В C# структура (struct) — это тип данных, похожий на класс, который используется для хранения небольших объектов, обычно содержащих несколько связанных значений. Структуры являются значимыми типами, а не ссылочными типами (как классы), что означает, что при передаче структуры по значению копируется её содержимое. Это делает их более производительными для небольших объектов, таких как точки на плоскости, цвет, размер и т.п.

1. \*\*Чем класс отличается от структуры?\*\*

Класс — это ссылочный тип, тогда как структура — это тип значения. Объекты класса хранятся в управляемой куче и передаются по ссылке, тогда как структура хранится в стеке и передается по значению. Классы поддерживают наследование, а структуры — нет.

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

В структуре могут быть поля, методы, конструкторы, свойства, но не могут быть указаны конструкторы без параметров (они добавляются компилятором по умолчанию), и структура не может наследоваться от других классов или структур.

3. \*\*Что такое перечисление? Приведите пример определения и использования перечисления.\*\*

Перечисление (enum) — это тип, задающий набор именованных констант. Пример:

```csharp

enum Days { Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday }

```

Использование:

```csharp

Days today = Days.Monday;

```

4. \*\*Перечислите и поясните стандартные интерфейсы .Net.\*\*

- `IComparable` — для сравнения объектов.

- `ICloneable` — для клонирования объектов.

- `IDisposable` — для освобождения неуправляемых ресурсов.

- `IEnumerable` — для перебора коллекции.

- `IList` — для создания коллекций с доступом по индексу.

5. \*\*Как используется интерфейс IComparable?\*\*

`IComparable` используется для сравнения объектов. Метод `CompareTo` возвращает 0, если объекты равны, положительное значение, если текущий объект больше, и отрицательное, если меньше.

6. \*\*Как используется интерфейс ICloneable?\*\*

`ICloneable` позволяет создать копию объекта. Метод `Clone` возвращает новый объект, копирующий текущее состояние объекта.

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

Полиморфизм — это возможность объектов разных типов обрабатывать идентичные вызовы методов. Формы:

- \*\*Перегрузка методов\*\*: методы с одинаковыми именами, но разными параметрами.

- \*\*Переопределение методов\*\*: наследуемые методы с новой реализацией.

Пример:

```csharp

public class Animal { public virtual void Speak() { Console.WriteLine("Animal sound"); } }

public class Dog : Animal { public override void Speak() { Console.WriteLine("Bark"); } }

```

8. \*\*Зачем в классе определяют виртуальные методы?\*\*

Виртуальные методы позволяют подклассам переопределять поведение базового класса, обеспечивая возможность создания специализированного поведения в производных классах.

9. \*\*Как сделать запрет переопределения методов?\*\*

Для запрета переопределения метода в производных классах используется модификатор `sealed`.