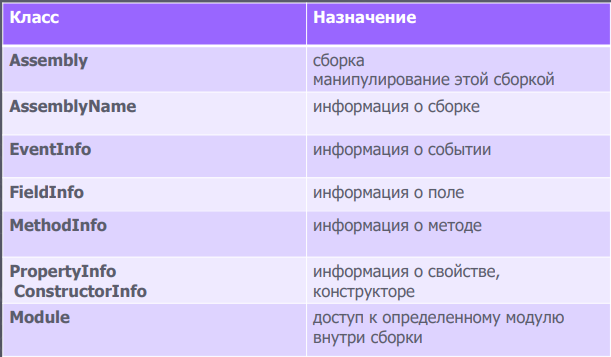
1. Что такое Рефлексия?

**Рефлексия** – процесс выявления типов, членов, сборках и других данных во время выполнения приложения

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

1) Type t = admin.GetType (); -экземпляр класса  
2) Type t1 = Type.GetType (“System.Int32”); -статический метод  
3) Type t2 = typeof (Point)

3. Охарактеризуйте классы из пр-ва имен System.Reflection



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

**SystemType** – класс, позволяет получить информацию о членах типа

**Методы:**  
**FindMembers()** – возвращает массив объектов MemberInfo данного типа  
**GetConstructors()** – конструкторы д-го типа в виде набора объектов ConstructorInfo  
**GetEvents()** – события д-го типа в виде массива объектов EventInfo  
**GetFields()** – поля д-го типа в виде массива объектов FieldInfo  
**GetInterfaces()** – реализуемые д-м типом интерфейсы в виде массива объектов Type  
**GetMembers()** – члены типа в виде массива объектов MemberInfo  
**GetMethods()** – методы типа в виде массива объектов MethodInfo  
**GetProperties()** – свойства в виде массива объектов PropertyInfo

**Свойства:**  
**IsAbstract** – тру, если тип абстрактный  
**IsArray** – тру, если тип - массив  
**IsClass** – тру, если тип - класс  
**IsEnum** – тру, если тип - перечисление  
**IsInterface** – тру, если тип – интерфейс

5. Что такое позднее и раннее связывание?

Это два способа вызова методов в C#.

Раннее связывание означает, что тип объекта и метод, который будет вызван, определяются на этапе компиляции.

Позднее связывание означает, что тип объекта и метод, который будет вызван, определяются на этапе выполнения .

6. Как динамически загрузить сборку в приложение?

При позднем связывании.

Динамически загрузить сборку в приложение можно с помощью класса Assembly из пространства имен System.Reflection. Он предоставляет статические методы для загрузки сборок по имени, пути или потоку.

7. Что такое позднее (раннее) связывание?

Позднее связывание – процесс динамической загрузки сборок и типов при работе приложения, создание экземпляров типов и работу с их элементами

Позволяет создавать расширяемые приложения, когда дополнительный функционал программы неизвестен и его могут разработать и подключить сторонние разработчики

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

Это перечисление, которое определяет флаги, используемые для управления привязкой и поиском членов типа с помощью рефлексии. Они позволяют указать, какие виды членов (статические, экземплярные, публичные, приватные и т.д.) должны быть включены в результаты поиска.

Чтобы иметь возможность получать приватные члены класса, необходимо использовать комбинацию флагов BindingFlags.NonPublic и BindingFlags.Instance или BindingFlags.Static, в зависимости от того, являются ли члены экземплярными или статическими.

Например: Type t = typeof(MyClass); FieldInfo[] fields = t.GetFields(BindingFlags.NonPublic | BindingFlags.Instance);