

Лабораторная работа № 7 «Итерационные циклические вычислительные процессы с управлением по функции»

Цель: разработать и научиться использовать алгоритмы, основанные на итерационных циклических вычислительных процессах, управление которыми осуществляется по функции.

Оборудование: ПК, среда разработки «PascalABC»

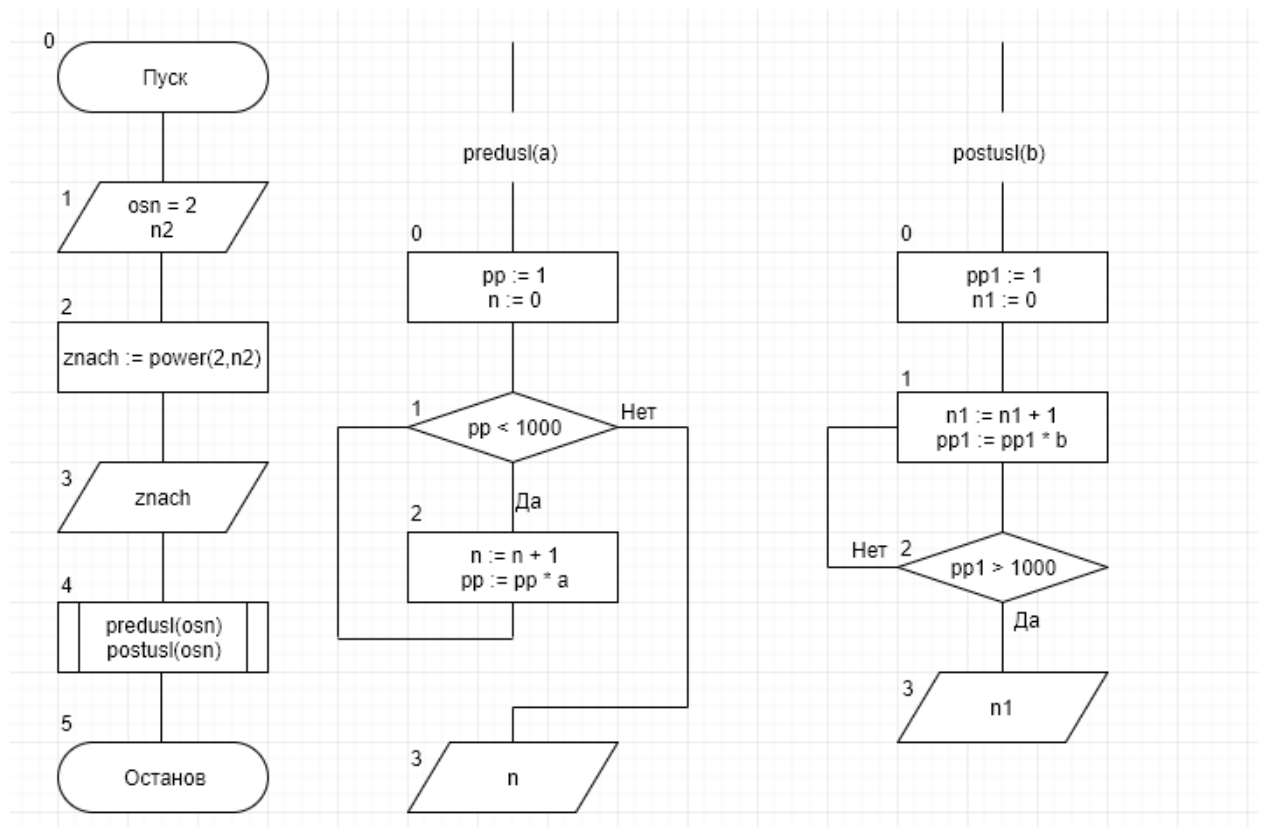
Задание 1

Постановка задачи: Вычислить 2 в степени n и при этом определить первое значение степени, при котором результат будет превышать значение 1000.

Математическая модель:

$$2^n = 2_1 * 2_2 * \dots * 2_n$$

Блок-схема:



Список идентификаторов:

Название	Тип	Функция
osn	integer	Хранение значения основания степени
n2	integer	Хранение значения показателя степени
znach	real	Хранение значения 2^{n2}
n	integer	Хранение значения минимального показателя степени n
pp	integer	Хранение числа равного 2^n
n1	integer	Хранение значения минимального показателя степени n1
pp1	integer	Хранение числа равного 2^{n1}
a	integer	Передача функции значения основания
b	integer	Передача функции значения основания

