Вопросы

1. Чем класс отличается от структуры?

2. Что может и чего не может быть в структуре?

3. Что такое перечисление? Приведите пример определения и использования

перечисления

4. Перечислите и поясните стандартные интерфейсы .Net?

5. Как используется интерфейс IComparable?

6. Как используется интерфейс ICloneable?

7. Что такое полиморфизм? Перечислите его формы. Приведите примеры.

8. Зачем в классе определяют виртуальные методы?

9. Как сделать запрет переопределения методов?

Ответы:

**1. Класс vs. Структура:**

- Класс: Класс является ссылочным типом и используется для создания объектов. Он может иметь методы, свойства, поля и конструкторы. Экземпляры классов создаются в куче (heap memory).

- Структура: Структура (struct) является значимым типом и используется для создания небольших объектов, которые хранятся в стеке (stack memory). В структуре могут быть поля, свойства, конструкторы, но она не поддерживает наследование.

**2. Содержимое структуры:**

- В структуре могут быть поля (fields), свойства (properties), конструкторы (constructors), методы (methods), а также события (events). Она также может иметь интерфейсы и перечисления, но они могут быть ограничены.

**3. Перечисление (Enum):**

- Перечисление (Enum) в C# - это тип данных, который состоит из набора именованных констант. Он облегчает работу с наборами значений, представляя их в виде читаемых символических имен. Например:

enum DaysOfWeek

{

    Sunday,

    Monday,

    Tuesday,

    Wednesday,

    Thursday,

    Friday,

    Saturday

}

DaysOfWeek today = DaysOfWeek.Wednesday;

**4. Стандартные интерфейсы .NET:**

- .NET предоставляет множество стандартных интерфейсов, таких как `IComparable`, `IList`, `IDisposable`, `IEnumerable` и другие. Эти интерфейсы предоставляют общие методы и функциональность, которые классы могут реализовывать для обеспечения определенного поведения.

**5. Использование интерфейса IComparable:**

- `IComparable` позволяет классам реализовать сравнение с другими объектами того же типа. Этот интерфейс требует реализации метода `CompareTo`, который возвращает отрицательное число, ноль или положительное число в зависимости от того, какой объект "меньше", "равен" или "больше". Это используется, например, для сортировки объектов.

**6. Использование интерфейса ICloneable:**

- `ICloneable` предоставляет метод `Clone`, который позволяет создать копию объекта. Однако этот интерфейс устарел, и в современном коде обычно используется более мощный механизм клонирования, такой как методы `MemberwiseClone` или сериализация.

**7. Полиморфизм:**

- Полиморфизм - это принцип ООП, который позволяет объектам разных классов реагировать на одну и ту же операцию (метод) в специфическом для них контексте. Формы полиморфизма включают:

- Полиморфизм во времени выполнения (ранний и поздний): Возможность вызывать методы подкласса через ссылку на суперкласс.

- Перегрузка методов: Несколько методов с одинаковыми именами в одном классе, но с разными параметрами.

- Интерфейсный полиморфизм: Реализация разных классов одним и тем же интерфейсом, что позволяет вызывать методы интерфейса на объектах разных классов.

**8. Виртуальные методы:**

- Виртуальные методы в классах позволяют подклассам переопределять эти методы и предоставлять свою реализацию. Они используются для реализации полиморфизма и переопределения поведения в производных классах.

**9. Запрет переопределения методов:**

- Для предотвращения переопределения методов в производных классах можно использовать ключ *“****sealed****”*