

Циклический алгоритм

Цикл ДО и ПОКА

Циклический алгоритм

Циклом называется программа (или часть программы), многократно выполняемая при заданном условии.

Основные виды циклов: *ДО* и *ПОКА*.

Цикл со счетчиком (ДО)

Если нужно выполнить много раз одно и то же действие, то на помощь приходит оператор цикла FOR...NEXT.

Для работы с оператором цикла FOR...NEXT нужно соблюдать некоторые правила:

- ✓ рассмотреть повторяющиеся действия и выделить в них равномерно изменяющуюся величину (параметр);

Правила для цикла FOR...NEXT

- ✓ дать параметру имя;
- ✓ определить для параметра *начальное значение, конечное значение и шаг изменения*, т.е. насколько за один раз увеличивается (или уменьшается) параметр.

Правила для цикла FOR...NEXT

✓ Написать оператор цикла, состоящий из трех частей:

Заголовок цикла

FOR параметр=нач_значение **TO** кон_значение **STEP** шаг

Тело цикла

В теле цикла указываются один или несколько операторов, предназначенных для повторяющихся действий, причем вместо конкретных значений изменяющейся величины указывают имя параметра.

NEXT параметр

Пример с мишенью

Линейный алгоритм

```
SCREEN 12:CLS  
CIRCLE (320, 240), 20, 15  
CIRCLE (320, 240), 40, 15  
CIRCLE (320, 240), 60, 15  
CIRCLE (320, 240), 80, 15  
CIRCLE (320, 240), 100, 15  
END
```

В результате работы этой программы на экран будет выведена мишень, состоящая из пяти колец.

Пример с мишенью

Решение той же задачи, но с использованием цикла.

```
SCREEN 12:CLS
```

```
FOR R = 20 TO 100 STEP 20
```

```
CIRCLE (320, 240), R, 15
```

```
NEXT R
```

```
END
```



Задача

Написать программу, подсчитывающую значения функции из промежутка $[-5 ; 1]$ с шагом $0,1$, где $f(x)=x^3+1$.

Решение.

1. Для решения задачи используем цикл FOR...NEXT. Постоянно изменяющаяся величина – значение аргумента функции.
2. Имя параметра цикла – **X**.

3. Определить начальное значение параметра – (-5); конечное значение параметра – 1; шаг изменения – 0,1.
4. Приступаем к составлению программы:

```
CLS
```

```
FOR X = -5 TO 1 STEP .1
```

```
Y=X^3+1
```

```
PRINT "f(x)= "Y " при X= "X
```

```
NEXT X
```

```
END
```



Еще задача.

Написать программу проверки таблицы умножения, содержащую 5 вопросов и выставляющую оценку.

Решение

```
RANDOMIZE TIMER
```

```
FOR X=1 TO 5
```

```
  B=INT(RND*10+1)
```

```
  C=INT(RND*10+1)
```

```
  PRINT "СКОЛЬКО БУДЕТ" B "*" C
```

```
  INPUT S
```

```
  IF B*C<>S THEN N=N+1
```

```
NEXT
```

```
L=5-N
```

```
PRINT "Ваша оценка:" L "Вы сделали" N "ошибок"
```

```
END
```

Цикл ПОКА

Цикл **WHILE - WEND** позволяет сочетать свойства условного оператора и оператора цикла. С их помощью можно выполнять повторяющиеся действия с *заранее неизвестным количеством повторений*.

WHILE <условие>

.....
.....
.....
.....

ТЕЛО ЦИКЛА

WEND

С начала идет оператор **WHILE** с условием, при котором цикл выполняется.

Ключевое слово **WEND** аналогично по своему назначению слову **NEXT**

Цикл ПОКА

В условии используются логические выражения, например: **A=B**, **A>=B+1**, **A<B**, **A<>B** и т.д.

Значения переменных могут быть подсчитаны до цикла, почти всегда значения переменных изменяются в теле цикла.

Работа такого цикла начинается с проверки условия на истинность или ложность, записанного в **WHILE**. Если условие истинно (ответ: Да), то начинается выполнение тела цикла, после чего оператор **WEND** вновь возвращает нас на начало цикла **WHILE**. Если в теле цикла происходят какие-то изменения, влияющие на результат проверки условия, то может наступить момент, когда это условие становится ложным, и управление из **WHILE** будет передано на оператор, следующий за **WEND**. В противном случае цикл будет бесконечным.

Пример

CLS

НЕТ

WHILE WORD\$ <> "TERMINATOR"

ДА

 INPUT "ВВЕДИТЕ ПАРОЛЬ"; WORD\$

WEND

 ?"ПРАВИЛЬНО! ВЫ ДОПУЩЕНЫ К
ПРОДОЛЖЕНИЮ РАБОТЫ"

END



Задача

Написать программу игры, в которой пользователю предлагается угадать задуманное ЭВМ число из промежутка $[1;100]$. За каждый неправильный ответ начисляется по три штрафных очка.

Решение

```
CLS
RANDOMIZE TIMER
X=INT(RND*100+1)
K=0
INPUT "Отгадайте задуманное число от 1 до 100"; N
WHILE X<>N
    IF N>X THEN
        PRINT "Ваше число больше задуманного"
    ELSE
        PRINT "Ваше число меньше задуманного"
    END IF
    K=K+3
    INPUT "Введите новое число"; N
WEND
PRINT "Вы угадали число"
PRINT "Сумма штрафных очков равна" K
END
```


Домашняя работа

Вычислить сумму чётных чисел, не равных нулю, из промежутка $[a;b]$ и назвать их количество.