Код программы:

```
program Zadaniel;  
  Const  
    osn = 2;  
  Var  
    n2 : integer;  
    znach : real;  
  Procedure predusl(a :integer);  
    Var n : integer;  
    pp : integer;  
  Begin  
    pp := 1;  
    n := 0;  
    while pp < 1000 do  
      begin  
        n := n + 1;  
        pp := pp * a;  
      end;  
    writeln('Минимальное начение степени n := ', n , '(Цикл с предусловием)');  
  end;  
  Procedure postusl(b :integer);  
    Var n1 : integer;  
    ppl : integer;  
  Begin  
    n1 := 0;  
    ppl := 1;  
    repeat  
      n1 := n1 + 1;  
      ppl := ppl * b;  
    until ppl > 1000;  
    writeln('Минимальное начение степени n := ', n1, '(Цикл с постусловием)');  
  end;  
begin  
  writeln('Введите значение n');  
  readln(n2);  
  znach := power(osn,n2);  
  writeln('2 в степени n = ', znach);  
  predusl(osn);  
  postusl(osn);  
end.
```

Результаты вычислений:

Окно вывода

```
Введите значение n  
5  
2 в степени n = 32  
Минимальное начение степени n := 10 (Цикл с предусловием)  
Минимальное начение степени n := 10 (Цикл с постусловием)
```

Задание 2

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.

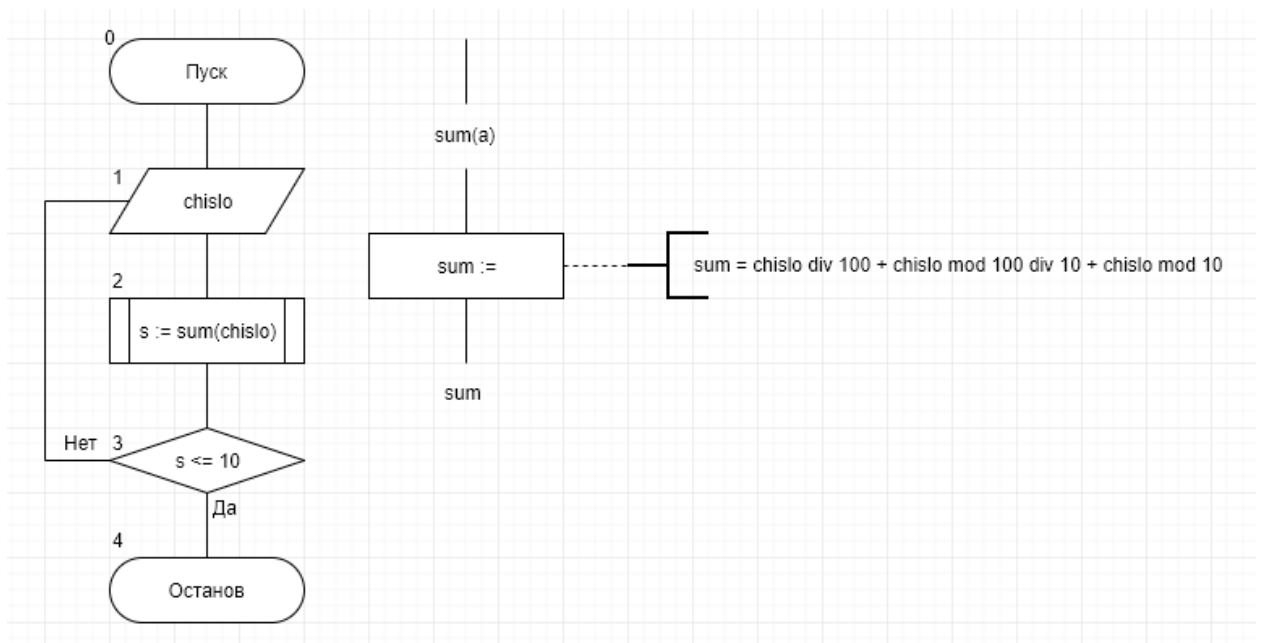
Математическая модель:

$a_1a_2a_3$ – введенное число

$$s = a_1 + a_2 + a_3$$

$$s \leq 10$$

Блок-схема:



Список идентификаторов:

Название	Тип	Функция
chislo	integer	Хранение значения введенного числа
s	integer	Хранение суммы чисел
a	integer	Передача функции значения введенного числа

Код программы:

```
Program Zadanie2;  
  Var  
    chislo,s  : integer;  
begin  
  repeat  
    writeln('Введите трехзначное число:');  
    readln(chislo);  
    s := chislo div 100 + chislo mod 100 div 10 + chislo mod 10;  
    until s <= 10  
  end.
```

Результаты вычислений:

```
Окно вывода  
Введите трехзначное число:  
346  
Введите трехзначное число:  
123
```

