1. Для чего используют статические классы?

Статические классы используются для группировки методов и свойств, которые не требуют создания экземпляров класса. Они обеспечивают удобный доступ к функциям и переменным, которые могут быть общими для всех объектов.

2. Что может содержать статический класс?

Статический класс может содержать:

- Статические методы

- Статические свойства

- Статические конструкторы

- Статические поля

3. Что такое производный и базовый классы?

Базовый класс — это класс, от которого наследуются другие классы. Производный класс — это класс, который наследует функциональность базового класса и может добавлять или изменять её.

4. Как используют ключевое слово ` base `?

Ключевое слово `base` используется для доступа к членам базового класса из производного класса. Оно позволяет вызывать конструкторы базового класса и доступ к его методам и свойствам.

5. В чем заключена основная задача наследования?

Основная задача наследования — это повторное использование кода и создание иерархий классов, где производные классы могут расширять и модифицировать функциональность базового класса.

*Наследование* − это механизм получения нового класса на основе уже существующего

6. Может ли объект производного класса иметь доступ к методу `basefunc()`?

Да, если метод `basefunc()` объявлен как `public` или `protected` в базовом классе, производный класс может иметь к нему доступ.

7. Объявление конструктора без аргументов для производного класса B:

8. Что такое полиморфизм? Приведите пример.

Полиморфизм - способность объекта изменять свое поведение в зависимости от контекста его использования.

9. Определите назначение виртуальных функций.

Виртуальные функции позволяют производным классам переопределять методы базового класса, обеспечивая гибкость и возможность реализации полиморфизма.

10. Кому доступны переменные с модификатором `protected`?

Переменные с модификатором `protected` доступны только в самом классе и в производных классах.

11. Наследуются ли переменные с модификатором `private`?

Нет, переменные с модификатором `private` не наследуются в производных классах.

12. as, is – что это, как применяется? В чем между ними отличие?

- is проверяет, является ли объект экземпляром указанного типа. Возвращает булевское значение, говорящее о том, можете ли вы преобразовать данное выражение в указанный тип

- as выполняет безопасное приведение типов; если приведение невозможно, возвращает `null`, позволяет преобразовывать тип в определенный ссылочный тип.

13. Поддерживает ли C# множественное наследование?

Запрещено множественное наследование классов (но не интерфейсов)

14. Можно ли запретить наследование от класса?

Да, для этого используется модификатор sealed.

15. Можно ли разрешить наследование класса, но запретить перекрытие метода?

Да, для этого используется модификатор `sealed` на виртуальном методе.

16. Что такое абстрактный класс?

Абстрактный класс — это класс, который не может быть инстанцирован и может содержать абстрактные методы, которые должны быть реализованы в производных классах.

Служит только для порождения потомков - предоставляют базовый функционал для классов-наследников.

Задает интерфейс для всей иерархии

Может содержать и полностью определенные методы, переменные, конструкторы, свойства

Создавать объект абстрактного класса нельзя!!!!!!! (ссылку можно)

17. В каком случае вы обязаны объявить класс абстрактным?

Если класс имеет хотя бы одно абстрактное свойство или метод, то он должен быть определен как абстрактный

18. В чем разница между абстрактными и виртуальными классами?

- Абстрактные классы не могут быть инстанцированы и могут содержать абстрактные методы.

- Виртуальные классы могут быть инстанцированы, и их методы могут быть переопределены.

19. Какие компоненты класса могут быть виртуальными?

Методы, свойства, индексаторы могут быть виртуальными.

20. Что такое интерфейс?

Интерфейс - перечень сигнатур методов, предназначенных для

реализации.

21. Что может содержать интерфейс?

Интерфейс может содержать:

- Методы

- Свойства

- События

- Индексаторы

22. Как работать с объектом через унаследованный интерфейс?

Объект может быть приведён к интерфейсу, и его методы могут быть вызваны через этот интерфейс:

IMyInterface obj = new MyClass();

obj.MyMethod();

23. Приведите пример явной реализации интерфейса.

