1. Для чего используют статические классы?

Статические классы объявляются с модификатором static и могут содержать только статические поля, свойства и методы Статические классы объявляются с модификатором static и могут содержать только статические поля, свойства и методы. По сути же, static классы - это классы содержащие глобальные переменные. static поля в публичных классах содержат неизменные данные для вообще всех экземпляров класса.

[Статический](https://learn.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/static) класс в основном такой же, как и нестатический класс, но имеется одно отличие: нельзя создавать экземпляры статического класса.

1. Что может содержать статический класс? 3

Статический класс может использоваться как обычный контейнер для наборов методов, статические поля и конструкторы статич????????????

. Что такое производный и базовый классы?

**Класс, члены которого наследуются, называется базовым классом.** **Класс, который наследует члены базового класса, называется производным классом**

1. Как используют ключевое слово base?

**Ключевое слово base используется** для доступа к членам базового из производного класса в следующих случаях: Вызов метода базового класса, который был переопределен другим методом. Определение конструктора базового класса, который должен вызываться при создании экземпляров производного класса

1. В чем заключена основная задача наследования?

Наследование (inheritance) является одним из ключевых моментов ООП. Благодаря наследованию один класс может унаследовать функциональность другого класса

1. 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, всегда ПРОВ public class A

{

private int \_value = 10;

public class B : A

{

public int GetValue()

{

return \_value;

}

}

}

public class C : A

{

// public int GetValue()

// {

// return \_value;

// }

}

public class AccessExample

{

public static void Main(string[] args)

{

var b = new A.B();

Console.WriteLine(b.GetValue());

}

}

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

8. Что такое полиморфизм? Приведите пример.

9. Определите назначение виртуальных функций. 10. Кому доступны переменные с модификатором protected? 11. Наследуются ли переменные с модификатором private? 12. As, is – что это, как применяется? В чем между ними отличие ? 13. Поддерживает ли C# множественное наследование? 14. Можно ли запретить наследование от класса? 15. Можно ли разрешить наследование класса, но запретить перекрытие метода? 16. Что такое абстрактный класс? 17. В каком случае вы обязаны объявить класс абстрактным? 18. В чем разница между абстрактными и виртуальными классами? Между виртуальными и абстрактными методами? 19. Какие компоненты класса могут быть виртуальными? 20. Что такое интерфейс?

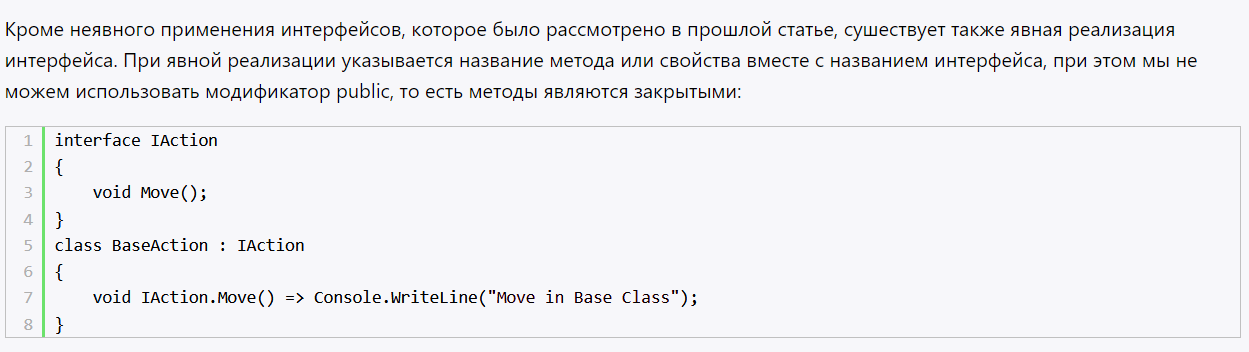
21. Что может содержать интерфейс?

* Методы
* Свойства
* Индексаторы
* События
* Статические поля и константы (начиная с версии C# 8.0)

22. Как работать с объектом через унаследованный интерфейс?



23. Приведите пример явной реализации интерфейса.



24. Почему нельзя указать модификатор видимости для методов интерфейса?

**Потому что все они должны иметь модификатор public, который и установлен по умолчанию**.

25. Можно ли наследовать от нескольких интерфейсов?

Один интерфейс может наследовать другой. Синтаксис наследования интерфейсов такой же, как и у классов. Когда в классе реализуется один интерфейс, наследующий другой, в нем должны быть реализованы все члены, определенные в цепочке наследования интерфейсов.

26. Назовите отличия между интерфейсом и абстрактным классом.

**1.Интерфейс описывает только поведение. У него нет состояния. А у абстрактного класса состояние есть: он описывает и то, и другое**

2. **Классы могут реализовывать сколько угодно интерфейсов, но наследоваться можно только от одного класса.**

3. **Абстрактный класс связывает между собой и объединяет классы, имеющие очень близкую связь. В то же время, один и тот же интерфейс могут реализовать классы, у которых вообще нет ничего общего.**

27. Для чего используются стандартные интерфейсы ICloneable, IComparable, IComparer, lEnumerable?

интерфейс IComparable задает метод сравнения объектов по принципу больше или меньше, что позволяет выполнять их сортировку

Интерфейс IComparable определен в пространстве имен System. Он содержит всего один метод CompareTo, возвращающий результат сравнения двух объектов — текущего и переданного ему в качестве параметра

ICloneable — клонировать объекты

Интерфейс IComparer определен в пространстве имен System.Collections. Он содержит один метод CompareTo, возвращающий результат сравнения двух объектов, переданных ему в качестве параметров

