1. Рефлексия - это возможность программы анализировать свою структуру и поведение во время выполнения. С помощью рефлексии можно получать информацию о типах, членах типов (полях, свойствах, методах и т.д.), создавать объекты, вызывать методы и многое другое.

2. Три способа получения объекта типа Type:

- Использование оператора `typeof`: `Type type = typeof(МойТип);`

- Вызов метода `GetType()` на объекте: `Type type = myObject.GetType();`

- Использование статического метода `Type.GetType()` и передача полного имени типа в виде строки: `Type type = Type.GetType("ПолноеИмяТипа");`

3. Классы из пространства имен `System.Reflection` предоставляют средства для работы с рефлексией в .NET Framework. Некоторые из этих классов:

- `Assembly` - представляет сборку, которая является файлом, содержащим один или несколько модулей с кодом.

- `Type` - представляет тип, описывающий объекты, которые можно создать или получить в .NET.

- `MethodInfo` - представляет информацию о методе.

- `PropertyInfo` - представляет информацию о свойстве.

- `FieldInfo` - представляет информацию о поле.

- и другие.

4. Класс `System.Type` позволяет получать информацию о типе и взаимодействовать с ним. Некоторые из его свойств и методов:

- Свойство `Name` - возвращает имя типа.

- Свойство `FullName` - возвращает полное имя типа.

- Метод `GetMethods()` - возвращает массив объектов `MethodInfo`, представляющих методы типа.

- Метод `GetProperties()` - возвращает массив объектов `PropertyInfo`, представляющих свойства типа.

- Метод `InvokeMember()` - вызывает указанный член типа (метод, свойство, поле и т.д.).

- и другие.

5. Позднее связывание (также известное как динамическое связывание) - это процесс определения и вызова методов или доступа к членам объекта во время выполнения программы. При позднем связывании информация о типе и его членах получается во время выполнения программы.

Раннее связывание - это процесс определения и вызова методов или доступа к членам объекта на этапе компиляции программы, когда информация о типе доступна статически.

6. Сборку можно динамически загрузить в приложение с помощью класса `Assembly` и его метода `Load()`. Например: `Assembly assembly = Assembly.Load("MyAssembly");`

7. Позднее связывание (также известное как динамическое связывание) - это процесс определения и вызова методов или доступа к членам объекта во время выполнения программы. При позднем связывании информация о типе и его членах получается во время выполнения программы.

Раннее связывание - это процесс определения и вызова методов или доступа к членам объекта на этапе компиляции программы, когда информация о типе доступна статически.

8. BindingFlags - это перечисление флагов, которые определяют способ поиска членов типа при использовании рефлексии. Некоторые из возможных комбинаций флагов:

- `BindingFlags.Public` - искать только публичные члены.

- `BindingFlags.NonPublic`- искать только приватные члены.

- `BindingFlags.Instance` - искать только экземплярные члены.

- `BindingFlags.Static` - искать только статические члены.

- `BindingFlags.FlattenHierarchy` - включить в поиск члены унаследованных типов.

Используя комбинацию флагов `BindingFlags.NonPublic | BindingFlags.Instance`, можно получить доступ к приватным экземплярным членам класса.