**1.Что такое рефлексия?**

Возможности программы анализировать свою структуру (типы данных, методы, свойства и т.д.) во время выполнения. C# предоставляет набор классов в пространстве имен System.Reflection, которые позволяют получать информацию о типах и их членах, создавать новые типы и даже вызывать методы динамически во время выполнения.

**2. Какие есть способы получения объекта типа Type. Приведите три  
способа получения типа.**

**Typeof** - используется на этапе компиляции для получения объекта типа для указанного типа данных

**GetType** - вызывается у конкретного объекта во время выполнения, что позволяет получить объект типа для текущего экземпляра.

**Type.GetType** - используется для получения объекта типа по строковому представлению имени типа.

**3. Охарактеризуйте классы из пространства имен System.Reflection.**

• 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

**4. Как можно использовать класс System.Type? Перечислите его свойства  
и методы.**

Класс Type представляет изучаемый тип, инкапсулируя всю информацию о нем.

**Методы**: FindMembers() GetConstructors() GetEvents() GetFields() GetInterfaces()

GetMembers() GetMethods() GetProperties()

**Свойства**: Name Assembly Namespace IsArray IsClass IsEnum IsInterface

**5. Что такое позднее и раннее связывание?**

* **Раннее связывание -** это процесс определения связей между вызовом метода и его реализацией на этапе компиляции. Вызов метода объекта известного типа, на основе статической информации о типах.
* **Позднее связывание** - это процесс определения связей между вызовом метода и его реализацией во время выполнения программы. Вызов метода, когда тип объекта определяется динамически.

**6. Как динамически загрузить сборку в приложение?**

В C# динамическая загрузка сборки обычно выполняется с использованием класса Assembly из пространства имен System.Reflection.

* Assembly.LoadFile(path);
* Затем с помощью метода GetType получаем тип
* GetMethode
* Invoke

**7. Для чего предназначены BindingFlags? Какую комбинацию флагов  
необходимо использовать, чтобы иметь возможность получать  
приватные члены класса?**

BindingFlags - это перечисление в пространстве имен System.Reflection, которое предоставляет различные опции для определения, каким образом искать члены типа (методы, свойства, поля и т.д.) при использовании рефлексии.

Некоторые из часто используемых флагов в BindingFlags:

* Public: Искать только общедоступные члены.
* NonPublic: Искать только члены с ограниченным доступом (не общедоступные).
* Instance: Искать только члены экземпляра.
* Static: Искать только статические члены.
* IgnoreCase: Искать с учетом регистра или без учета регистра.
* DeclaredOnly: поиск должен осуществляться только в текущем типе и не включать члены унаследованных базовых типов.

Для получения приватных членов класса, вам нужно использовать комбинацию флагов, указывающую на необходимость поиска членов с ограниченным доступом (NonPublic).

BindingFlags.NonPublic | BindingFlags.Instance