Свойства

Последнее обновление: 30.10.2015

В VB.NET кроме обычных методов и конструкторов также есть специальные методы доступа - **свойства**. Свойства помогают получить доступ к полю класса, узнать его значение или наоборот установить его. Для объявления свойства используется ключевое слово **Property**:

```
Public Class State

    Dim name_ As String

    Public Property Name As String
        Get
            Return name_
        End Get
        Set(value As String)
            name_ = value
        End Set
    End Property

End Class
```

У нас определено поле **name_**, оно закрытое, и чтобы получить к нему доступ из других частей программы и классов, мы будем использовать свойство **Name**. При объявлении мы также задаем тип поля, для которого это свойство создано (в данном случае String). Обратите внимание на синтаксис свойства: оно имеет два блока - **Get** и **Set**. В блоке **Get**:

```
Get
    Return name_
End Get
```

мы получаем с помощью оператора **Return** значение переменной name_, оно же и будет значением данного свойства. В блоке **Set** мы осуществляем присваивание значения переменной name_:

```
Set(value As String)
    name_ = value
```

```
End Set
```

Параметр value представляет передаваемое значение. Мы можем использовать данное свойство следующим образом:

```
Sub Main()

    Dim s As State = New State()

    'Устанавливаем свойство - отрабатывает блок Set
    s.Name = "Russia"

    'Получаем значение свойства и присваиваем его переменной - отрабатывает
    Dim stateName = s.Name

    Console.ReadLine()
End Sub
```

Несмотря на то, что для свойства определен модификатор Public, мы можем задать модификатор и для блока Set или Get:

```
Public Property Name As String
    Private Get
        Return name_
    End Get
    Set(value As String)
        name_ = value
    End Set
End Property
```

В этом случае мы уже не сможем в модуле получить значение свойства:

```
Sub Main()

    Dim s As State = New State()

    'Устанавливаем свойство - отрабатывает блок Set
    s.Name = "Russia"

    'Получаем значение свойства - блок Get объявлен как Private,
    'поэтому здесь будет ошибка
    Dim stateName = s.Name

    Console.ReadLine()
End Sub
```

Read-Only и Write-Only в определениях свойств

Иногда необходимо сделать свойства доступными только для чтения или записи. В этом случае мы можем использовать ключевые слова **ReadOnly** и **WriteOnly**. Если мы хотим сделать свойство только для чтения, то надо использовать слово **ReadOnly** и опустить Set. Если надо создать свойство только для записи, то используется слово **WriteOnly** и опускается Get:

```
Public Class State

    Dim name_ As String

    Dim population_ As Integer

    ReadOnly Property Name As String
        Get
            Return name_
        End Get
    End Property

    WriteOnly Property Population As Integer
        Set(value As Integer)
                population_ = value
        End Set
    End Property

End Class
```

Автоматические свойства

В .NET 4 была добавлена новая функциональность - **автоматические свойства**. Благодаря таким свойствам можно использовать сокращенный синтаксис объявления свойств и не создавать дополнительно поля в классе, с которыми связаны эти свойства:

```
Public Class State

    Public Property President() As String
    Public Property Area() As Double

End Class
```

Правда, использование автосвойств имеет ограниченные возможности, во-первых, поскольку мы не можем при получении или присвоении значения добавить в свойства дополнительную логику; во-вторых, мы не можем таким образом создавать свойства только для записи или только для чтения.

Назад Содержание Вперед

qiwi.com/n/METANIT

Перевод на карту
Номер карты:
4048415020898850

Вконтакте | Телеграм | Twitter | Помощь сайту

Контакты для связи: metanit22@mail.ru