The background features abstract, overlapping geometric shapes in various shades of green, ranging from light lime to dark forest green. The shapes are primarily triangles and polygons, creating a dynamic, layered effect. The central text is positioned in the white space between these green elements.

# Циклические вычислительные процессы

## Циклический вычислительный процесс.

Циклическим вычислительным процессом называется многократно повторяемая последовательность действий. Все циклические процессы подразделяются на два основных типа:

- циклы с известным числом повторений;
- итерационные циклы.

Существует формула для подсчёта количества шагов в цикле:

$$X = \left[ \frac{X_{\text{кон}} - X_{\text{нач}}}{\Delta X} \right] + 1$$

где

- N - число повторений;
- X<sub>кон</sub>, X<sub>нач</sub> - начальное, конечное значение переменной цикла;
- ΔX - шаг изменения переменной цикла;
- [ ] - указание на необходимость округления дроби до целого значения.

Управляющей переменной цикла (параметром) называется переменная, значениями которой определяется естественное окончание цикла. Группа операторов, повторяющаяся в цикле, называется телом цикла или областью действия цикла. Если по условиям задачи не хватает данных для использования вышеприведённой формулы, то такой цикл относится к итерационным, т.е. с неизвестным числом повторений. Если в теле одного цикла содержится другой, то такая конструкция называется вложенным циклом. Цикл, находящийся в теле другого, называется внутренним циклом, а охватывающий его - внешним. Число вложений циклов неограниченно. Основное требование к группе вложенных циклов - отсутствие пересечений областей действия циклов.

## Циклы с известным числом повторений

Формат оператора:

```
FOR VAR:=EXPR1 TO {DOWNTO} EXPR2 DO ( S1; S2; ... ;  
Sn);
```

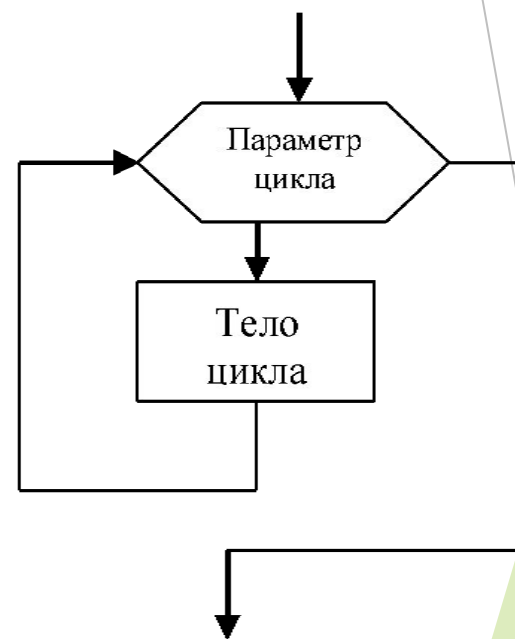
где

- FOR - первое ключевое слово, в переводе означает "от";
- VAR - управляющая переменная (параметр) цикла;
- EXPR1 - начальное значение управляющей переменной, может быть записано в виде константы, переменной или выражения порядкового (целого) типа;
- TO - второе ключевое слово, в переводе означает "до";
- DOWNTO - второе ключевое слово, в переводе означает "до";
- EXPR2 - конечное значение управляющей переменной, может быть записано в виде константы, переменной или выражения порядкового (целого) типа;
- DO - третье ключевое слово, в переводе означает "выполнить";
- ( S1; S2; ... ; Sn ) - составной оператор, тело цикла.

Когда в операторе фигурирует в качестве второго ключевого слова "TO", переменная цикла изменяется с шагом +1, а если "DOWNTO", то шаг становится равным -1.

Оператор работает следующим образом: прочитав заголовок цикла (строку FOR...DO), машина присваивает переменной цикла начальное значение EXPR1 и запоминает конечное - EXPR2. Потом сразу производит сравнение текущего значения параметра цикла с конечным.

Если условие выхода из цикла ещё не достигнуто, начинают исполняться операторы тела цикла S1, S2 и т.д. После обработки этих строк переменная цикла изменяется на шаг, и компьютер передаёт управление на начало цикла, где происходит проверка условия окончания цикла. Так будет происходить до тех пор, пока условие окончания цикла не будет достигнуто.



