



# The C# Programming Language

## Professional

Отражение (Reflection).

# Рефлексия

## Reflection

Рефлексия (отражение) - это процесс, во время которого программа может отслеживать и модифицировать собственную структуру и поведение во время выполнения.

Парадигма программирования, положенная в основу отражения, называется рефлексивным программированием. Это один из видов метапрограммирования.

# Рефлексия

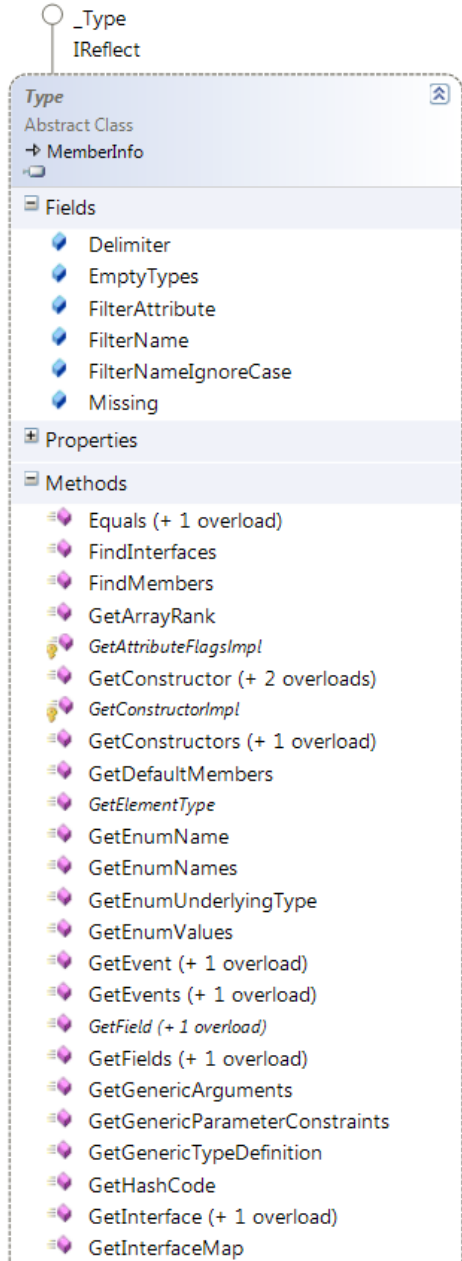
## System.Reflection

- System.Reflection
  - AmbiguousMatchException
  - Assembly
  - AssemblyAlgorithmIdAttribute
  - AssemblyCompanyAttribute
  - AssemblyConfigurationAttribute
  - AssemblyCopyrightAttribute
  - AssemblyCultureAttribute
  - AssemblyDefaultAliasAttribute
  - AssemblyDelaySignAttribute
  - AssemblyDescriptionAttribute
  - AssemblyFileVersionAttribute
  - AssemblyFlagsAttribute
  - AssemblyInformationalVersionAttribute
  - AssemblyKeyFileAttribute
  - AssemblyKeyNameAttribute
  - AssemblyName
  - AssemblyNameProxy
  - AssemblyProductAttribute
  - AssemblyTitleAttribute
  - AssemblyTrademarkAttribute
  - AssemblyVersionAttribute
  - Binder
  - ConstructorInfo
  - CustomAttributeData
  - CustomAttributeFormatException
  - DefaultMemberAttribute
  - EventInfo
  - ExceptionHandlingClause
  - FieldInfo
  - InvalidFilterCriteriaException
  - LocalVariableInfo
  - ManifestResourceInfo
  - MemberInfo
  - MethodBase
  - MethodBody
  - MethodInfo
  - Missing
  - Module
  - ObfuscateAssemblyAttribute
  - ObfuscationAttribute
  - ParameterInfo
  - PointerType

`System.Reflection` - пространство имен, которое содержит классы для применения рефлексии в языке C#.

# Type

## Класс



`Type` является корневым классом для функциональных возможностей рефлексии и основным способом доступа к метаданным.

С помощью членов класса `Type` можно получить сведения об объявленных в типе элементах: конструкторах, методах, полях, свойствах и событиях класса, а также о модуле и сборке, в которых развернут данный класс.

