**Тема лабораторной работы:** [Итерационные циклические вычислительные процессы с управлением по функции](https://moodle.herzen.spb.ru/mod/url/view.php?id=36906)

**Цель л/р:** Научиться реализовывать ИЦВП с управлением по функции при помощи компилятора Pascal.

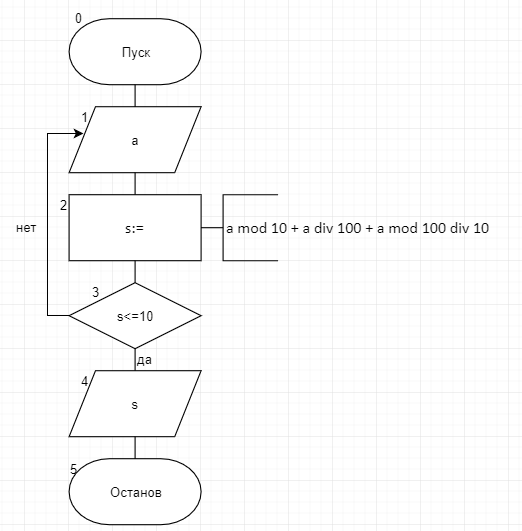
**Используемое оборудование:** PascalABC(код программы), draw.io(блок-схемы)

**Задание 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:** s = a mod 10 + a div 100 + a mod 100 div 10

**Блок-схема:**



**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| a | Вводимое число | integer |
| s | Сумма цифр числа | integer |

**Код программы:**

**program** lr71;

**var** a, s: integer;

**begin**

**repeat**

write('Трёхзначное число - ');

readln(a);

s := a **mod** 10 + a **div** 100 + a **mod** 100 **div** 10;

**until** s <= 10;

writeln('Сумма цифр числа - ', s);

**end**.

**Результат выполненной работы:**

Трёхзначное число - 444

Трёхзначное число - 158

Трёхзначное число - 145

Сумма цифр числа – 10

**Анализ результатов вычислений:**

Программа работает таким образом: если сумма цифр числа, вводимого с клавиатуры будет больше 10, цикл повторится, при выполнении условия сумма цифр подходящего числа будет выведена на экран.

**Задача 2**

**Постановка задачи**

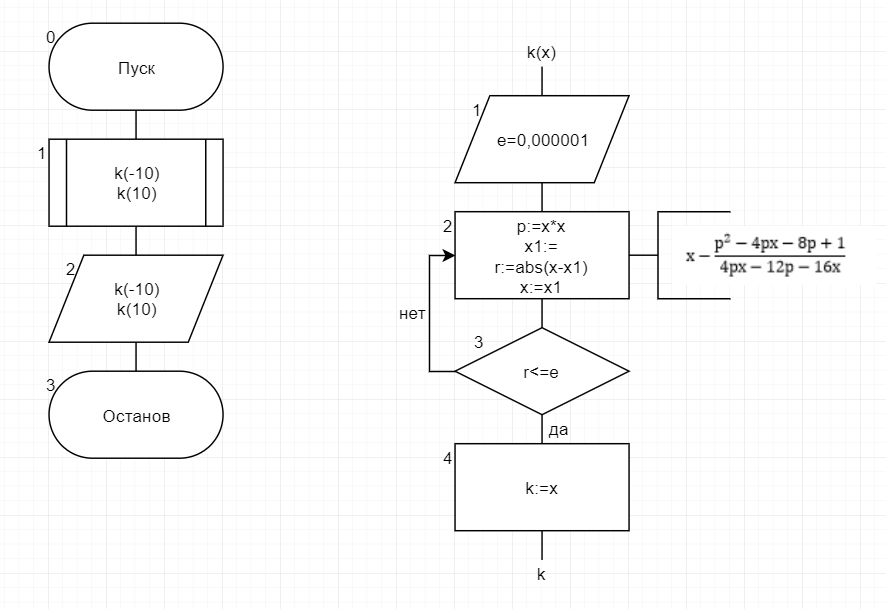
Решить нелинейное уравнение методом Ньютона

**Математическая модель**



x1 = x -

**Блок-схема**



**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| e | точность | real |
| x | начальное приближение | real |
| x1 | следующее приближение | real |
| r | промежуточная переменная | real |
| p | промежуточная переменная | real |
| k | значение функции | real |

**Код программы**

**program** lr72;

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

**const** e = 0.000001;

**var** x1, r, p: real;

**begin**

**repeat**

p := x\*x;

x1 := x - (p\*p - 4\*p\*x - 8\*p + 1) / (4\*p\*x - 12\*p - 16\*x);

r := abs(x1 - x);

x := x1;

**until** r <= e;

k := x;

**end**;

**begin**

writeln('Левая граница отрезка ', k(-10):1:5);

writeln('Правая граница отрезка ', k(10):1:5);

**end**.

**Результат выполненной работы**

Левая граница отрезка -1.38840

Правая граница отрезка 5.45926

**Анализ результатов вычислений**

Программа выполняется до тех пор, пока модуль разности следующего и предыдущего приближения не будет превышать точность.

**Вывод**

Была изучена реализация ИЦВП с управлением по функции при помощи компилятора Pascal.