Класс это абстрактный тип данных, определяемый программистом (пользователем). С помощью классов определяются свойства объектов. Объекты это экземпляры класса. Создание экземпляра (объекта) класса осуществляется с помощью оператора new:

Имя\_класса имя\_объекта = new имя\_класса();

**1. Назовите принципы ООП. Поясните каждый из них.**

*Инкапсуляция*

Механизм, связывающий вместе данные и код, обрабатывающий эти данные, и сохраняющий их от внешнего воздействия и ошибочного использования.

*Наследование*

Процесс, благодаря которому один объект может наследовать (приобретать) свойства от другого объекта.

*Полиморфизм*

Способность вызывать метод потомка через экземпляр предка

*Абстракция данных*

Абстракция подразумевает разделение и независимое рассмотрение интерфейса и реализации.

**3. Охарактеризуйте открытые методы System.Object.**

Метод ToString служит для получения строкового представления данного объекта.

Метод **GetHashCode** позволяет возвратить некоторое числовое значение, которое будет соответствовать данному объекту или его хэш-код.

Метод GetType позволяет получить тип данного объекта.

Метод Equals позволяет сравнить два объекта на равенство.

Метод Finalize вызывается, когда сборщик мусора определяет, что объект является мусором, но до возвращения занятой объектом памяти в кучу.

Метод Clone создает новый экземпляр типа и присваивает полям нового объекта значения объекта this.

**7.** **В чем отличие между объектом и классом?**

class это статическая часть кода, состоящая из атрибутов, которые не меняются во время выполнения программы - наподобие определений методов класса. **Объект это экземпляр класса**.

**8. Что такое конструктор? Когда вызывается конструктор?**

Конструктор – это метод класса, имеющий имя класса. Конструкторов в классе может быть несколько или ни одного. Конструктор автоматически вызывается на этапе компиляции при создании экземпляра данного класса.

**9. Перечислите свойства конструктора?**

1) не имеет возвращаемого значения

2) имя такое же, как и имя типа (класса)

3) не наследуются

4) нельзя применять модификаторы virtual, new, override, sealed и abstract

5) для класса без явно заданных конструкторов компилятор создает конструктор по умолчанию (без параметров)

6) для статических классов (seald и abstract) компилятор не создает конструктор по умолчанию

7) может определяться несколько конструкторов, сигнатуры и уровни доступа к конструкторам обязательно должны отличаться

**10. Что такое деструктор (destructor)?**

Метод, который вызывается для гарантированного освобождения памяти под объект.

**11. Что такое this?**

Ключевое слово, которое обеспечивает доступ к текущему экземпляру класса

**13. Какие спецификаторы доступа для класса и методов класса существуют в C#?**

Private, public, protected, internal

**14. Опишите модификатор protected internal.**

protected – член класса доступен только внутри данного класса и внутри классов, производных от данного. internal – член класса доступен только внутри данной сборки (программы).

**15. Зачем и как используются ref и out параметры функции?**

При передаче параметров по ссылке перед параметрами используется модификатор **ref.** Чтобы сделать параметр выходным, перед ним ставится модификатор out.

**16. Приведите пример необязательных и именованных параметров метода.**

**Необязательные аргументы** позволяют определить используемое по умолчанию значение для параметра метода. **Именованные аргументы:** значение аргумента присваивается параметру по его позиции в списке аргументов.

**17. Приведите пример полей класса – статические, константные, только для чтения.**

**18. Приведите пример определения свойств класса. Как свойства связаны с инкапсуляцией?**

Они обеспечивают простой доступ к полям классов и структур, узнать их значение или выполнить их установку. Свойства в С# состоят из объявления поля и методов-аксессоров для работы с этим полем.

**19. Назовите явное имя параметра, передаваемого в метод set свойства класса?**

value

**20. Что такое автоматические свойства?**

Свойства управляют доступом к полям класса. Однако что, если у нас с десяток и более полей, то определять каждое поле и писать для него однотипное свойство было бы утомительно.

**21. Что такое индексаторы класса? Какие ограничения существуют на индексатор?**

Индексаторы позволяют приложению обращаться с объектом класса так, как будто он является массивом. Индексатор во многом напоминает свойство, но в отличие от свойства он принимает в качестве параметра индекс массива.

Ограничения на индексаторы:

1) значение, выдаваемое индексатором, нельзя передавать методу в качестве параметра ref или out

2) индексатор не может быть объявлен как static

**22. Что такое перегруженный метод?**

Иногда возникает необходимость создать один и тот же метод, но с разным набором параметров. И в зависимости от имеющихся параметров применять определенную версию метода. Такая возможность еще называется **перегрузкой методов**

**23. Что такое partial класс и какие его преимущества?**

Классы могут быть частичными. То есть мы можем иметь несколько файлов с определением одного и того же класса, и при компиляции все эти определения будут скомпилированы в одно.

Правила использования частичных методов:

► внутри частичного класса или структуры

► должны всегда иметь возвращаемый тип void

► не могут иметь параметров out

► может иметь параметры ref, универсальные параметры, экземплярные или статические, unsafe

► private не пишется (закрыт

**24. Что такое анонимный тип в C#?**

Позволяют создать объект с некоторым набором свойств без определения класса (тип в одном контексте или один раз).

**25. Для чего делают статические классы?**

Преимущество применения статических классов заключается в том, что компилятор может проверить отсутствие случайно добавленных членов экземпляров. Таким образом, компилятор гарантирует невозможность создания экземпляров таких классов.

**26. В чем отличие статического поля от экземплярного?**

Статический метод - вызывается на уровне типа (вызов: Имя\_типа точка Имя\_метода)  
Экземплярный метод - вызывается на уровне объявленного экземпляра типа (вызов: Имя\_переменной точка Имя\_метода).

**27. Поясните работу статических конструкторов.**

**28. Какая разница между поверхностным (shallow) и глубоким (deep) копированием?**

**29. В чем разница между равенством и тождеством объектов?**

**30. Что такое частичные классы и частичные методы?**

Классы могут быть частичными. То есть мы можем иметь несколько файлов с определением одного и того же класса, и при компиляции все эти определения будут скомпилированы в одно.

Частичные классы могут содержать частичные методы. Такие методы также опреляются с ключевым словом **partial**. Причем определение частичного метода без тела метода находится в одном частичном классе, а реализация этого же метода - в другом частичном классе.