# Type

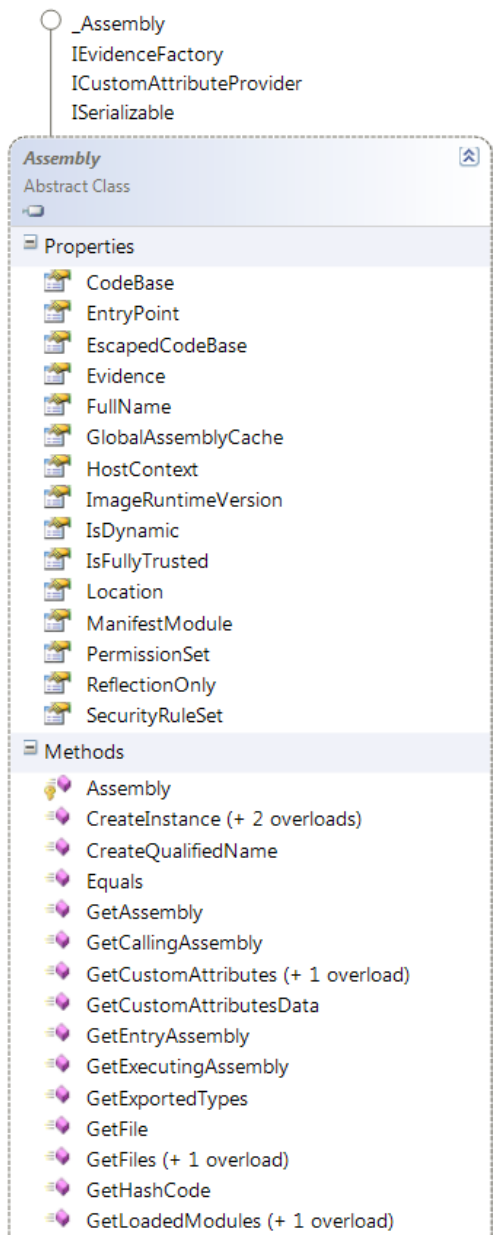
## Способы получения экземпляра

1. Вызов метода GetType( ) на экземпляре требуемого класса.
2. Вызов статического метода GetType( ) класса **Type**.
3. Использование оператора **typeof**( ).

В приведенных выше примерах результатом будет ссылка на объект **Type**, содержащий информацию о целевом типе .

# Assembly

## Класс



Класс `Assembly` представляет собой сборку, которая является модулем с возможностью многократного использования, поддержкой версий и встроенным механизмом описания общезыковой исполняющей среды.

# Сборка

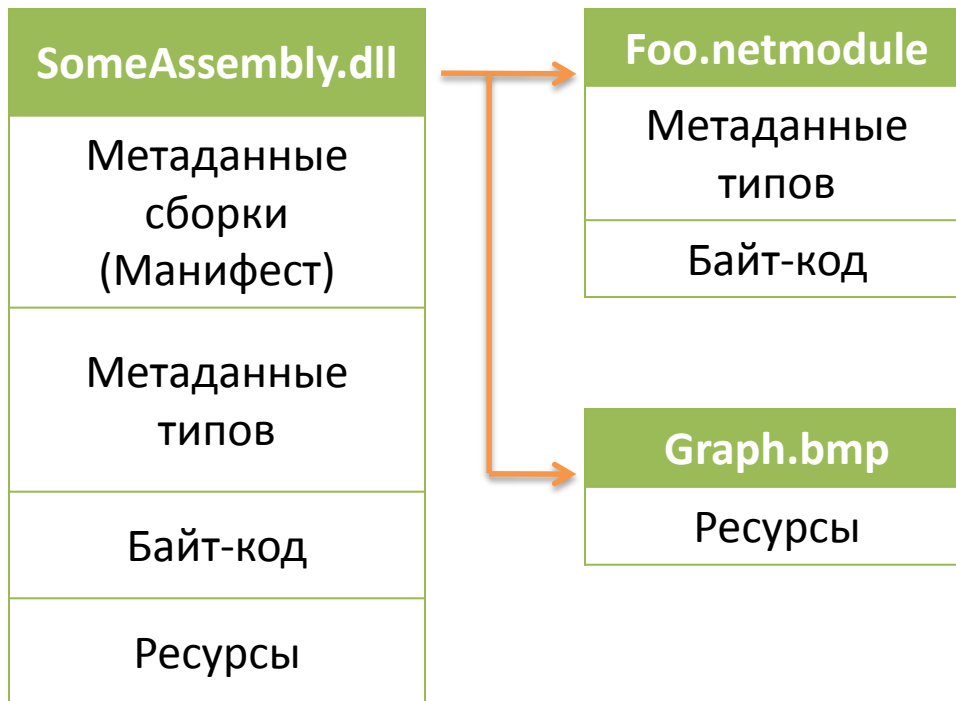
## Однофайловая

SomeAssembly.dll
Метаданные сборки (Манифест)
Метаданные типов
Байт-код
Ресурсы

Все компоненты хранятся в  
одном файле.

