Лабораторная работа № 7

**Тема:** Итерационные ЦВП с управлением по функции

**Цель:** Научиться создавать алгоритмы, используя итерационные ЦВП с управлением по функции

**Оборудование:** ПК, PascalABC.NET

Задача №1

**Постановка задачи:** С клавиатуры вводится трехзначное число, считается сумма его цифр. Если сумма цифр числа больше 10, то вводится следующее трехзначное число, если сумма меньше либо равна 10 – программа завершается.

**Математическая модель:** Для этой задачи можно написать 2 алгоритма: первый только для трехзначных чисел, второй для любых.

В первом случае сумма (s) цифр в трехзначном числе n находится как

Во втором случае сумма цифр будет накапливаться в цикле. Параметром цикла будет само число, и пока это число не примет значение «0» мы будем считать значение суммы (s) как:

А параметр цикла будет изменяться:

**Блок-схема:**

Первый случай:



Второй случай:



**Список идентификаторов:**

1)

|  |  |  |
| --- | --- | --- |
| a | исходное число | integer |
| i | параметр цикла | integer |

2)

|  |  |  |
| --- | --- | --- |
| x | исходное число | integer |
| i | параметр цикла | integer |
| f | функция суммы цифр | integer |
| s | переменная в функции | integer |

**Код программы:**

1)

**var**

a: integer;

**begin**

readln(a);

**while** (a **mod** 10 + a **div** 100 + a **div** 10 **mod** 10) > 10 **do**

**begin**

writeln('+');

readln(a);

**end**;

**end**.

2)

**var**

x: integer;

**function** f(n: integer): integer;

**var**

s: integer;

**begin**

s := 0;

**while** n <> 0 **do**

**begin**

s := s + n **mod** 10;

n := n **div** 10;

**end**;

f := s

**end**;

**begin**

readln(x);

**while** f(x) > 10 **do**

**begin**

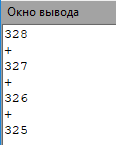
readln(x);

**end**;

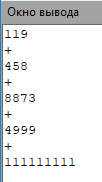
**end**.

**Результат выполненной работы:**

1)



2)



**Анализ результатов вычисления:**

Для решения задачи было написано 2 программы для подсчета суммы цифр в числе, первая подходит только для трехзначных чисел, но она задействует меньше переменных и ее проще реализовать, а вторая подходит для любых чисел, но сложнее в написании и более затратная по памяти

Задача №2

**Постановка задачи:** Решить уравнение методом Ньютона:

; на отрезке от -10 до 10 с точностью 10-6

**Математическая модель:** Метод ньютона заключается в том что мы строим касательные, постепенно приближаясь к точкам пересечения графика с осью абсцисс. Графики касательных строятся исходя из зависимости:

**Блок-схема:**



**Список идентификаторов:**

|  |  |  |
| --- | --- | --- |
| x,x1 | значения x | real |
| Fn | функция | real |
| E | точность | real |

**Код программы:**

**const**

E = 10e-6;

**var**

x, x1: real;

**function** Fn(x: real): real;

**begin**

Fn := x - (-x \* x \* x \* x - 18 \* x \* x + 6) / (-4 \* x \* x \* x - 36 \* x);

**end**;

**begin**

x1 := -10;

**repeat**

x := x1;

x1 := Fn(x);

**until** abs(x1 - x) <= E;

writeln(x1);

x1 := 10;

**repeat**

x := x1;

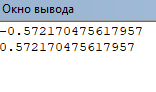
x1 := Fn(x);

**until** abs(x1 - x) <= E;

writeln(x1);

**end**.

**Результат выполненной работы:**



**Анализ результатов вычисления:**

Для упрощения вычислений зависимость была записана одной функцией «Fn»

**Вывод:** Средствами Pascal были созданы алгоритмы, в которых используются итерационные ЦВП с управлением по функции