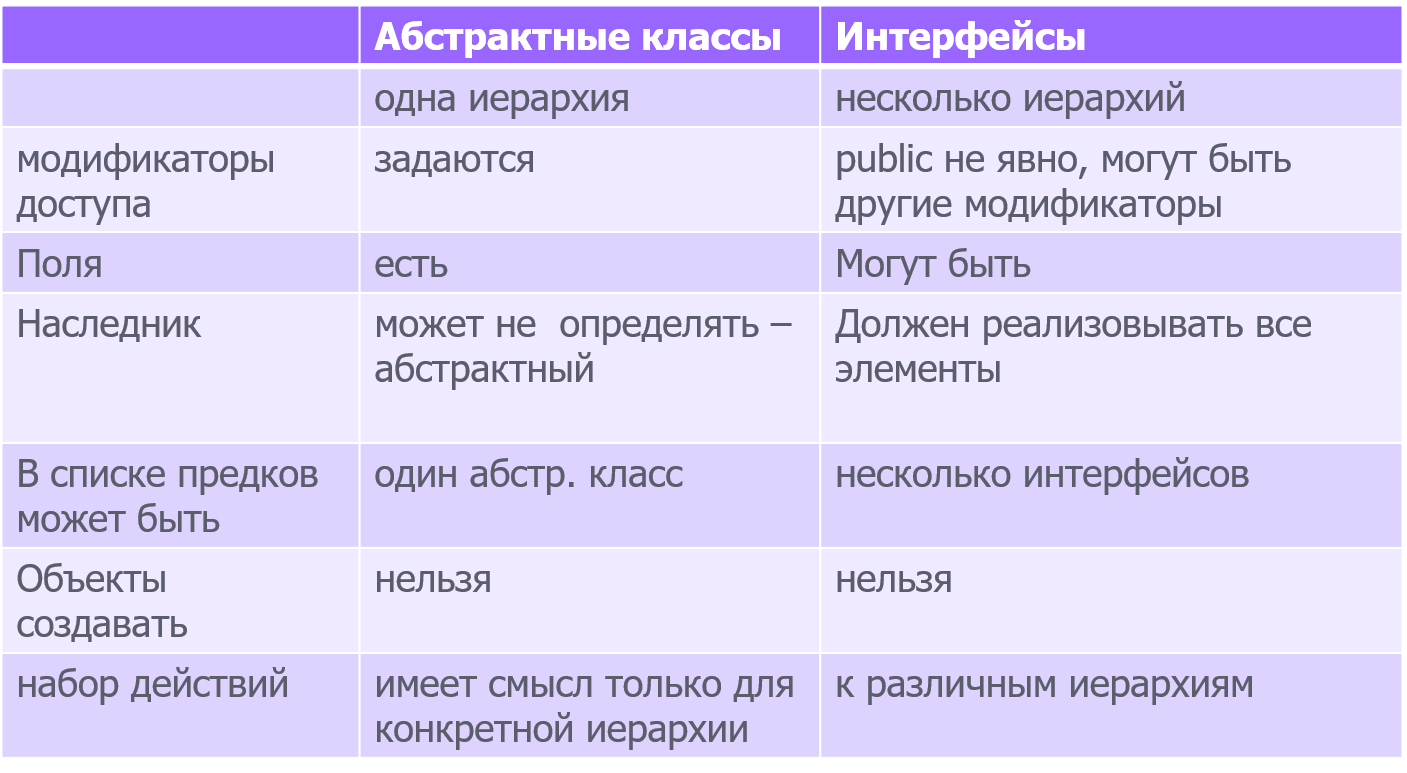
24. Почему нельзя указать модификатор видимости для методов интерфейса?

Методы интерфейсов всегда публичные по определению, так как интерфейс предназначен для предоставления общего контракта.

