**Вопросы.**

1. **Что такое ООП? Какие основные принципы ООП?**

**Объектно-ориентированное программирование** - это методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования

Принципы объектно-ориентированного программирования:

**Инкапсуляция** – механизм сохраняющий данные и код, от внешнего воздействия и ошибочного использования.

**Наследование** – процесс благодаря которому, один объект может наследовать свойства другого, позволяющий описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью.

**Полиморфизм** – способность вызывать метод наследника через объект предка.

**Абстракция** —  означает выделение значимой информации и исключение из рассмотрения незначимой. Это разделение программы на объекты. Абстракция позволяет отобрать главные характеристики и опустить второстепенные.

1. **Какой класс .NET является базовым для всех остальных? Охарактеризуйте его открытые методы.**
2. **Что такое конструктор класса? Какие виды конструкторов существуют в .NET?**

Конструкторы — это специальные методы, позволяющие корректно инициализировать новый экземпляр типа. Создание экземпляра объекта ссылочного типа

1) выделяется память для полей данных экземпляра

2) инициализируются служебные поля

3) вызывается конструктор экземпляра, устанавливающий исходное состояние нового объекта память всегда обнуляется до вызова конструктора

экземпляра типа. Любые поля, не задаваемые конструктором явно, гарантированно содержат 0 или null.

1. **В чем особенность статического конструктора?**

Конструктор экземпляра инициализирует данные экземпляра.

Статический конструктор нельзя вызывать напрямую. Если обычный конструктор вызывается в момент создания экземпляра класса, то статический – вызывается автоматически перед созданием первого экземпляра класса или перед использованием ссылки на какие-либо статические члены.

1. **Что такое деструктор? Для чего он используется? В какой момент вызывается деструктор?**
2. **Что такое «свойство класса»? В чем заключается отличие свойства от поля?**

**7. Перечислите виды свойств в .NET.**

**8. Что обозначает ключевое слово readonly?**

Поля для чтения можно инициализировать при их объявлении либо на уровне класса, либо инициализировать и изменять в конструкторе. Инициализировать или изменять их значение в других местах нельзя, можно только считывать их значение.

Поле для чтения объявляется с ключевым словом **readonly**.

**9. Что обозначает ключевое слово static? В чем заключается особенность статического класса?**

**10. В чем отличие между public static readonly и public const членами класса?**

**11. Что такое модификаторы доступа? Какие модификаторы доступа вы знаете? Охарактеризуйте каждый из них.**

**12. Какие модификаторы доступа являются дефолтными (по умолчанию) для полей класса? Методов класса? Классов?**

**13. Что такое ref/out параметры? В чем их отличие?**

**14. Что такое in параметры?**

**15. Что обозначает ключевое слово partial? Что мы можем пометить как partial в .NET? Могут ли быть partial методы? Если да, то какие существуют ограничения на подобные методы?**

**16. Что такое метод расширения? Какие ограничения существуют для создания метода расширения? В каких случаях используются методы расширения?**

**17. За счет какого механизма происходит переопределение метода? Что обозначают ключевые слова virtual и override?**

**18. Что такое индексатор? Как определяется индексатор?**

**19. Что такое анонимные типы?**

Анонимные типы позволяют создать объект с некоторым набором свойств без определения класса. Анонимный тип определяется с помощью ключевого слова **var** и инициализатора объектов.

В данном случае user - это объект анонимного типа, у которого определены два свойства Name и Age. И мы также можем использовать его свойства, как и у обычных объектов классов. Однако тут есть ограничение - свойства анонимных типов доступны только для чтения.

При этом во время компиляции компилятор сам будет создавать для него имя типа и использовать это имя при обращении к объекту. Нередко анонимные типы имеют имя наподобие "<>f\_\_AnonymousType0'2".

Для исполняющей среды CLR анонимные типы будут также, как и классы, представлять ссылочный тип.

**20. В чем разница между глубоким (deep) и поверхностным (shallow) копированием?**

**21. Что обозначает ключевое слово sealed?**

**Повышенный уровень.**

22. Что такое паттерн «Одиночка»? Для чего он применяется? Какие главные характерные черты класса, реализующего данный паттерн?

23. Что такое паттерн **Dispose**? Для чего он применяется?

24. Для чего служит метод *Finalize*? Как он связан с деструктором класса?

25. Можем ли мы переопределить метод *Dispose*? *Finalize*