Проект «Запутанная программа»

Проект был выдан преподавателем Высшей Школы Экономики - Плаксиным Михаилом Александровичем для домашнего изучения. Сделано студентом первого курса ПИ-18-2, Чепоковым Елизаром.

**Задачи проекта:**

1. Что делает программа P?
2. Определить смысл запутанной программы.
3. Ответить на вопрос: зачем, с какой целью было дано это задание.

**Решение:**

1. Программа P вычисляет число **π** (pi) с указанной точностью и выводит информацию о точности числа, самом **π** и количестве операций (начиная с 1 и умноженного на 2) для выведения числа.
2. Изначально был задан код:

program p;

var ol, o1, tmp: real; {!!!}

ll, i1: integer;

l1: integer;

begin

readln(o1);

o1:= o1\*0.10; {ghbrjk!}

ol:= 1;

ll:= 1;

l1:= -1;

repeat tmp:= Ol; {временно}

ll:=ll + 2;

ol:= tmp+l1/ll;

l1:= -l1;

until abs(ol-tmp) < O1;

writeln(o1\*10{маразм},ol\*4,ll);

end.

Я его преобразил, для наглядности и более легкого использования:

program p;

var b, a, tmp: real;

c: integer;

d: integer;

begin

readln(a); {Ввод переменной a}

a:= a\*0.10; {Умножение a на 0.1}

b:= 1; {Присваивание переменной b значения 1}

c:= 1; {Присваивание переменной c значения 1}

d:= -1; {Присваивание переменной d значения -1}

repeat tmp:= b; {Начало цикла}

c:=c + 2; {Счет операций}

b:= tmp+d/c; {Частный случай сложения arctg 1}

d:= -d; {Смена переменной (грани ОДЗ)}

until abs(b-tmp) < a; {Конец цикла}

writeln(a\*10); {Вывод данных о точности}

writeln(b\*4); {Вывод данных о числе pi}

writeln(c); {Количество операций = (с-1)/2}

end.

Вот схема программы, для наглядности (для правильного отображения нужно разрешить редактирование):

Ввод a

Начало

α

b = 1

a = a\*0.10

c = 1

Остановка

Вывод c

Вывод b\*4

Вывод а\*10

Конец цикла

|c = 1| < a

d = -d

Нет

Да

b = tmp + d/c

c = c + 2

Начало цикла tmp= b

α

Сначала вводятся все данные, прогоняются в цикле и выводятся в виде ответа.

1 число — это вводимая аккуратность чем ближе число к 0, тем точнее число **π**. Сам 0 и числа <0 не используются.

2 число — это число **π** (частный случай разложения arctg pi/4, умноженное на 4).

3 число — это количество проведённых циклов, умноженное на 2 и +1.

Максимальная допустимая аккуратность — это 0.00000001 или 1E-08 далее программа перестает вычислять из-за большого количества итераций и на долго зависает.

При вводе чисел ≥ 3.4 программа начинает выдавать один и тот же ответ b = 2.66666666666667.

Так же есть предположение что программа вычисляет сумму последовательности (-1) ^n/(2n+1) при n = 0, 1, 2, 3, 4, …, n+1, … умноженную на 4, пока изменение по модулю не будет <o1/10, но я не совсем уверен в этом предположении.

1. Сначала стоит упомянуть о минусах, таких как, почти одинаковые переменные, не разборчивае в прогрмме, лишняя переменная i1, которая только занимает место, отсутствие комментариев и неразборчивый выход информации, который стоит разделить.

Это задание было дано для понимания, что вне зависимости как криво мы напишем программу она должна работать правильно.

Так же она дана для того чтобы понять, как трудно бывает работать с чужими программами, без внятных пояснений и марок.

Она дана с целью обучить, как важно правильно оформлять программы. Так же она дана чтобы понять, что любую программу можно сделать удобной и понятной.