1. Назовите принципы ООП. Поясните каждый из них.

**Абстракция**: Абстракция позволяет выделить сущности и концепции, которые являются важными для решения задачи, и скрыть детали их внутренней реализации. Это означает, что вы можете сосредоточиться на необходимых характеристиках и поведении объектов, не вдаваясь в детали их реализации. Абстракция помогает создавать более понятный и гибкий код.

**Наследование**: Наследование позволяет создавать новые классы на основе существующих классов, позволяя наследникам наследовать свойства и методы родительского класса. Это позволяет повторно использовать код, улучшает расширяемость и обеспечивает иерархию классов с общими характеристиками.

**Инкапсуляция**: Инкапсуляция означает объединение данных и методов, работающих с этими данными, в единый объект. При инкапсуляции данные скрыты от прямого доступа извне, и доступ к ним осуществляется через методы объекта. Это обеспечивает контроль над доступом к данным и обеспечивает безопасность и целостность объекта.

**Полиморфизм**: Полиморфизм позволяет объектам одного типа проявлять различное поведение в зависимости от контекста. Это означает, что один и тот же метод может иметь различную реализацию в разных классах. Полиморфизм позволяет программисту работать с объектами разных классов с использованием общего интерфейса, что упрощает разработку гибкого и расширяемого кода.

**Композиция**: Композиция представляет собой способ создания более сложных объектов путем объединения простых или других объектов. Вместо того, чтобы наследовать поведение, как в случае с наследованием, объекты создаются путем объединения других объектов внутри себя в качестве частей. Это позволяет создавать более гибкие и модульные системы, где изменение одной части не влияет на остальные.

1. Назовите класс .NET, от которого наследуются все классы.

В языке программирования C# и платформе .NET Framework класс, от которого наследуются все классы, называется System.Object или просто Object. Класс Object является базовым классом для всех типов данных в .NET и предоставляет основные функциональные возможности, такие как методы сравнения, хеширования, приведения типов и т.д. Все классы в C# неявно наследуются от класса Object, если явно не указан другой базовый класс.

1. Охарактеризуйте открытые методы System.Object.

Equals(Object) Определяет, равен ли указанный объект текущему объекту. Equals(Object, Object) Определяет, считаются ли равными указанные экземпляры объектов.

GetHashCode() Служит хэш-функцией по умолчанию.

GetType() Возвращает объект Type для текущего экземпляра.

ReferenceEquals(Object, Object) Определяет, совпадают ли указанные экземпляры Object.

ToString() Возвращает строку, представляющую текущий объект.

1. Охарактеризуйте закрытые методы System.Object.

Finalize() Позволяет объекту попытаться освободить ресурсы и выполнить другие операции очистки, перед тем как он будет уничтожен во время сборки мусора.

MemberwiseClone() Создает неполную копию текущего объекта Object.

1. Приведите пример определения класса.

public class MyClass

{

// Поля (переменные) класса

private int myInt;

public string myString;

// Конструктор класса

public MyClass(int intValue, string stringValue)

{

myInt = intValue;

myString = stringValue;

}

// Методы класса

public void PrintValues()

{

Console.WriteLine("myInt: " + myInt);

Console.WriteLine("myString: " + myString);

}

}

1. Какие ключевые слова можно использовать при определении класса?

class: Ключевое слово class используется для определения нового класса. Пример: public class MyClass { }.

public, private, protected, internal, file: Эти ключевые слова определяют уровень доступа к классу и его членам. public делает класс и его члены доступными из любого места в программе. private делает класс и его члены доступными только внутри самого класса. protected делает класс и его члены доступными внутри самого класса и его производных классов. internal делает класс и его члены доступными только в пределах текущей сборки (assembly).

static: Ключевое слово static используется для определения статических членов класса, которые принадлежат самому классу, а не его экземплярам. Пример: public static int myStaticVariable.

abstract: Ключевое слово abstract указывает на абстрактный класс. Абстрактные классы не могут быть инстанциированы, а только использованы в качестве базового класса для других классов. Пример: public abstract class MyAbstractClass { }.

sealed: Ключевое слово sealed указывает на запечатанный класс. Запечатанные классы не могут быть производными классами. Пример: public sealed class MySealedClass { }.

partial: Ключевое слово partial позволяет определить класс в нескольких файлах. Каждый файл содержит часть определения класса с ключевым словом partial. Все эти части объединяются в один класс при компиляции. Пример: public partial class MyPartialClass { }.

1. В чем отличие между объектом и классом?

Класс - это шаблон, описывающий структуру и поведение объектов.

Объект - это конкретный экземпляр класса, созданный на основе его определения (класса).

1. Что такое конструктор? Когда вызывается конструктор?

Конструктор - это специальный метод класса, который используется для инициализации объектов при их создании.

