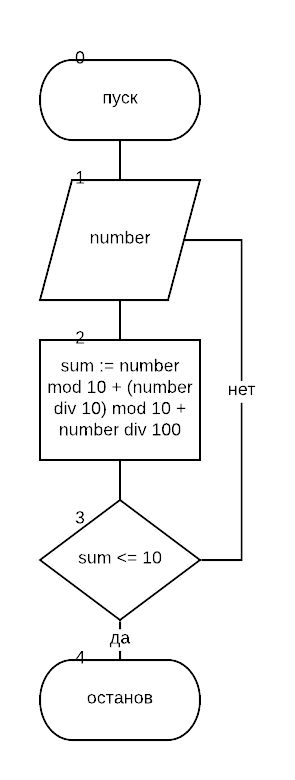
# Лабораторная работа № 7

## 1 задание

1. Итерационные циклические вычислительные процессы с управлением по функции.
2. Научится реализовать алгоритм для вычисление детерминированного ЦВП с управлением по индексу средствами компилятора и языка программирования Pascal.
3. ПК, PascalABC.NET 3.4.2.
4. 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.
5. sum := number mod 10 + (number div 10) mod 10 + number div 100
6. 

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| sum | Сумма | integer |
| number | Число, кот. вводит пользов. | integer |

**program** pr7;

**var**

number, sum: integer;

**begin**

**repeat**

**begin**

writeln('Введите целое трёхзначное число: ');

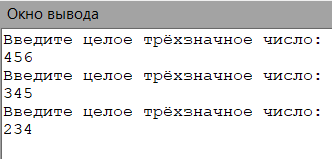
readln(number);

sum := number **mod** 10 + (number **div** 10) **mod** 10 + number **div** 100

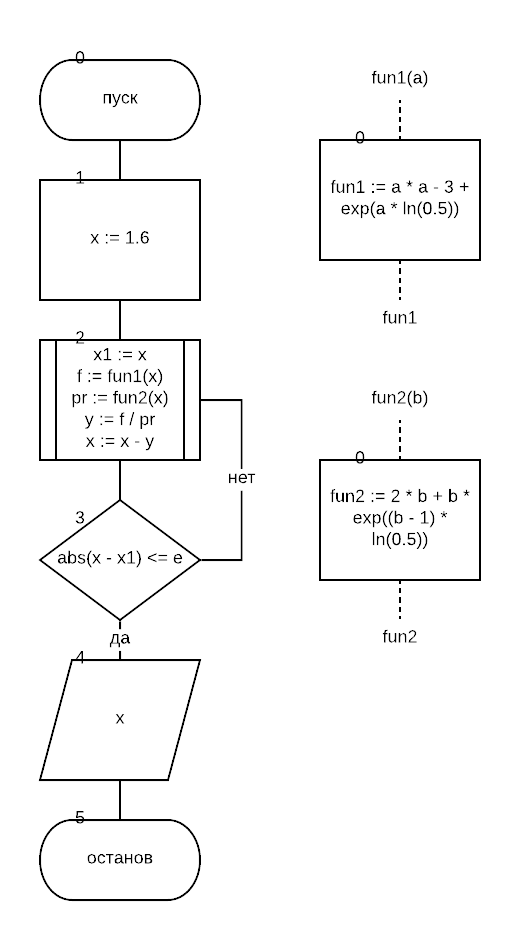
**end**;

**until** sum <= 10

**end**.

1. 
2. Реализовал программу с помощью оператора repeat…until, т. к. в этом случае код программы получился меньше, чем при использовании while.
3. Реализовал программу на вычисление суммы чисел для трёхзначного числа. Если сумма цифр числа больше 10, то вводится следующее трехзначное число, если сумма меньше либо равна 10 – программа завершается.

## 2 задание

1. Итерационные циклические вычислительные процессы с управлением по функции.
2. Научится реализовать алгоритм для вычисление итерационных циклических вычислительных процессов с управлением по функции средствами компилятора и языка программирования Pascal.
3. ПК, PascalABC.NET 3.4.2.
4. Решить нелинейное уравнение методом Ньютона.  
   
5. 
6. 

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| x | Решение | real |
| x1 | Предыдущее значение | real |
| f | Функция | real |
| pr | Производная | real |
| y | Отнош. функции к произв. | real |
| e | Точность | real |

**program** pr7;

**var**

x, x1, f, pr, y: real;

**const**

e = 0.000001;

**function** fun1(a:real):real;

**begin**

fun1 := a \* a - 3 + exp(a \* ln(0.5))

**end**;

**function** fun2(b:real):real;

**begin**

fun2 := 2 \* b + b \* exp((b - 1) \* ln(0.5))

**end**;

**begin**

x := 1.6;

**repeat**

**begin**

x1 := x;

f := fun1(x);

pr := fun2(x);

y := f / pr;

x := x - y;

**end**;

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

writeln(x:0:6)

**end**.

1. 
2. Реализовал программу с помощью оператора repeat…until, т. к. в этом случае код программы получился меньше, чем при использовании while.
3. Решил нелинейное уравнение методом Ньютона.