Абдрашитова Анжела ИУ5-31Б

Что такое рефлексия?

Рефлексия представляет собой процесс выявления типов во время выполнения приложения. Каждое приложение содержит набор используемых классов, интерфейсов, а также их методов, свойств и прочих кирпичиков, из которых складывается приложение. И рефлексия как раз и позволяет определить все эти составные элементы приложения. Рефлексия с одной стороны включает достаточно большое количество сведений. С другой стороны, рефлексия очень широко используется в современном языке C#. Фактически работа со сборками, типами данных и атрибутами – основная задача рефлексии.

Рефлексия позволяет: перечислять члены типа, создавать новые экземпляры объекта, запускать на выполнение члены объекта, извлекать информацию о типе, извлекать информацию о сборке, исследовать пользовательские атрибуты, примененные к типу, создавать и компилировать новые сборки.

Основной функционал рефлексии сосредоточен в пространстве имен **System.Reflection**. В нем мы можем выделить следующие основные классы:

- **Assembly**: класс, представляющий сборку и позволяющий манипулировать этой сборкой
- **AssemblyName**: класс, хранящий информацию о сборке
- **MemberInfo**: базовый абстрактный класс, определяющий общий функционал для классов EventInfo, FieldInfo, MethodInfo и PropertyInfo
- **EventInfo**: класс, хранящий информацию о событии
- **FieldInfo**: хранит информацию об определенном поле типа
- **MethodInfo**: хранит информацию об определенном методе
- **PropertyInfo**: хранит информацию о свойстве
- **ConstructorInfo**: класс, представляющий конструктор
- **Module**: класс, позволяющий получить доступ к определенному модулю внутри сборки
- **ParameterInfo**: класс, хранящий информацию о параметре метода

Строго говоря, рефлексия не является объектно-ориентированным механизмом. Для получения информации о типах разработчики языка C# могли создать отдельный языковой механизм. Но поскольку C# – объектно-ориентированный язык, то и рефлексию типов разработчики языка C# реализовали на основе объектно-ориентированного подхода. Type – это специализированный класс, содержащий информацию о других классах. Объект класса Type содержит информацию о конкретном классе. Получить информацию о типе можно двумя способами. Если создан

объект класса, то получить информацию о классе можно с помощью метода GetType.

```
ForInspection obj = new ForInspection();
Type t = obj.GetType();
```

Таким образом, класс Type содержит свойства и методы, позволяющие получить информацию о конструкторах, методах, свойствах, полях данных исследуемого класса, проверить от какого класса наследуется класс, какие реализует интерфейсы.