 **Для чего используют статические классы?**  
Статические классы используются для создания контейнера для методов, свойств или полей, которые не привязаны к конкретному экземпляру класса. Такие классы удобны для реализации вспомогательных функций или утилит, которые должны быть доступны глобально в приложении.

 **Что может содержать статический класс?**  
Статический класс может содержать только статические методы, свойства, поля и события. Нельзя создавать экземпляры статического класса, и он не может содержать конструкторы экземпляров.

 **Что такое производный и базовый классы?**

* **Базовый класс** — это класс, от которого наследуются другие классы. Он может предоставлять общую функциональность для производных классов.
* **Производный класс** — это класс, который наследуется от базового. Он может добавлять новую функциональность или переопределять методы базового класса.

 **Как используют ключевое слово base?**  
Ключевое слово base используется в производном классе для обращения к членам базового класса, например, для вызова конструктора базового класса или доступа к его методам и свойствам.

 **В чем заключена основная задача наследования?**  
Основная задача наследования заключается в возможности использовать функциональность базового класса в производных классах, расширять или изменять её поведение, а также повторно использовать код.

 **Пусть базовый класс содержит метод basefunc(), а производный класс не имеет метода с таким именем. Может ли объект производного класса иметь доступ к методу basefunc()? Если да, то при каких условиях?**  
Да, объект производного класса может вызвать метод basefunc() при условии, что метод имеет модификатор доступа public, protected или internal в базовом классе.

 **Напишите объявление конструктора без аргументов для производного класса B, который будет вызывать конструктор без аргументов базового класса A.**

csharp

Копировать код

public class B : A

{

public B() : base()

{

}

}

 **Что такое полиморфизм? Приведите пример.**  
Полиморфизм — это возможность работы с объектами производных классов через интерфейсы или базовые классы, предоставляя единый интерфейс для разных реализаций.  
Пример:

csharp

Копировать код

public class Animal

{

public virtual void Speak()

{

Console.WriteLine("Animal speaks");

}

}

public class Dog : Animal

{

public override void Speak()

{

Console.WriteLine("Dog barks");

}

}

Animal myAnimal = new Dog();

myAnimal.Speak(); // Вызовет метод Speak() из класса Dog

 **Определите назначение виртуальных функций.**  
Виртуальные функции позволяют производным классам переопределять реализацию методов, определённых в базовом классе. Это обеспечивает возможность изменять поведение метода в производном классе.

 **Кому доступны переменные с модификатором protected?**  
Переменные с модификатором protected доступны в пределах самого класса и его производных классов.

 **Наследуются ли переменные с модификатором private?**  
Переменные с модификатором private наследуются, но недоступны в производных классах.

 **as, is — что это, как применяется? В чем между ними отличие?**

* is используется для проверки, является ли объект заданного типа.
* as используется для безопасного приведения типа. Если приведение невозможно, возвращается null.

 **Поддерживает ли C# множественное наследование?**  
Нет, C# не поддерживает множественное наследование классов, но поддерживает множественное наследование интерфейсов.

 **Можно ли запретить наследование от класса?**  
Да, использование ключевого слова sealed запрещает наследование от класса.

 **Можно ли разрешить наследование класса, но запретить перекрытие метода?**  
Да, для этого нужно объявить метод с модификатором sealed.

 **Что такое абстрактный класс?**  
Абстрактный класс не может быть создан как экземпляр и служит в качестве базового класса для других классов. Он может содержать абстрактные методы без реализации.

 **В каком случае вы обязаны объявить класс абстрактным?**  
Если в классе есть хотя бы один абстрактный метод, то класс должен быть объявлен абстрактным.

 **В чем разница между абстрактными и виртуальными классами? Между виртуальными и абстрактными методами?**  
Абстрактные классы нельзя создать напрямую, а виртуальные классы в C# отсутствуют. Виртуальные методы могут иметь реализацию по умолчанию, а абстрактные методы — нет.

 **Какие компоненты класса могут быть виртуальными?**  
Методы, свойства, события и индексаторы.

 **Что такое интерфейс?**  
Интерфейс — это набор методов и свойств, которые класс должен реализовать.

 **Что может содержать интерфейс?**  
Интерфейс может содержать методы, свойства, события и индексаторы.

 **Как работать с объектом через унаследованный интерфейс?**  
Можно присвоить объект переменной интерфейса и вызвать методы интерфейса.

 **Приведите пример явной реализации интерфейса.**

csharp

Копировать код

public interface IExample

{

void DoWork();

}

public class Example : IExample

{

void IExample.DoWork()

{

Console.WriteLine("Работа выполнена.");

}

}

 **Почему нельзя указать модификатор видимости для методов интерфейса?**  
Все методы интерфейса по умолчанию являются public и не могут иметь другой модификатор видимости.

 **Можно ли наследовать от нескольких интерфейсов?**  
Да, класс может наследовать несколько интерфейсов.

 **Назовите отличия между интерфейсом и абстрактным классом.**

* Интерфейс не может содержать реализации, абстрактный класс — может.
* Класс может реализовать несколько интерфейсов, но наследовать только один абстрактный класс.

 **Для чего используются стандартные интерфейсы ICloneable, IComparable, IComparer, IEnumerable?**

* ICloneable — для создания копий объектов.
* IComparable — для сравнения объектов.
* IComparer — для пользовательской сортировки объектов.
* IEnumerable — для перебора коллекций.