



Циклы

Последнее обновление: 30.10.2015



Еще одним видом управляющих конструкций являются циклы. В VB.NET используется несколько видов циклов.

- For...Next
- For Each...Next
- While
- Do

Цикл For...Next

В этом цикл выполняется определенное число раз, причем это число задается счетчиком:

```
1 For i As Integer = 1 To 9
2     Console.WriteLine("Квадрат числа {0} равен {1}", i, i * i)
3 Next
```

Здесь переменная *i* выполняет роль счетчика. После слова **To** мы помещаем максимальное значение счетчика. При каждом цикле значение счетчика увеличивается на единицу. И это значение сравнивается со значением после **To**. Если эти два значения равны, то цикла прекращает свою работу.

При работе с циклами мы можем увеличивать значение счетчика при каждом проходе не только на единицу, но и вообще на любое число. Для этого нужно либо использовать ключевое слово **Step** и после него указать шаг цикла, на который будет увеличиваться значение счетчика, либо можно увеличивать счетчик непосредственно в цикле:

```
1 For i As Integer = 1 To -9 Step -1
2     For j As Integer = 1 To 9
3         Console.WriteLine("Произведение чисел i и j равно {0}", i * j)
4         j += 1
5     Next
```

Обратите внимание, что в качестве шага в первом цикле выбрано отрицательное значение и значение счетчика с каждым проходом уменьшается на единицу. Во внутреннем цикле счетчик *j* при каждом проходе увеличивается на 2, так как он по умолчанию увеличивается на единицу, и еще мы явным образом увеличиваем его в цикле на единицу. В итоге внутренний цикл обрабатывает пять раз, а внешний девять, то есть фактически получается 45 циклов.

Цикл For Each...Next

Цикл For Each осуществляет перебор элементов в определенной группе, например, в массиве или в коллекции. Предположим у нас есть некоторый массив типа Integer и нам надо инициализировать этот массив случайными значениями и затем вывести все его элементы на экран:

```
1  'Создаем массив из пяти чисел
2  Dim nums(4) As Integer
3  Dim r As New Random()
4  'инициализируем массив
5  For i As Integer = 0 To nums.Length - 1
6      nums(i) = r.Next(100)
7  Next
8  'Выводим элементы массива
9  For Each i As Integer In nums
10     Console.Write("{0} ", i)
11 Next
```

В выражении For Each мы сначала указываем переменную, которая будет принимать значения элементов массива. А после ключевого слова **In** указываем группу, в которой надо перебрать все элементы.

Цикл While

В цикл While выполняется до тех пор, пока соблюдается определенное условие, указанное после слова While:

```
1  Dim j As Integer = 10
2  While j > 0
3      Console.WriteLine(j)
4      j -= 1
5  End While
```

Цикл Do

Цикл Do, также как и цикл While, выполняется, пока соблюдается определенное условие. Однако он имеет разные формы. Так, в следующем примере сначала проверяется

условие, а затем выполняется блок кода, определенный в цикле:

```
1 Dim j As Integer = 10
2 Do While j > 0
3     Console.WriteLine(j)
4     j -= 1
5 Loop
```

В данном случае цикл выполняется, пока значение *j* больше нуля. Но есть еще одна запись, где вместо слова **While** используется слово **Until**, а цикл выполняется пока не соблюдено определенное условие, то есть пока значение *j* не станет меньше нуля:

```
1 Dim j As Integer = 10
2 Do Until j < 0
3     Console.WriteLine(j)
4     j -= 1
5 Loop
```

Если изначально условие, заданное в цикле, неверно, то цикл не будет работать. Но мы можем определить проверку в конце цикла, и таким образом, наш цикл как минимум один раз отработает:

```
1 Dim j As Integer = -1
2 Do
3     Console.WriteLine(j)
4     j -= 1
5 Loop Until j < 0
6 'либо
7 Do
8     Console.WriteLine(j)
9     j -= 1
10 Loop While j > 0
```

Операторы Continue и Exit

Нередко возникает необходимость не дожидаться окончания цикла, а сразу же выйти из цикла, в случае соблюдения определенного условия. Для этого используют оператор **Exit**, после которого указывают тип цикла, из которого осуществляется выход, например, **Exit Do** (**Exit While**):

```
1 Dim r As New Random()
2 Dim num As Integer = r.Next(100)
3
4 For i As Integer = 0 To 100
5     num -= 1
6     If num < 50 Then Exit For
7 Next
8 Console.WriteLine(num)
```

Существует и другая задача - осуществить выход не из цикла, а из текущего прохода или итерации и перейти к следующему. Для этого используют оператор **Continue**, после которого указывают тип цикла, из которого осуществляется выход, например, `Continue While`:

```
1 Dim r As New Random()
2 Dim num As Integer = r.Next(100)
3
4 For i As Integer = 0 To 10
5     num -= 7
6     If num < 50 AndAlso num > 25 Then
7         Continue For
8     End If
9
10    Console.WriteLine(num)
11 Next
```

В данном случае мы в каждом проходе цикла вычитаем из `num` число 7 и затем смотрим, не принадлежит ли число `num` интервалу от 25 до 50. И если принадлежит, переходим к новой итерации цикла, а если нет, то выводим его на экран.

[Назад](#) [Содержание](#) [Вперед](#)



ТАКЖЕ НА METANIT.COM

Клиент на Xamarin Forms для SignalR

10 дней назад • 1 коммента...

Клиентское приложение на Xamarin Forms для SignalR в ASP.NET Core, ...

Взаимодействие с кодом Python

5 месяцев назад • 4 коммен...

Взаимодействие с кодом Python в программе на языке Си, установка Qt, ...

Параметры строки запроса

5 месяцев назад • 1 коммен...

Параметры строки запроса query string в приложении Blazor на ...

ListVi

2 меся

ListVi
созда
получ

G

Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS

?

Имя

♡ 2

Поделиться

Лучшие

Новые

Старые

S

sherzod

3 года назад

Из предложенных вариантов выберите оператор цикла, который выполняет цикл один раз, проверяет условие в конце цикла и, если оно ложно, продолжает выполнять цикл, пока условие не станет истинным.

Выберите один ответ:

Do While...Loop

Do Until...Loop

Do...Loop Until

Do...Loop While

For...Next

0 0 Ответить • Поделиться >

Помощь сайту

YooMoney:

410011174743222

Qiwi:

qiwi.com/n/METANIT

Перевод на карту

Номер карты:

4048415020898850

[Вконтакте](#) | [Телеграм](#) | [Twitter](#) | [Помощь сайту](#)

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2023. Все права защищены.