Вызов конструктора происходит при создании нового объекта.

1. Перечислите свойства конструктора?

Имя, модификаторы доступа, параметры

1. Что такое деструктор (destructor) ?

Метод класса для уничтожения класса

1. Что такое this?

ссылаться на текущий экземпляр (объект) класса, в котором оно используется.

12. Что будет выведено в результате выполнения

class A

{

private int \_num;

public A(int num) { Num = num; }

public int Num { get { return \_num; } set { \_num = value; } }

}

static void Main(string[] args)

{

A a = new A(5);

A b = a;

Console.WriteLine(a.Num + " " +b.Num);

a.Num = 7;

Console.WriteLine(a.Num + " " + b.Num);

}

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

public: Класс доступен из любого места в программе.

internal: Класс доступен только внутри сборки (assembly), в котором он определен. Это является значением по умолчанию для классов, если не указан явно.

protected: Класс доступен только внутри сборки и в производных классах.

protected internal: Класс доступен внутри сборки и в производных классах, а также из других сборок через экземпляры производных классов.

private: Класс доступен только внутри того же класса, в котором он определен.

file: доступ только из текущего файла

14. Опишите модификатор protected internal.

Класс доступен внутри сборки и в производных классах, а также из других сборок через экземпляры производных классов.

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

ключевые слова ref и out используются для передачи аргументов в функции по ссылке

ref параметр должен быть инициализирован перед передачей в функцию, в то время как out параметр не требуется быть инициализированным. Вместо этого функция должна установить значение out параметра до возврата из функции.

ref параметр может быть изменен функцией до и после его использования, в то время как out параметр должен быть изменен функцией до возврата из функции.

16. Приведите пример необязательных и именованных параметров метода.

Необязательные параметры это параметры по умолчанию которые если не указать то выберутся те которые мы указали в конструкторе

Именованные параметры мы можем указывать не по порядку на писать что это за параметр

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

Static присваивается классу и будет одинаковым для всех экземпляров

Const не может быть изменено

Readonly определяется только внутри конструктора или класса и не может изменять из вне

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

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

value

20. Что такое автоматические свойства?

позволяет определить свойства класса без явного определения полей и методов доступа

21. Что такое индексаторы класса? Какие ограничения существуют на

индексатор?

Принимает только целочисленные значения, не должен выходить за размер, в одном классе могут быть несколько индексаторов

22. Что такое перегруженный метод?

Перегруженный метод - это метод с тем же именем, но с различными параметрами в классе.

23. Что такое partial класс и какие его преимущества?

Partial класс в C# - это класс, который может быть разделен на несколько файлов. Каждый файл содержит часть определения класса с помощью ключевого слова partial. Компилятор объединяет все файлы с частичным классом в один класс при компиляции.

24. Что такое анонимный тип в C#?

Анонимный тип в C# - это тип данных, который создается автоматически компилятором на основе инициализатора объекта.(new) не может быть изменен

25. Для чего делают статические классы?

Статические классы в C# служат для группировки связанных методов и свойств, которые могут быть использованы без необходимости создания экземпляра класса.

26. В чем отличие статического поля от экземплярного?

Статический принадлежит классу а не экземпляру и инициализируется один раз и сохраняет свое значение до конца выполнения программы

Экземплярное поле принадлежит экземпляру класс, объявляется без static, изменяется через экземпляр класса

27. Поясните работу статических конструкторов.

Статический конструктор (static constructor) - это специальный блок кода в классе, который выполняется автоматически один раз перед первым обращением к классу или к любому статическому члену этого класса.

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

копированием?

Для ссылочных типов при поверхностном создаются новые ссылки на существующие объекты, при полном создаются новые объекты

29. В чем разница между равенством и тождеством объектов?

Равенство сравнивает значения, а тождественные являются если ссылки указывают на один и тот же объект

30. Что такое частичные классы и частичные методы?

Частичные классы позволяют разделить определение класса на несколько файлов, объединяющихся в один класс при компиляции. Частичные методы позволяют разделить реализацию метода на объявление и реализацию, при этом реализацию можно определить только в одной из частей класса.

31. Что будет выведено на консоль результате выполнения следующего кода:

System.Int32

32. Что будет выведено на консоль результате выполнения следующего кода:

A=2 b=1

33. Пусть задан следующий класс. Какой из конструкторов задан неверно?

public A(){} так как он имеет такое же имя и параметры как и класс

34. Пусть задан следующий класс. Сколько аргументов может быть задано при вызове конструктора данного класса?

До двух так как параметры указаны по умолчанию

35. Почему не удается создать объект класса A?

Потому что все конструкторы созданы internal

36. Что будет выведено в консоль при выполнении данной программы?

A static

A

37. Какая строка приведенного далее класса вызовет ошибку компиляции?

B = 30 так как readonly