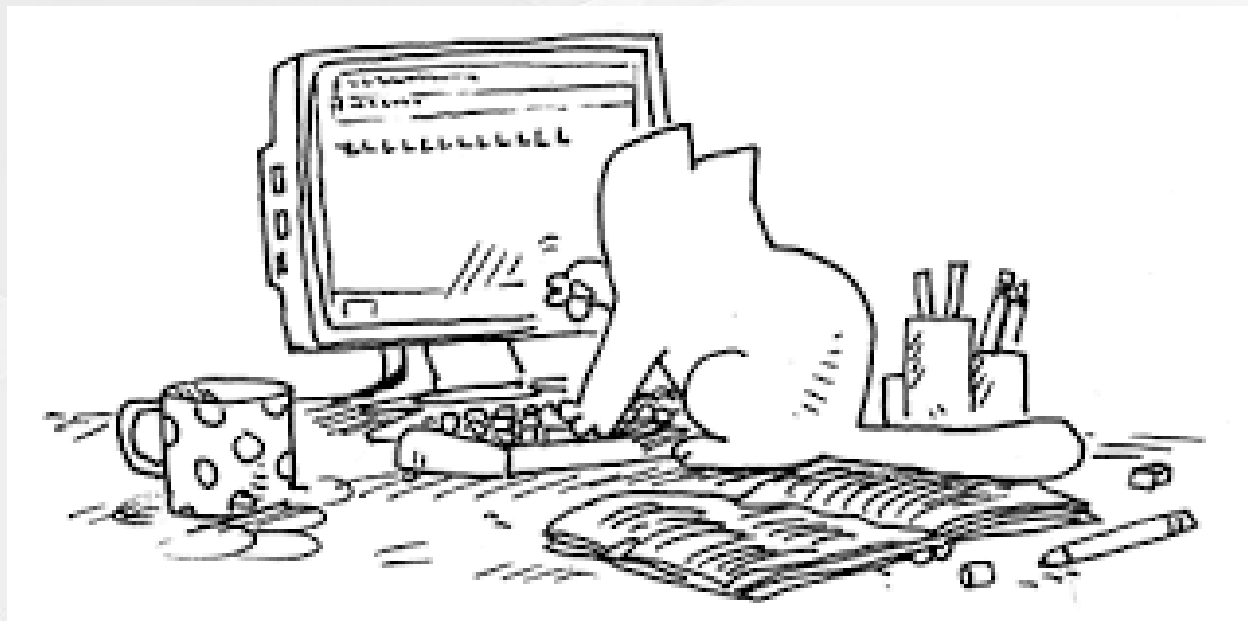


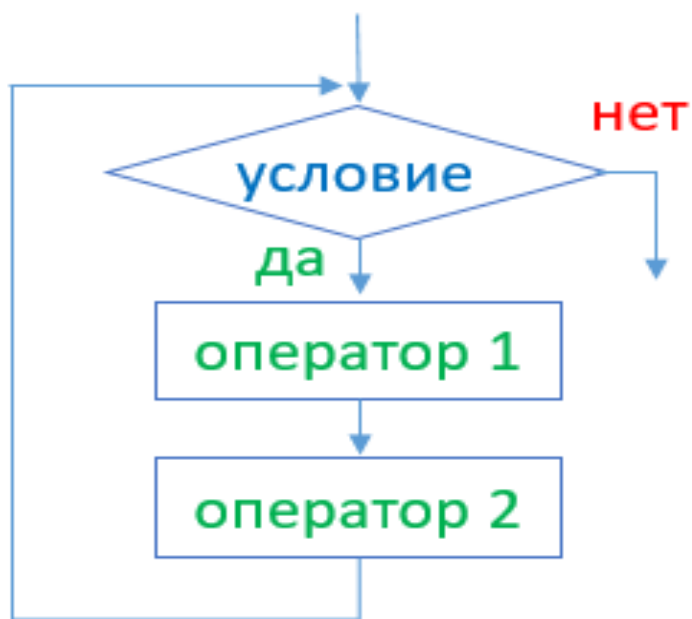
Цикл с предусловием (цикл ПОКА...)

Алгоритмическая конструкция **ПОВТОРЕНИЕ**



Цикл используется для многократного выполнения набора команд.

Цикл с предусловием Pascal (цикл-пока)



Использование более одного оператора в цикле:

```
while условие do  
begin  
оператор 1;  
оператор 2;  
end;
```

Один оператор в цикле:

```
while условие do оператор 1;
```

Цикл с предусловием (цикл Пока...)

- Операторы цикла будут повторяться до тех пор, пока **условие истинно**.
- Проверка условия делается каждый раз **перед выполнением операторов цикла**.
- Цикл *может ни разу не выполниться, если его условие ложно*.
- Если в цикле нужно выполнить **более 2-х команд**, то эти команды нужно записать в операторных скобках – перед записью команд поставить **begin**, а после команд – **end**.

