Циклические программы

Оператор цикла repeat

REPEAT оператор ; оператор ; . . . ; оператор

UNTIL условие

Повторяй выполнение операторов, стоящих между словами repeat и until, до тех пор, пока не станет истинным условие.

Сначала компьютер по очереди выполняет операторы, стоящие после слова *repeat*, пока не дойдет до слова *until*, после чего проверяет истинность условия, стоящего после *until*.

Если условие ложно, то компьютер снова по очереди выполняет эти операторы и снова проверяет истинность условия и т.д.

Если условие оказывается истинным, то работа оператора *repeat* прекращается и компьютер переходит к выполнению следующего по порядку оператора.

Конструкция repeat . . . . . until a+2>3*b переводится **"повторяй.....до тех пор, пока a+2 не станет больше 3*b".**

Задача: При помощи цикла напечатать на экране:
Начало счета 3 5 7 9 Конец счета

```
VAR f : Integer;
BEGIN
 Write('Начало счета   ');
 f:=3;
 repeat
      Write(f,'  ');
      f:=f+2;
 until  f>9;

 Write(' Конец счета')
END.
```

Задача: Компьютер предлагает человеку ввести слово, после чего распечатывает это слово, снабдив его восклицательным знаком. Затем снова предлагает ввести слово и так до тех пор, пока человек не введет слово "Хватит". Распечатав его с восклицательным знаком, компьютер отвечает "Хватит так хватит" и заканчивает работу.

```
VAR Slovo : String;
BEGIN
 repeat
     WriteLn('Введите слово');
     ReadLn(Slovo);
     WriteLn(Slovo, '!')
 until Slovo='Хватит';
 WriteLn('Хватит так хватит')
END.
```

Задание 37: Усложним эту задачу. Пусть компьютер перед распечаткой каждого слова ставит его порядковый номер.

```
VAR     Slovo :String;
        Nomer      :Integer;
BEGIN
 Nomer:=1;
 repeat
        WriteLn('Введите слово');
        ReadLn(Slovo);
        WriteLn(Nomer, ' ', Slovo, '!');
        Nomer:=Nomer+1;
 until Slovo='Хватит';
 WriteLn('Хватит так хватит');
 ReadLn
END.
```

Оператор цикла while

Синтаксис оператора while:

WHILE условие **DO** оператор

- ***Пока*** *условие истинно,* ***делай*** *оператор*.
- Например, $while\ a>b\ do\ b:=b+1$.
- Работает оператор while так:

Сначала компьютер проверяет истинность условия, стоящего после слова *while*. Если условие истинно, то выполняется оператор, стоящий после *do*. Затем снова проверяется истинность условия и в случае истинности снова выполняется этот оператор. И т.д. Если условие ложно, то оператор *while* прекращает свою работу и компьютер переходит к выполнению следующего оператора.

- Оператор, стоящий после while, вполне может быть составным, поэтому тело цикла у оператора while, так же как и у оператора repeat, может состоять из многих операторов.

Решим при помощи while ту же задачу о печати чисел *3 5 7 9*

- ```
  VAR f : Integer;
  BEGIN
   Write('Начало счета   ');
   f:=3;
   while f<=9 do
        begin
             Write(f,'  ');
             f:=f+2;
        end;
   Write(' Конец счета')
  END.
  ```
- Здесь после *do* стоит составной оператор *begin Write(f,' '); f:=f+2; end*.
  ```

Отличия операторов repeat и while

1. Компьютер выходит из цикла оператора *repeat* тогда, когда условие истинно, а из цикла оператора *while* - когда условие ложно.
2. *while* может ни разу не выполнить оператор, стоящий после *do*. *repeat* же хотя бы раз операторы, стоящие между *repeat* и *until*, выполнит.

Например, фрагмент

k:=8; repeat k:=1 until 3>2; WriteLn(k) напечатает 1.

А фрагмент

k:=8;while 2>3 do k:=1; WriteLn(k) напечатает 8.

Оператор цикла for

- Выполняя программу печати чисел *3 5 7 9*, оператор repeat выполнил цикл 4 раза. То же самое сделал и оператор while. Однако, обычно, когда мы пишем операторы repeat и while, нам совсем неважно знать, сколько раз они выполнят цикл. Тем не менее, существует много задач, для решения которых цикл нужно выполнить именно определенное количество раз. В этом случае удобно использовать оператор цикла for.

Задача: 200 раз напечатать слово ФУТБОЛ.

- **VAR** i : Integer;
  BEGIN
  for i:=1 **to** 200 **do**
  WriteLn('ФУТБОЛ')
  END.
- Конструкция **for i:=1 to 200 do** по-русски читается так:
- ***Для i, изменяющегося от 1 до 200, делай оператор, стоящий после слова do.***
- Смысл повторения здесь такой же, как и в операторе while. *Оператор, стоящий после do, тоже, конечно, может быть составным.*

Синтаксис оператора for:

FOR имя **:=** *выражение1* **TO** *выражение2* **DO**

оператор

- ***имя*** - это имя произвольной переменной порядкового типа (см. 5.7 и 12.8), в частности целочисленной, называемой ***переменной цикла***,
- ***выражение1*** **и** ***выражение2*** - произвольные выражения порядкового типа, в частности - целого.

Работа оператора for:

Прежде всего вычисляется *выражение1*, и переменной цикла (пусть это будет *i*) присваивается его значение. Затем вычисляется *выражение2* и сравнивается с *i*. Если *i* > *выражения2*, то оператор *for* завершает свою работу, так ничего и не сделав. В противном случае выполняется оператор, стоящий после *do*. После выполнения этого оператора значение *i* увеличивается на *1* и снова сравнивается с *выражением2*. Если *i* > *выражения2*, то оператор *for* завершает свою работу, иначе снова выполняется оператор, стоящий после *do*, снова *i* увеличивается на 1 и т.д.

В нашем примере переменная *i* кроме того, что обеспечивает нам выход из цикла, никакой "полезной" работы не выполняет. Усложним же немного задачу. Пусть компьютер печатает такую информацию:

10 ФУТБОЛ 11 ФУТБОЛ 12 ФУТБОЛ . . . . 200 ФУТБОЛ