# Сборка

## Многофайловая



Метаданные сборки обязательно должны находиться в главном файле.

Метаданные типов, код и ресурсы могут храниться как в главном файле сборки, так и во вспомогательных файлах.



# Метаданные сборки

## Манифест

Метаданные сборки (манифест) состоят из описания сборки:  
имя, версия, строгое имя, информация о культуре.

# Метаданные типов

## Информация описывающая тип

Метаданные типов включают пространство имен и имя типа, члены типа и параметры, если имеются.

# Байт-код

## Byte-code

Байт-код (псевдокод) — машинно-независимый код низкого уровня, генерируемый транслятором и исполняемый интерпретатором. Большинство инструкций байт-кода эквивалентны одной или несколькими командам ассемблера. Трансляция в байт-код занимает промежуточное положение между компиляцией в машинный код и интерпретацией.

Байт-код называется так, потому что длина каждого кода операции — один байт, но длина кода команды различна. Каждая инструкция представляет собой однобайтовый код операции от 0 до 255, за которым следуют такие параметры, как регистры или адреса памяти.

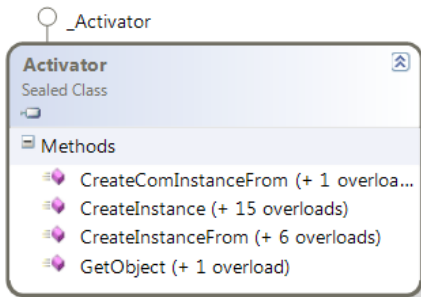
# Ресурсы

## Resources

Ресурсы – это объекты, которые используются кодом:  
строки, изображения, различные файлы.

# Activator

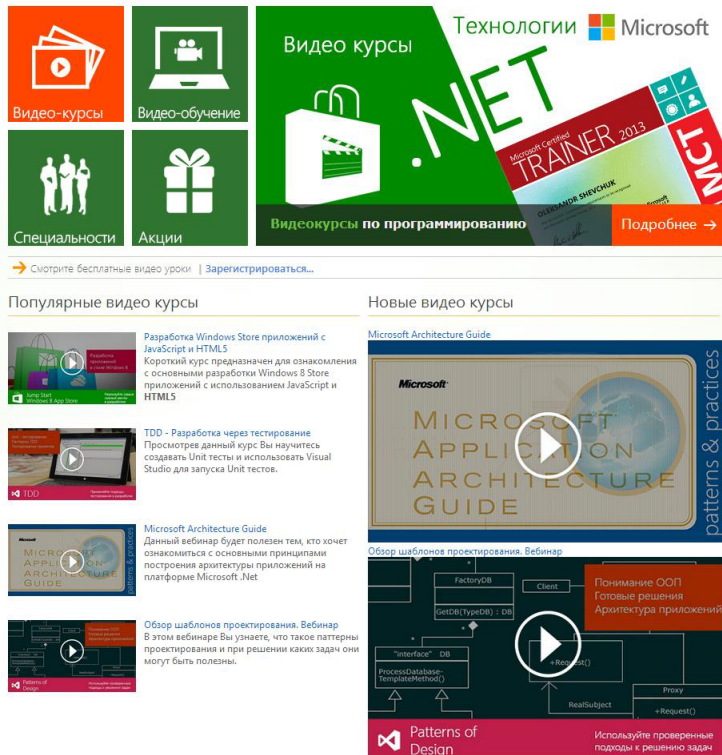
## Класс



Класс **Activator** содержит методы для локального создания типов объектов.

Метод `CreateInstance()` создает экземпляр типа, определенного в сборке путем вызова конструктора, который наилучшим образом соответствует заданным аргументам.