Пример 1. Вывести n раз слово «ПРИВЕТ»

```
program primer1;  
var  
    i,n:integer;  
begin  
    i:=1;  
    write('введите количество раз '); read(n);  
    while i<=n do begin  
        write ('привет ');  
        i:=i+1;  
    end;  
end.  
|
```

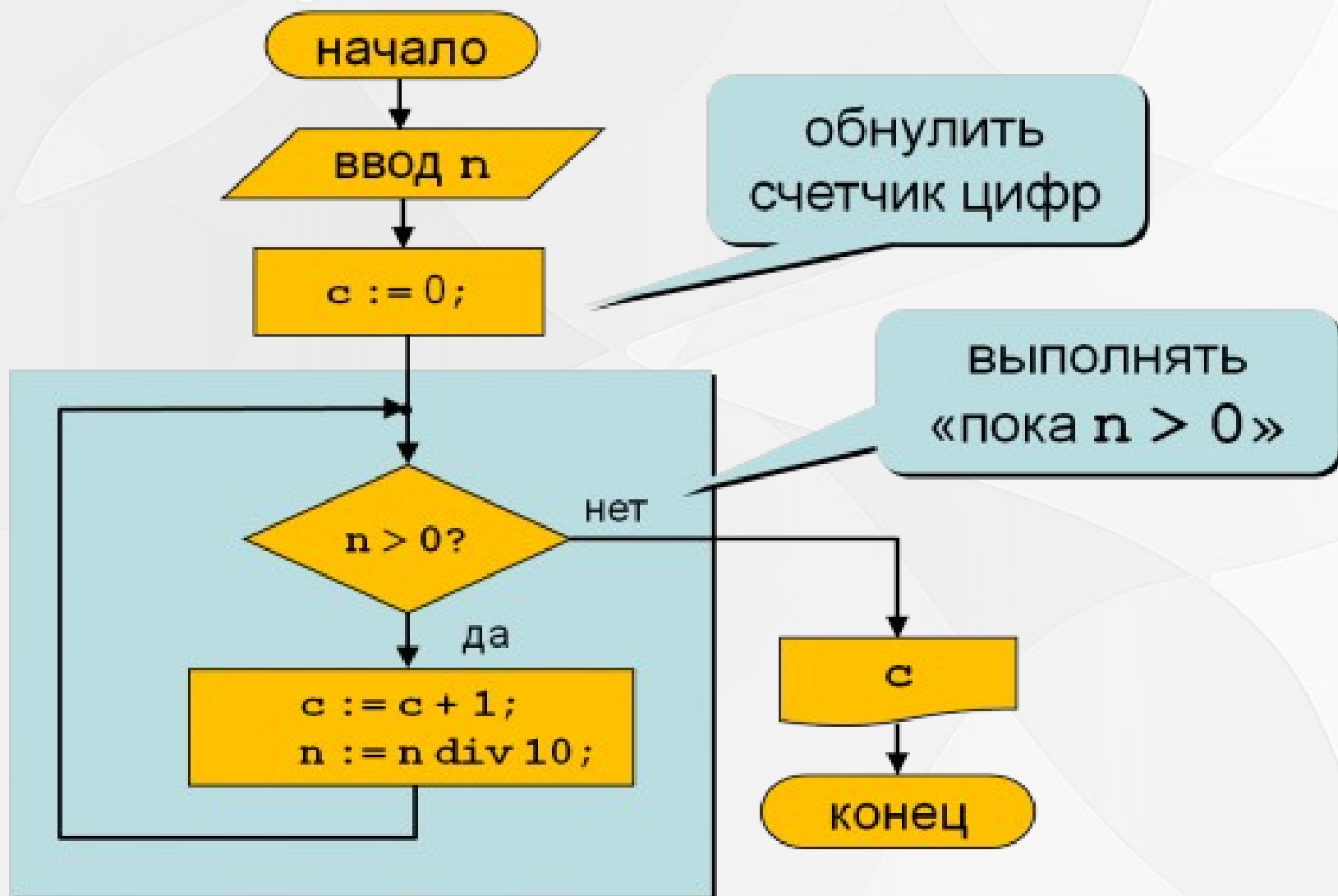
Пример 2: Ввести с клавиатуры число N ($N < 2000000$). Найти количество цифр в этом числе и вывести результат на экран.

Алгоритм: Отделяем и удаляем последовательно последнюю цифру, наращивая счетчик.

n	счетчик
246	0
24	1
2	2
0	3

- **Рассмотрим пример для числа 246.**
- **В чем сложность?** Заранее не определено и неизвестно, сколько цифр нужно убрать, т.е. сколько шагов необходимо сделать.
- **Как выполнить?** Надо перестать отделять цифры, когда $n = 0$, т.е. надо выполнять пока $n > 0$

Блок-схема решения:



Решение примера на Паскале:

```
program qq;  
var n, c, nn: integer;  
begin  
  writeln('Введите целое число');  
  read(n); nn := n;  
  c := 0;  
  while n > 0 do begin  
    c := c + 1;  
    n := n div 10;  
  end;  
  writeln('В числе ', nn, ' нашли ',  
        c, ' цифр');  
end.
```