- ```
  VAR i : Integer;
  BEGIN
   for i:=10 to 200 do
       begin       Write(i);
                   Write (' ФУТБОЛ ')
       end
  END.
  ```
- *Здесь после do стоит уже составной оператор.*
  ```

Удобно ли использовать оператор for для печати такой информации?:

100 ФУТБОЛ 99 ФУТБОЛ 98 ФУТБОЛ . . . 40 ФУТБОЛ

Оператор for позволяет не только увеличивать, но и ***уменьшать переменную цикла.*** Однако, для этого нельзя писать $for\ i:=100\ to\ 40$, а нужно писать ***for i:=100 downto 40,*** переводится буквально "$вниз\ до$". Соответственно, для выхода из цикла должно быть истинным не условие $i > выражения2$, а условие $i < выражения2$.

Вот объединенный синтаксис оператора for:

FOR имя **:=** выражение1 **TO** | **DOWNTO** выражение2 **DO** оператор

Вертикальная черта | между двумя элементами конструкции «TO и DOWNTO» говорит о том, что в конструкции должен присутствовать один из этих элементов.

Вычислительная циклическая программа

Задача: Во дворце 40 залов. Известны длина, ширина и высота каждого зала. Вычислить площадь пола и объем каждого зала.

Сначала напишем фрагмент для одного зала:

```
ReadLn (dlina, shirina, visota);
S:=dlina*shirina;                {Площадь пола}
V:=S*visota;                     {Объем}
WriteLn(S,’  ‘,V)
```

Для решения задачи этот фрагмент нужно выполнить 40 раз, для чего вполне естественно использовать оператор for:

```
VAR i, dlina, shirina, visota, S, V: Integer;
BEGIN
    for i:=1 to 40 do begin
        ReadLn (dlina, shirina, visota);
        S:=dlina*shirina;
        V:=S*visota;
        WriteLn(S,’  ‘,V)
    end {for}
END.
```

Теперь дадим возможность пользователю самому задавать число залов во дворце:

```
VAR   i, dlina, shirina, visota, N, S, V : Integer;
BEGIN
  WriteLn('Введите число залов');
  ReadLn (N);                    {N - число залов}
  for i:=1 to N do begin
        WriteLn('Введите длину,ширину и высоту зала');
        ReadLn (dlina, shirina, visota);
        S:=dlina*shirina;
        V:=S*visota;
        WriteLn('Площадь пола=',S,' Объем зала=',V)
  end
END.
```

Пусть во дворце три зала размерами 20*15*4, 30*20*5 и 10*5*3. В этом случае мы вводим N=3 и оператор for выполняет цикл три раза. На каждом выполнении цикла компьютер останавливается на операторе ReadLn (dlina, shirina, visota), мы вводим числа и получаем результаты:

- **Площадь пола=300 Объем зала=1200**
- **Площадь пола=600 Объем зала=3000**
- **Площадь пола=50 Объем зала=150**

Задание 42: Напечатать с помощью оператора for:

Прямой счет: -5 -4 -3 -2 -1 0 1 2 3 4 5 Обратный счет: 5 4 3 2 1 0 -1 -2 -3 -4 -5 Конец счета

- ```
  VAR i : Integer;
  BEGIN
      Write('Прямой счет: ');
      for i:=   -5 to 5 do  Write(i,' ');
      Write('Обратный счет: ');
      for i:=    5  downto  -5 do Write(i,' ');
      Write('Конец счета');
  END.
  ```
  ```

Задание 43: "Даны стороны N кубиков. Вычислить объем каждого.

```
VAR  i, N, a  : Integer;
BEGIN
 WriteLn('Введите число кубиков');
 ReadLn (N);
 for i:=1 to N do begin
     WriteLn('Введите длину стороны кубика');
     ReadLn (a);
     WriteLn('Объем кубика=', a*a*a)
 end;
 ReadLn
END.
```

Задания 44-45.

Определите без компьютера, что будет, если

44) строку for i:=1 to N do begin поместить под строкой ReadLn (dlina, shirina, visota)

45) поменять местами строки WriteLn('Площадь пола=',S,' Объем зала=',V) и end

- ***Задание 44*** Компьютер напечатает:
- Площадь пола=300 Объем зала=1200
- Площадь пола=300 Объем зала=1200
- Площадь пола=300 Объем зала=1200
- и не спросит размеры 2 и 3 залов.

- ***Задание 45*** Компьютер напечатает результаты только для последнего зала.

Счетчики

Задача 1: В компьютер с клавиатуры вводятся числа. Компьютер после ввода каждого числа должен печатать, сколько среди них уже введено положительных.

Задача 2: В компьютер вводится ровно 200 чисел. Компьютер должен подсчитать и один раз напечатать, сколько среди них положительных.

- **Программа:**
- **VAR** c,i :Integer;
- a :Real;
- **BEGIN**
- c:=0; {Обнуляем счетчик}
- **for** i:=1 **to** 200 **do begin**
- ReadLn(a);
- **if** a>0 **then** c:=c+1
- **end** {for};
- WriteLn('Из них положительных - ' ,c)
- **END.**