METANIT.COM
Сайт о программировании

Статические члены классов

Последнее обновление: 30.10.2015

Обычно, чтобы использовать какой-нибудь класс, мы должны создать его объект. Однако если данный класс имеет статические или разделяемые методы, то чтобы получить к ним доступ, мы не обязательно должны использовать объекты. Например, создадим новый класс **Algorithm** и добавим в него две функции для вычисления числа Фибоначчи и факториала:

```
Public Class Algorithm

    Public Shared pi As Double = 3.14

    Public Shared Function Factorial(x As Integer) As Integer
        If (x = 1) Then
            Return 1
        Else
            Return x * Factorial(x - 1)
        End If
    End Function

    Public Shared Function Fibbonachi(x As Integer) As Integer
        If x = 0 Then
            Return 1
        ElseIf x = 1 Then
            Return 1
        Else
            Return Fibbonachi(x - 1) + Fibbonachi(x - 2)
        End If
    End Function

End Class
```

Обратите внимание на модификатор **Shared**, он указывает, что данные члены будут доступны для всего класса, то есть будут общими. В этом случае мы их можем использовать следующим образом:

```
Sub Main()
```

```
    Dim num1 As Integer = Algorithm.Factorial(5)

    Dim num2 As Integer = Algorithm.Fibbonachi(5)

    Algorithm.pi = 3.14159

    Console.WriteLine("Более точное значение числа PI = {0}", Algorithm.pi)
End Sub
```

В данном случае нам не надо создавать экземпляр или объект класса, мы можем обратиться к общему члену напрямую, указав только имя класса.

Статический конструктор

Мы можем объявлять в классах также и статические конструкторы:

```
Public Class State

    Shared Sub New()
        Console.WriteLine("Создано первое государство")
    End Sub

End Class
```

Статические конструкторы выполняются при самом первом создании объекта данного класса:

```
Sub Main()

    'В этом месте выполняется обычный и статический конструктор,
    ' Так как это первое создание объекта State в программе
    Dim s1 As State = New State()
    'В этом месте выполняется обычный конструктор
    Dim s2 As State = New State()

End Sub
```

Ключевое слово Static

Чтобы понять смысл модификатора **Static**, сначала посмотрим на примере, в котором целесообразно его использование. Итак, у нас определен цикл:

```
For i As Integer = 0 To 4
    Dim temp As Integer = 4
    Console.WriteLine(temp)
    temp += 1
Next
```

В этом цикле у нас определена переменная `temp`, которая после вывода на экран ее значения увеличивается на единицу. Однако при каждом новом проходе цикла эта переменная создается заново, и ее значение при выводе на экран всегда будет равно 4. Однако если мы заменим Dim на Static, то значение переменной при каждом новом проходе будет сохраняться, и при выводе на экран мы увидим, что ее значение увеличивается:

```
For i As Integer = 0 To 4
    Static temp As Integer = 4
    Console.WriteLine(temp)
    temp += 1
Next
```

Посмотрим на другом примере. Создадим метод со статической переменной, которая будет увеличиваться на единицу и отображаться на экране. И затем дважды вызовем этот метод:

```
Private Sub Display()
    Static i As Integer = 6
    i += 1
    Console.WriteLine(i)
End Sub

Sub Main()

    Display()
    Display()

End Sub
```

Мы увидим, что несмотря на то, что вызов метода был завершен, значение переменной сохранилось, а при следующем вызове оно равно не 6, а 7. Тогда как если бы мы заменили слово Static на Dim, переменная после работы метода уничтожалась, а при следующем вызове метода создавалась бы вновь. В тоже время вместо переменной Static, мы можем использовать глобальную переменную на уровне класса или модуля.

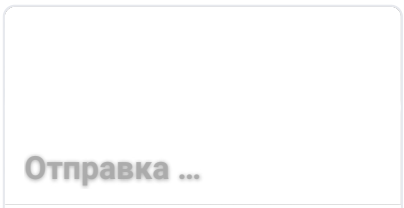
И Static, и Shared могут использоваться в качестве модификаторов переменных. Однако отличие между ними состоит в том, что с помощью Static определяются переменные внутри методов, а с помощью Shared - переменные на уровне классов и модулей.

Назад **Содержание** **Вперед**

ТАКЖЕ НА METANIT.COM

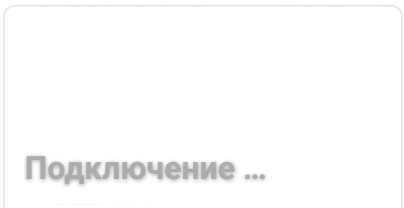
5 месяцев назад · 4 коммен...

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



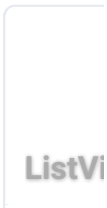
5 месяцев назад · 1 коммен...

Отправка запросов на сервер HttpServer с помощью класса ...



5 месяцев назад · 1 коммен...

Библиотека sqlite3, подключение к базе данных SQLite в ...



2 меся

ListVie
созда
получ

2 Комментариев

Войти ▼

Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ ИЛИ ЧЕРЕЗ DISQUS ?

Имя

♡ Поделиться Лучшие Новые Старые

Alexandr

7 лет назад edited

Подскажите, плз, ведь Module в VB.NET есть аналог static class в C#? И если нужен полностью static class, а не какие-то Shared члены обычного, то достаточно использовать модуль, без всяких там Shared?

1 0 Ответить • Поделиться ›

Metanit Модератор → Alexandr

7 лет назад

вообще да, наиболее близкой конструкцией к статическим классам в C# в VB NET считается модуль

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.