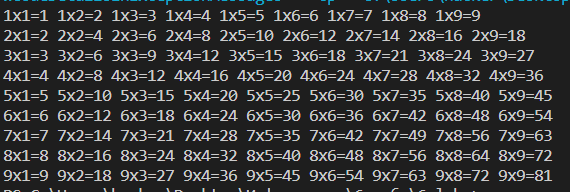
            System.out.println();

        }

    }

}

## 7.3. Тестирование кода 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;

public class Task8 {

    public static double FunctionMy(double a1, double razn) {

        int s = 0;

        for (int i = 1; i <= 10; i++) {

            s += a1;

            a1 += razn;

        }

        return s;

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Введите a1");

        double a1 = (sc.nextInt());

        System.out.println("Введите a5");

        double a5 = sc.nextInt();

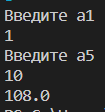
        double razn = (a5 - a1) / 4;

        System.out.println(FunctionMy(a1, razn));

    }

}

## 8.3. 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 java.util.\*;

import static java.lang.Math.\*;

public class Task9 {

    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;

                }

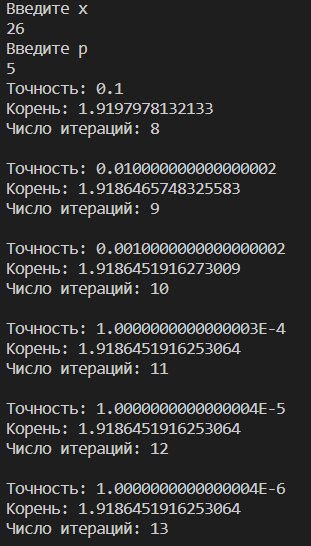
            }

        }

    }

}

## 9.3. Тестирование кода 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  – центральный прямоугольник, т.к. высота прямоугольника берется как значение функции в центре 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 центральный прямоугольник

## 10.2. Решение задачи, код программы

import static java.lang.Math.\*;

public class Task10 {

    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 += (20\*pow(n-h\*i,2)-5\*pow(sin(n-h\*i)+(n-h\*i),2)) \* h;

            }

            System.out.println(n + "\n" + h + "\n" + sum + "\n5.18\n" + abs(sum - 5.18)+ "\n");

        }

    }

}

## 10.3. Тестирование кода программы