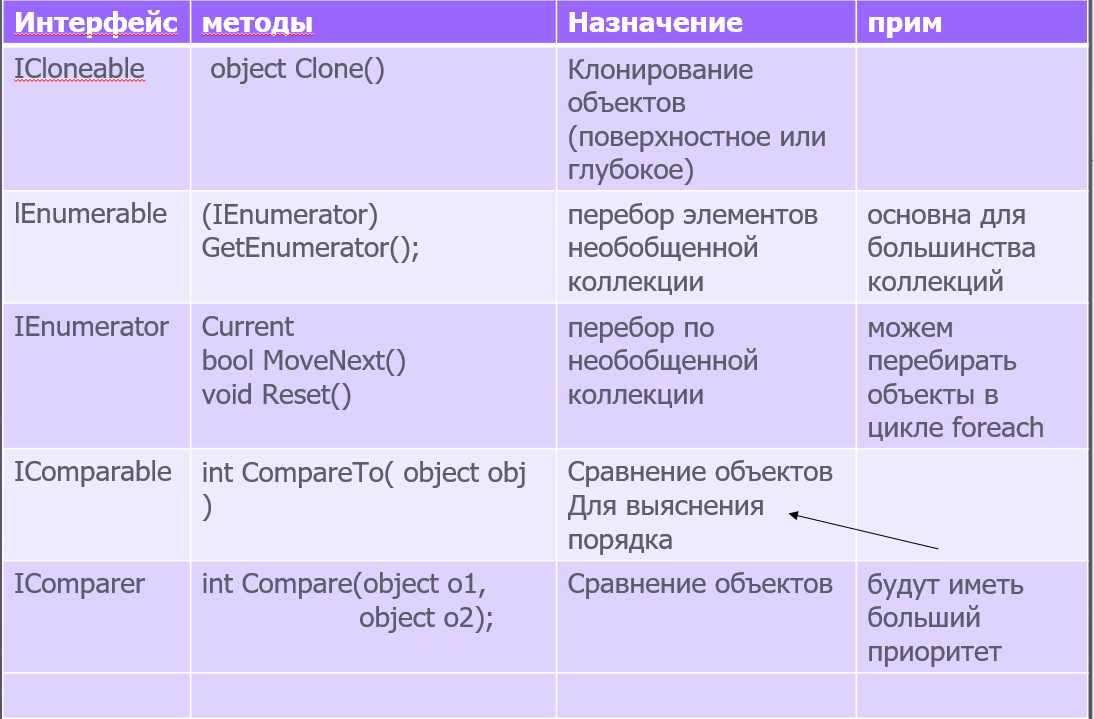
25. Можно ли наследовать от нескольких интерфейсов?

Да, Интерфейс или класс может наследовать свойства нескольких интерфейсов, в этом случае предки перечисляются через запятую.

26. Назовите отличия между интерфейсом и абстрактным классом.



27. Для чего используются стандартные интерфейсы: `ICloneable`, `IComparable`, `IComparer`, `IEnumerable`?



28. В какой строке приведенного ниже фрагмента листинга не содержится ошибки?

Строка \*\*2\*\* не содержит ошибки

29. Что будет выведено на консоль в результате выполнения следующего фрагмента?

На консоль будет выведено: 3 4

- В `Console.Write(x + " " + base.x);` выводится `3` (значение из `B`) и `4` (значение из `A`).

30. Что будет выведено на консоль в результате выполнения следующего фрагмента кода?

На консоль будет выведено: A B

- Вызов `a.mA();` обращается к методу `mA()` базового класса `A`, который выводит `"A "`.

- Вызов `b.mA();` обращается к переопределенному методу `mA()` в классе `B`, который выводит `"B "`.

31. Чем может быть M4, если дано следующее определение:

M4 - только интерфейс.

32. Выберите верное присваивание для объектов, определенных в листинге.

2) a = b;

33. Что будет выведено на консоль в результате выполнения следующего фрагмента, если раскомментировать строчку 1?

3) A B

34. В какой строке приведенного ниже фрагмента листинга содержится ошибка?

Ошибка содержится в строке \*\*2\*\*.

35. Почему приведенный ниже фрагмент листинга содержит ошибку?

Ошибка возникает в строке \*\*4\*\*.

36. В какой строке может быть ошибка компиляции?

Ошибка компиляции может возникнуть в строке \*\*4\*\*.

37. Что будет выведено на консоль в результате выполнения следующего фрагмента листинга:

На консоль будет выведено: new G