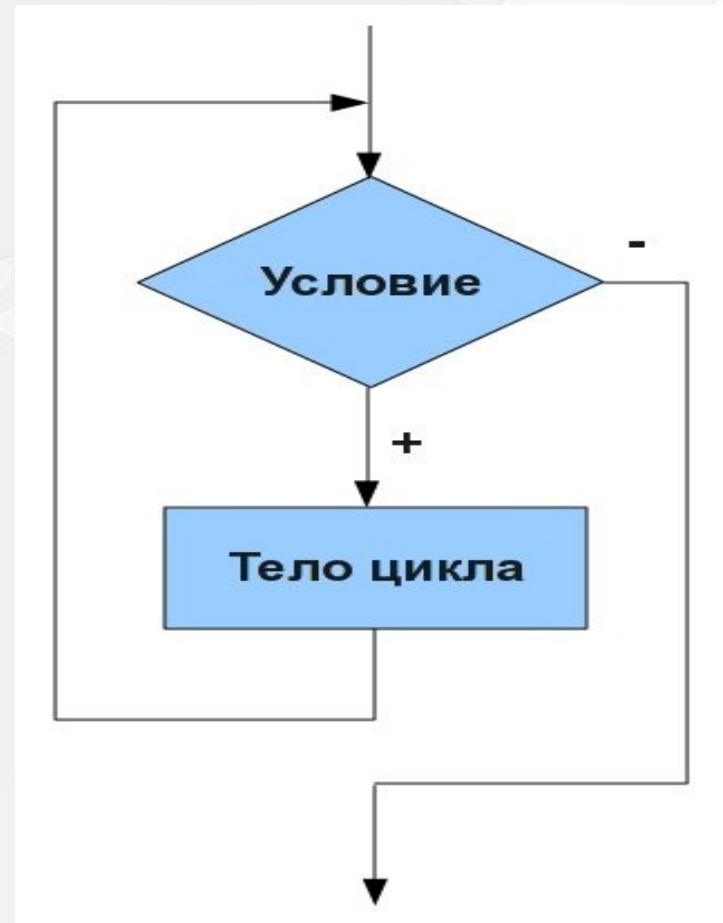
выполнять
пока $n > 0$

nn для удобства
вывода

Выводы

Цикл **while** в Паскале применяется для создания циклов с **неизвестным заранее** числом повторений.

Повторения будут выполняться, **пока истинно некоторое условие.**



Правила записи на Паскале

`while условие do begin` {Пока условие истинно
выполняется оператор}

`Оператор1;
Оператор2;
End;`

- Операторы, стоящий после служебного слова **do**, образуют **тело цикла** и будут выполняться, пока значение "условия" **истинно**.
- **Условие проверяется** каждый раз **при входе** в цикл.
- Если в цикле несколько команд (операторов), то нужно использовать в цикле операторные скобки (**begin... end**).

Задания

- 1) Проверить решение задачи “Выделение цифр числа”.
- 2) Ввести целое число и найти сумму его цифр.

Например:

Введите целое число:

1234

Сумма цифр числа 1234 равна 10.

Программа “Сумма цифр числа”

```
program primer3;
var
  r,s,k,n:integer;
begin
  s:=0;
  Writeln('введите целое число:');
  read(n);      {вводим число}
  while n<>0 do begin
    r:=n mod 10; {r - остаток от деления числа на 10 (т.е. последняя цифра числа)}
    n:=n div 10; {теперь наше число без последней цифры, ,т.к. берем целую часть от}
    s:=s+r;     {считаем сумму цифр}
  end;
  write(s);
end.
|
```

Задачи

Последовательность целых чисел вводится с клавиатуры до тех пор, пока не будет введено число 0. Числа могут быть положительными и отрицательными.

Найти:

- 1) Сумму всех чисел этой последовательности.
- 2) Среднее арифметическое чисел этой последовательности.
- 3) Сумму всех положительных чисел этой последовательности.
- 4) Сумму всех отрицательных чисел этой последовательности.

Программа

```
Var x, summa, spol, sotr, k:integer;  
sr:real;  
begin  
summa:=0; spol:=0; sotr:=0; k:=0;  
readln(x);  
while x<>0 do begin  
summa:=summa+x;      // сумма всех чисел  
k:=k+1;              // количество всех чисел  
if x>0 then spol:=spol+x; // сумма положительных  
if x<0 then sotr:=sotr+x; // сумма отрицательных  
readln(x);           // читаем следующее число  
end;  
sr:=summa/k;  
writeln('сумма всех чисел=', summa);  
writeln('среднее арифметическое=', sr);  
writeln('сумма положительных чисел=', spol);  
writeln('сумма отрицательных чисел=', sotr);  
end.
```

Пример работы программы

Ввод чисел с клавиатуры:

3
-5
6
7
8
-2
0

Вывод результата:

сумма всех чисел=17

среднее арифметическое= 2.8333333333333333335E+000

сумма положительных чисел=24

сумма отрицательных чисел=-7