Ответы на вопросы

1. Для чего используют статические классы?

Статический класс может использоваться как обычный контейнер для наборов методов, работающих на входных параметрах, и не должен возвращать или устанавливать каких-либо внутренних полей экземпляра.

1. Что может содержать статический класс?

Статические классы объявляются с модификатором static и могут содержать только статические поля, свойства и методы.

1. Что такое производный и базовый классы?

Базовым называется класс, который наследуется, производным – кто наследует.

1. Как используют ключевое слово base?

Для доступа к членам базового из производного класса.

1. В чем заключена основная задача наследования?

Облегчение и уменьшение кода. Также позволяет избавиться от дублирования кода и облегчает его редактирование.

6. 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?

Может, только если метод будет иметь модификатор доступа protected или public

7. Напишите объявление конструктора без аргументов для производного класса B,

который будет вызывать конструктор без аргументов базового класса A.

Class Prim{

public string name;

public Prim(\_name){

name = \_name

}

}  
class Prim1: Prim{

Public Prim1

: base(name) {}

}

8. Что такое полиморфизм? Приведите пример.

Полиморфизм — это способность обьекта использовать методы производного класса, который не существует на момент создания базового.

Примером полиморфизма является перегрузка функций

9. Определите назначение виртуальных функций.

Виртуальными являются функции с ключевым словом virtual. Они позволяют переопределять функции.

10. Кому доступны переменные с модификатором protected?

В любом месте текущего класса или в производных классах.

11. Наследуются ли переменные с модификатором private?

Доступен только в том же классе или контексте.

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

Применяются для работы с типами данных. As – для преобразования типа, а is для проверки принадлежит ли переменная данному типу, возвращает false или true.

13. Поддерживает ли C# множественное наследование?

Нет. Поддерживает транзитивное.

14. Можно ли запретить наследование от класса?

Да, с помощью ключевого слова sealed.

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

 Да. Указываем класс как public, а метод как sealed.

16. Что такое абстрактный класс?

Класс с ключевым словам abstract и от которого нельзя создать объект.

Этот класс предназначен только для того, чтоб быть базовым классом для других классов.

17. В каком случае вы обязаны объявить класс абстрактным?

Когда от этого класса не нужно создавать объектов.

18. В чем разница между абстрактными и виртуальными классами? Между

виртуальными и абстрактными методами?

Ключевое слово virtual говорит о возможности переопределения, а abstract – о том, что обязано быть реализовано.

19. Какие компоненты класса могут быть виртуальными?

Методы, св-ва, индексаторы и события.

20. Что такое интерфейс?

Интерфейс представляет ссылочный тип, который может определять некоторый функционал. Затем этот функционал обязаны реализовать классы и структуры, которые его применяют

21. Что может содержать интерфейс?

Статические конструкторы, поля, константы или операторы.

22. Как работать с объектом через унаследованный интерфейс?

23. Приведите пример явной реализации интерфейса.

Interface Iprimer(){

Void Prikol();

}

class Primer{

void Iprimer.Prikol(){

//some code

}

}

24. Почему нельзя указать модификатор видимости для методов интерфейса?

Потому что они все имеют модификатор public по умолчанию

25. Можно ли наследовать от нескольких интерфейсов?

Да

26. Назовите отличия между интерфейсом и абстрактным классом.

Интерфейс может определять функционал разрозненных объектов, а абстрактный класс лишь связанных.

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

ICloneable – позволяет копировать объекты.

IComparable – позволяет сортировать объекты.

IComparer – Позволяет сравнивать объекты.

IEnumereble – позволяет перебирать объекты в цикле foreach.

28. 2

29. 3 4

30. A B

31. 1

32.

33. B

34. 2

35. Потому что создаётся объект от абстрактного класса.

36.

37. new G