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.

2. Что может содержать статический класс?

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

3. Что такое производный и базовый классы?

Базовый класс – класс, на основе которого создаются другие классы.  
Произвольный класс – класс, полученный на основе базового класса.

4. Как используют ключевое слово base?

Это ключевое слово можно применять подобно любой другой переменной экземпляра, но его можно использовать только внутри блока конструктора экземпляра, метода экземпляра или средства доступа к свойству. Применение его в статических методах не допускается.

Для вызова конструктора базового класса используется ключевое слово *base*.

5. В чем заключена основная задача наследования?

Роль наследования – формирование иерархии классов.

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?

Да, может, если метод открытый.

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

Public B(): base(){ //тело конструктора }

8. Что такое полиморфизм? Приведите пример.

Полиморфизм – способность к изменению функционала, унаследованного от базового класса.

9. Определите назначение виртуальных функций.

Виртуальный метод отличается тем, что он может быть переопределен в одном или нескольких производных классах. Виртуальные функции предоставляют механизм позднего или динамического связывания (высокая гибкость выполняемой программы, возможность реакции на события).

10. Кому доступны переменные с модификатором protected?

Такой член класса доступен из любого места в текущем классе или в производных классах.

11. Наследуются ли переменные с модификатором private?

Да, но они будут недоступны.

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

Операция **is** возвращает булевское значение, говорящее о том, можете ли вы преобразовать данное выражение в указанный тип.  
*операнд is <тип>*  
Операция **as** позволяет преобразовать тип в определенный ссылочный тип с применение следующего синтаксиса:  
*операнд as <тип>*  
Возвращает true если 1) операнд имеет тип, заданный в <тип>; 2) операнд может быть неявно преобразовать в <тип>; 3) операнд может быть упакован в <тип>;

13. Поддерживает ли C# множественное наследование?

Классов – нет, интерфейсов – да

14. Можно ли запретить наследование от класса?

Да. sealed class A {}

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, экземпляры которого создавать запрещено. Служит для порождения потомков – предоставляет базовый функционал для классов-наследников. Задает интерфейс для всей иерархии.

17. В каком случае вы обязаны объявить класс абстрактным?

Если один из методов или свойств является абстрактным.

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

Виртуальный метод имеет реализацию и м.б. переопределен в производном классе. Абстрактный метод не имеет реализации, только описание метода, который д.б. реализован в производных классах.

19. Какие компоненты класса могут быть виртуальными?

Методы, Свойства, Индексаторы, События

20. Что такое интерфейс?  
  
Набор абстрактных методов, свойств, событий и индексаторов, которые должны быть реализованы в производных классах.

21. Что может содержать интерфейс?

Методы, свойства, индексаторы и события.

НЕ могут содержать члены данных, в них нельзя определить конструкторы, деструкторы, операторные методы.

22. Как работать с объектом через унаследованный интерфейс?

Реализовать весь функционал, представленный в интерфейсе.

23. Приведите пример явной реализации интерфейса.

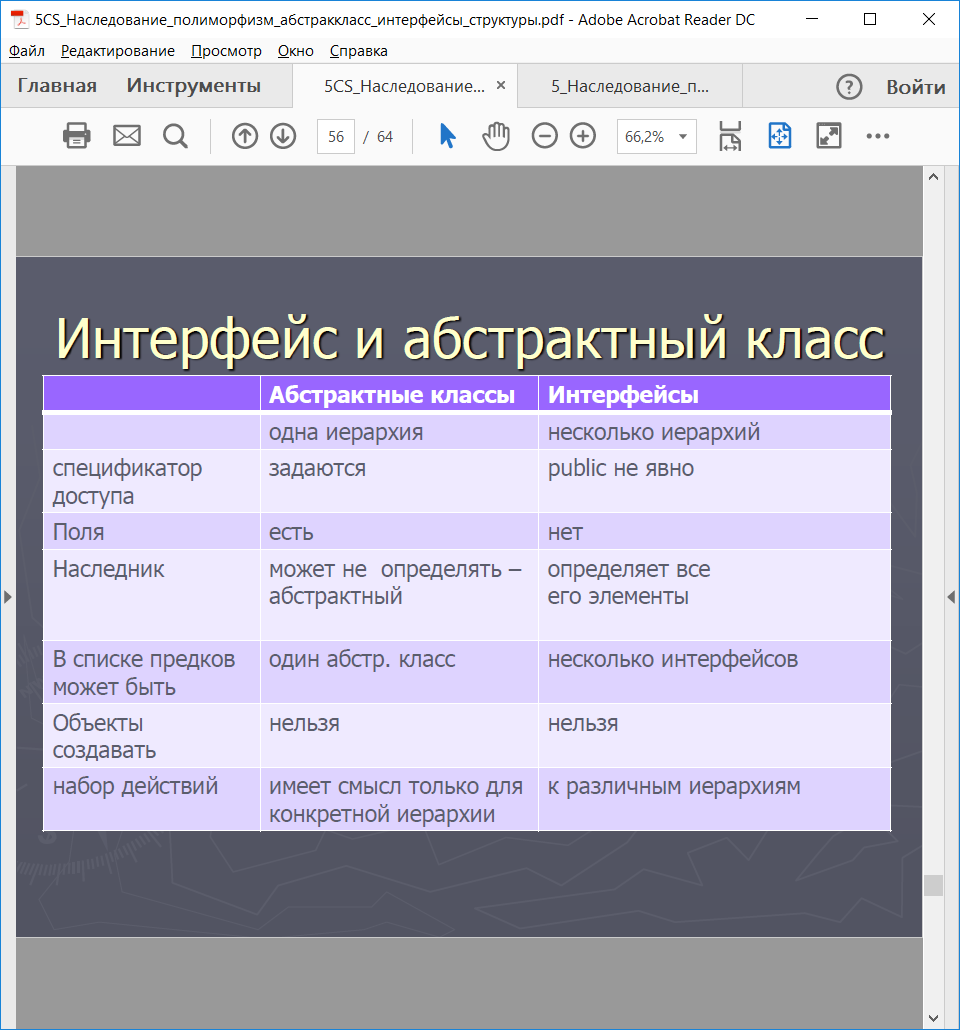
interface IDo { void Sleep(); }  
public class ExplicitDo: IDo { void IDo.Sleep() {} }

24. Почему нельзя указать модификатор видимости для методов интерфейса?

Они обязательно должны быть реализованы (public по умолчанию).

25. Можно ли наследовать от нескольких интерфейсов?

Да

26. Назовите отличия между интерфейсом и абстрактным классом. 

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

ICloneable – object Clone() - клонирование объектов  
IComparable – int CompareTo(object obj) - сравнение объектов  
IComparer – int Compare(object o1, object o2) – сравнение объектов  
lEnumerable – GetEnumerator() – перебор элементов необобщенной коллекции

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

2

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

3 4

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

А В

31. Чем может быть M4 если дано следующее определение:

1) M4 - только интерфейс

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

1) b = a;

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

3) A B

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

2

35. Почему приведенный ниже фрагмент листинга содержит ошибку?

4 – нельзя создавать экземпляр абстрактного класса

36. В какой строке может быть ошибка компиляции?

нету ошибок

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

new G