Задание 3

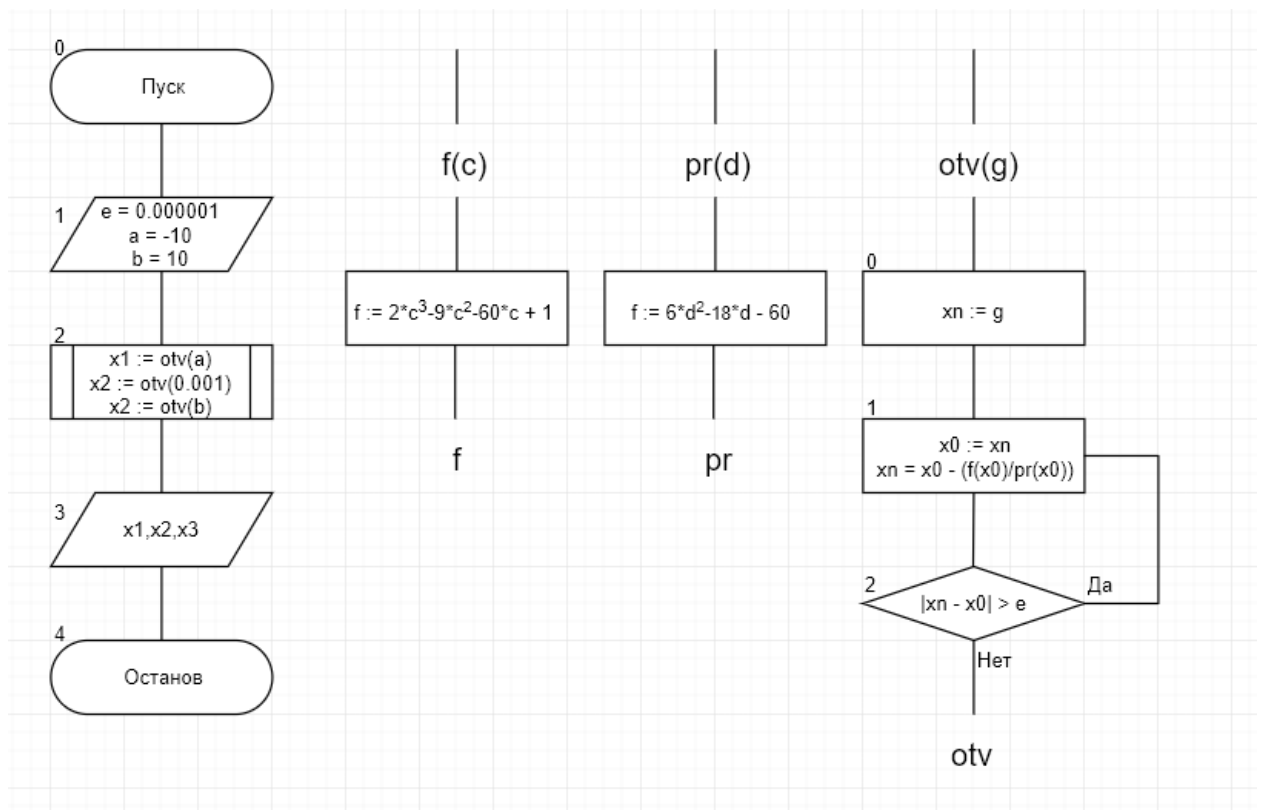
Постановка задачи: Решить нелинейное уравнение методом Ньютона:

$2x^3 - 9x^2 - 60x + 1 = 0$ на отрезке от -10 до 10 с точностью 10^{-6}

Математическая модель:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad |x_{n+1} - x_n| < \varepsilon$$

Блок-схема:



Список идентификаторов:

Название	Тип	Функция
e	real	Хранение значения точности
a	integer	Хранение значения левой границы интервала
b	integer	Хранение значения правой границы интервала
x1	real	Хранение значения первого корня
x2	real	Хранение значения второго корня
x3	real	Хранение значения третьего корня
c	real	Передача функции значения аргумента для подсчета значения функции

d	real	Передача функции значения аргумента для подсчета значения производной
g	real	Передача функции стартового значения корня
xn	real	Хранение значения более точного значения корня
x0	real	Хранение значения менее точного значения корня

Код программы:

```

Program zadanie3;
const
  e = 0.000001;
  a = -10;
  b = 10;
Var
  x1,x2, x3 : real;
function f(c :real) : real;
begin
  f := 2 * c * c * c - 9 * c * c - 60 * c + 1;
end;
function pr(d :real) : real;
begin
  pr := 6 * d * d - 18 * d - 60 ;
end;
function otv(g : real): real;
var xn,x0 : real;
begin
  xn := g;
  repeat
    x0 := xn;
    xn := x0 - f(x0) / pr(x0);
  until abs( xn - x0) > e;
  otv := xn;
end;
begin
  x1 := otv(a);
  x2 := otv(0.001);
  x3 := otv(b);
  writeln('x1 = ',x1,' x2 = ', x2,' x3 = ',x3);
end.

```

Результаты вычислений:

```

Окно вывода
x1 = -6.806944444444444 x2 = 0.0166618197202659 x3 = 8.608333333333333

```

Вывод: Использование итерационных циклических вычислительных позволяет решать раздел задач и умение их применять является обязательным для любого программиста. ИЦВП приходит на место ДЦВП и нужен для того, чтоб совершать циклические вычисления, о количестве

которых мы не можем сказать сразу. Реализовать их можно при помощи всех 3 конструкций циклов: «For», «while» и «repeat-until». Наиболее важны они для применения вычислительных методов математики.