## Операторы итерационных циклов

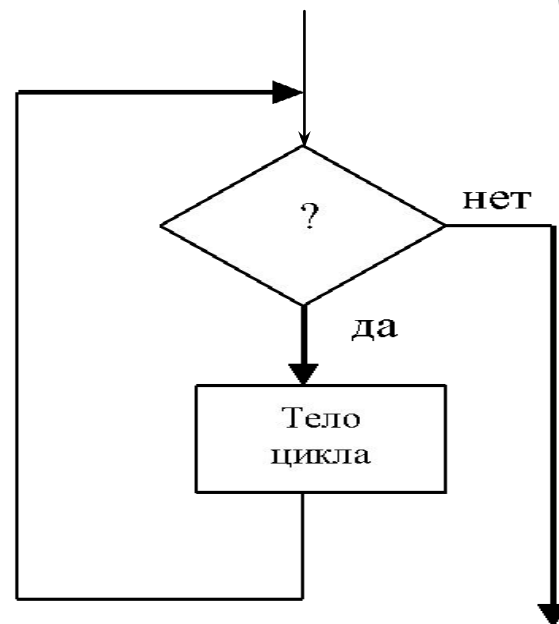
Формат оператора итерационного цикла с предусловием:

```
WHILE <усл> DO ( S1; S2; ... ; Sn );
```

где

- WHILE - первое ключевое слово, в переводе означает "пока";
- <усл> - условие естественного окончания цикла, записанное в виде логического выражения или операции отношения;
- DO - второе ключевое слово, в переводе означает "выполняем";
- (S1;S2;...;Sn) - составной оператор, тело цикла.

Оператор работает следующим образом: до начала исполнения операторов тела цикла происходит проверка условия выполнения цикла (отсюда взято название цикла - "с предварительной проверкой условия"). Если оно не нарушено, тело цикла исполняется. Но может быть, что при неправильном составлении программы, <усл> не выполняется сразу, следовательно, тело цикла тоже ни разу не будет выполнено. Работа продолжается, пока выполняется <усл>.



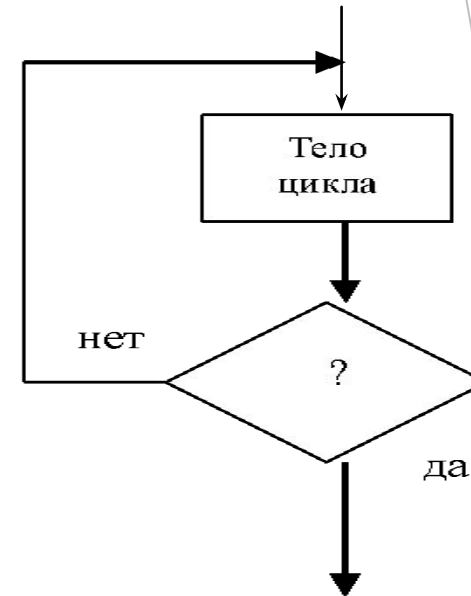
Формат оператора итерационного цикла с постусловием:

REPEAT S1; S2; ... ; Sn UNTIL <усл>;

где

- REPEAT - первое ключевое слово, в переводе означает "повторять";
- S1; S2; ... ; Sn - составной оператор, тело цикла. В этом составном операторе слова BEGIN ... END не нужны, поскольку считается, что слова REPEAT ... UNTIL сами являются операторными скобками;
- UNTIL - второе ключевое слово, в переводе означает "до тех пор, пока";
- <усл> - условие естественного окончания цикла, записанное в виде логического выражения или операции отношения.

Оператор работает следующим образом: после первого выполнения операторов тела цикла происходит проверка условия окончания цикла (отсюда взято название цикла - "с последующей проверкой условия"). Если оно не выполняется, управление передаётся на начало цикла и работа продолжается до выполнения <усл>.



```
var k, s: integer;  
begin  
  s:=2;  
  k:=2;  
  while s < 50 do begin  
    s:=s+k;  
    k:=k+2;  
  end;  
  write(k);  
end.
```

$S=2$

$K=2$

шаг 1:

$s:=s+k=2+2=4$

$k:=k+2=2+2=4$

шаг 2:

$s:=s+k=4+4=8$

$k:=k+2=4+2=6$

шаг 3:

$s:=s+k=8+6=14$

$k:=k+2=6+2=8$

шаг 4:

$s:=s+k=14+8=22$

$k:=k+2=8+2=10$

шаг 5:

$s:=s+k=22+10=32$

$k:=k+2=10+2=12$

шаг 6:

$s:=s+k=32+12=44$

$k:=k+2=12+2=14$

шаг 7:

$s:=s+k=44+14=58$

$k:=k+2=14+2=16$

**Ответ:  $k=16$**