# Q&A



**Перейти к видеопорталу**  
[video.cbsystematics.com](http://video.cbsystematics.com)

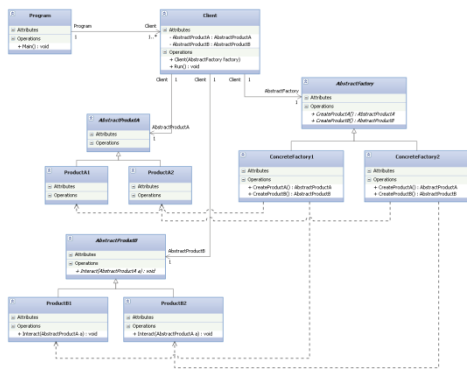
## Гарантия качества

Видео курсы Учебного центра CyberBionic Systematics - результат проверенной годами методики обучения программистов. Они разработаны сертифицированными тренерами Microsoft для учебного центра CyberBionic Systematics

## Преимущества видео обучения

- Вы можете просматривать учебный материал повторно необходимое количество раз
- Вы можете делать остановки в обучении для выполнения задания с учетом Вашей способности восприятия нового материала
- Вы обучаетесь у сертифицированных тренеров Microsoft

Видео курсы - это возможность обучаться самостоятельно, а также многократно просматривать и повторять материал учебного курса, если Вы обучаетесь очно или on-line. Мы рекомендуем видеообучение также специалистам, которым нужно систематизировать и углубить знания, полученные ранее в ВУЗе.



Задачи, с которыми сталкиваются разработчики программного обеспечения, как правило, довольно однотипны. Кроме того, в том или ином виде они уже были решены до нас. Шаблоны проектирования представляют собой коллекцию тщательно отобранных, наиболее общих принципов решения типовых проблем. Их высокий уровень абстракции позволяет отделить основные принципы реализации от конкретных прикладных областей, что, в свою очередь, дает прекрасную возможность не просто реализовывать шаблоны непосредственно на практике, но и использовать их как некий набор условных обозначений для четкой классификации даже самых сложных задач. В этом контексте, шаблоны проектирования являются неким общим языком, который исключает неоднозначность толкования и значительно ускоряет процесс разработки.

## Описание курса:

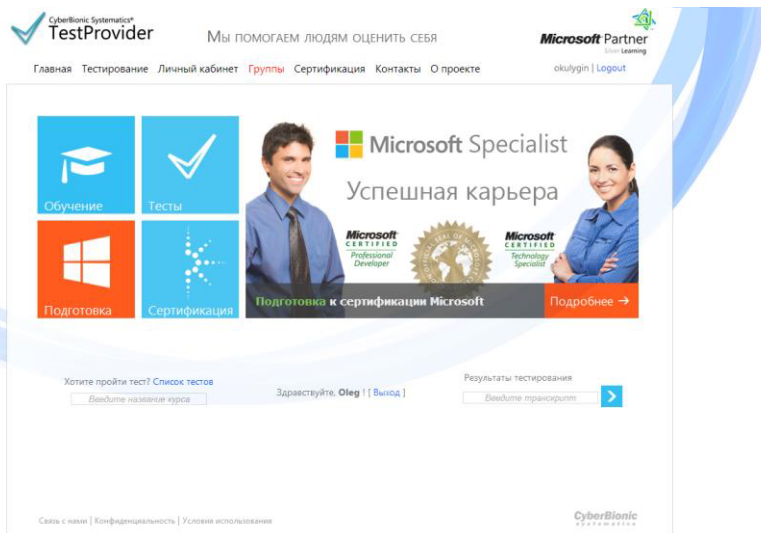
Курс "Шаблоны проектирования" поможет Вам в кратчайшие сроки освоить приемы проектирования. Вы сможете четко классифицировать задачи проектирования и однозначно описывать наиболее подходящие способы их решения. Каждый шаблон представляет собой инструмент, который Вы будете неоднократно использовать в своей практике, получая при этом все преимущества, которые дают надежные, проверенные временем решения.

**Длительность:**  
20 часов/10 дней.

Узнать более подробно о курсе на сайте:

[edu.cbsystematics.com](http://edu.cbsystematics.com)





Перейти к тестированию  
[www.TestProvider.com](http://www.TestProvider.com)

## Тестирование IT-специалистов

**TestProvider** обеспечивает надежную и объективную оценку технических знаний и опыта работы IT-специалиста с программными продуктами Microsoft.

## Сертификация IT-специалистов

Подтвердите ваш практический опыт работы с технологиями **Microsoft**, получив сертификацию, соответствующую той работе, которую вы выполняете сейчас или желаете получить в будущем.

Компании **Microsoft**, **CyberBionic Systematics** и **Intel** на базе портала [TestProvider](http://TestProvider) компании CyberBionic Systematics с использованием платформы Microsoft Azure совместно с Министерством науки и образования Украины проводят Всеукраинское дистанционное мониторинговое исследование уровня сформированности у выпускников учебных заведений навыков использования информационно-коммуникативных технологий в практической деятельности.

# **CyberBionic** **s y s t e m a t i c s**

*Coevolution of humans and machines.*