



Классы. ООП

Классы и объекты

Последнее обновление: 30.10.2015



Visual Basic.NET является полноценным объектно-ориентированным языком, а это значит, что программа может быть представлена в виде взаимосвязанных объектов, которые взаимодействуют между собой. Описанием объекта является **класс**, в то время как объект - экземпляр этого класса. Класс определяется с помощью ключевого слова **Class**:

```
1 Class Book
2 End Class
```

Всю функциональность класса обеспечивают его члены - поля, свойства, методы, конструкторы, события. Поля представляют обычные переменные? обычным образом также определяются процедуры и функции (в этом плане классы во многом похожи на структуры):

```
1 Class Book
2     'Название книги
3     Dim name As String
4     'Автор
5     Dim author As String
6     'Год издания
7     Dim year As Integer
8
9     'Метод для вывода информации о книге
10    Sub GetInformation()
11        Console.WriteLine("Книга '{0}' (автор {1}) была издана в {2} году",
12        End Sub
13 End Class
```

Кроме обычных методов в классах существуют специальные методы - **конструкторы**. Конструкторы вызываются при создании нового объекта класса. Чтобы объявить конструктор, надо использовать ключевое слово **New**. Зачем нужен конструктор?

Обычно конструктор выполняет инициализацию членов класса. Объявим в классе конструктор, который будет инициализировать поля нашего класса Book:

```
1 Public Class Book
2     'Название книги
3     Dim name As String
4     'Автор
5     Dim author As String
6     'Год издания
7     Dim year As Integer
8
9     Sub New(name As String, author As String, year As Integer)
10        Me.name = name
11        Me.author = author
12        Me.year = year
13    End Sub
14
15 End Class
```

Здесь мы создали конструктор, который принимает три параметра - название книги, ее автора и год издания. Затем в конструкторе мы присваиваем значения параметров полям класса. Поскольку параметры и поля класса называются одинаково, то нам надо использовать ключевое слово **Me**, которое предоставляет ссылку на данный класс. Если бы параметры имели другие имена, то использование слова Me было бы необязательным. Если мы не создадим свой конструктор, тогда будет использоваться конструктор по умолчанию:

```
1 Sub New()
2
3 End Sub
```

Теперь используем класс. Создайте новое консольное приложение. Затем нажмите правой кнопкой мыши на название проекта в окне **Solution Explorer (Обозреватель решений)** и в появившемся меню выберите пункт **Add (Добавить)**, в другом появившемся меню выберите пункт **Class (Класс)**. В окне создания нового класса присвойте ему имя **Book** и нажмите кнопку **Add (Добавить)**. В проект будет добавлен новый класс **Book**, который будет находиться в файле **Book.vb**. Перенесите в этот класс следующий код:

```
1 Public Class Book
2     'Название книги
3     Dim name As String
4     'Автор
5     Dim author As String
6     'Год издания
7     Dim year As Integer
8
```

```

9      Sub New(name As String, author As String, year As Integer)
10          Me.name = name
11          Me.author = author
12          Me.year = year
13      End Sub
14      'Конструктор по умолчанию
15      Sub New()
16          name = "Евгений Онегин"
17          author = "А. С. Пушкин"
18          year = 1833
19      End Sub
20
21      'Метод для вывода информации о книге
22      Sub GetInformation()
23          Console.WriteLine("Книга '{0}' (автор {1}) была издана в {2} году",
24          End Sub
25  End Class

```

Теперь в основной файл приложения - в модуль добавим следующий код:

```

1  Module Module1
2
3      Sub Main()
4
5          Dim b1 As Book = New Book("Война и мир", "Л. Н. Толстой", 1869)
6          b1.GetInformation()
7
8          'Используем конструктор по умолчанию
9          Dim b2 As Book = New Book()
10         b2.GetInformation()
11
12         Console.ReadLine()
13     End Sub
14 End Module

```

Если мы запустим код на выполнение, то консоль выведет нам информацию о книгах b1 и b2. Обратите внимание, что чтобы создать новый объект кроме конструктора нам надо использовать ключевое слово **New**. В первом случае мы используем свой конструктор, а во втором - конструктор по умолчанию.

Частичные классы

Частичные классы представляют возможность разделения одного класса на несколько файлов. Например, в одном файле может быть:

```

1  Partial Public Class Book
2      'Название книги
3      Dim name As String
4

```

```
5 End Class
```

а в другом:

```
1 Partial Public Class Book
2
3     'Автор
4     Dim author As String
5     'Год издания
6     Dim year As Integer
7
8 End Class
```

Для создания частичного класса перед его объявлением надо поставить ключевое слово **Partial**. В итоге в результате компиляции будет создан единый класс, который не будет отличаться от других классов.

Ключевое слово With

Предположим, что у нас есть следующий класс State:

```
1 Class State
2     Public capital As String
3     Public area As Integer
4     Public population As Integer
5
6     Sub New(cap As String)
7         capital = cap
8     End Sub
9 End Class
```

При создании объекта класса мы указываем столицу, значения для остальных полей мы должны указать отдельно, например так:

```
1 Dim statel As New State("City")
2 statel.area = 200
3 statel.population = 200
```

Однако таких свойств может быть множество. И чтобы сократить объем кода, мы можем использовать конструкцию **With...End With**. Чтобы применить эту конструкцию, после слова With указывается объект, а затем построчно свойства объекта, которые мы хотим получить или присвоить:

```
1 Dim capit As String
2 Dim statel As New State("City")
3
4 With statel
5     'Установка остальных полей
6     .area = 200
```

```
7      .population = 200
8      'Получение значение поля capital
9      capit = .capital
10     End With
```

Кроме того, мы можем использовать сокращенный синтаксис инициализации объекта с помощью ключевого слова `With` и одновременно указать все необходимые свойства:

```
1 Dim state2 = New With {.capital = "City2", .area = 300, .population = 300}
```

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



ТАКЖЕ НА METANIT.COM

Ассемблер MASM. Установка и начало ...

3 месяца назад · 4 коммент...

Ассемблер MASM.
Установка и начало
работы, Visual Studio, ...

ListView

2 месяца назад · 1 коммент...

ListView в JavaFX,
создание списков,
получение выбранных в ...

Встроенные компоненты ввода

5 месяцев назад · 1 коммен...

Встроенные компоненты
ввода Blazor из
пространства имен ...

Клиентские Form...

10 дне

Клиентские
на Xamarin
SignalR

G

Начать обсуждение...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS

?

Имя

♡

Поделиться

Лучшие

Новые

Старые

Прокомментируйте первым.

Подписаться

О защите персональных данных

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.