1. Перечислите свойства внутренних и вложенных классов?

**Внутренние (Inner) классы:**

1. объявлен внутри другого класса.
2. имеет доступ к членам внешнего класса, включая закрытые (private) члены.
3. используются для группировки связанных классов и сокрытия их реализации от внешнего кода.
4. не могут существовать вне экземпляра внешнего класса и могут быть созданы только после создания экземпляра внешнего класса.

**Вложенные (Nested) классы:**

1. объявлен внутри другого класса,
2. не имеет доступа к членам внешнего класса, включая закрытые члены.
3. Вложенные классы полезны, когда вам нужно организовать классы внутри другого класса, но они не зависят от состояния внешнего класса.
4. мб созданы без создания экземпляра внешнего класса.

2. Что такое статический класс и какие у него свойства?

Указывается ключевое слово **static** и содержит только статические члены (поля, методы, свойства и т.п.), **создавать объекты такого класса нельзя**!

1. прямой потомок System.Object
2. экземпляры такого класса создавать запрещено(можно через сам класс)
3. не должен реализовывать никаких интерфейсов (не вызвать)
4. нельзя использовать в качестве поля, параметра метода или локальной переменной
5. от него запрещено наследовать
6. все элементы такого класса должны явным образом объявляться с модификатором static
7. может иметь статический конструктор
8. компилятор не создает автоматически конструктор по умолчанию

3. Каково назначение перегрузки операторов?

Перегрузка операторов - это возможность определять собственное поведение операторов для пользовательских типов данных. Это позволяет разработчикам переопределить стандартное поведение операторов, чтобы они работали с их собственными типами данных так, как это логически ожидается.  
**public static возвр\_тип operator оператор(параметры) {..}**

Свойства:  
1) открытыми и статическими  
2) обязаны возвращать значения

3)префиксная и постфиксная формы операций ++ и --, в отличие от оригинальных операций, семантически не различаются

*class BigInt {  
 public int Value { get; set; }  
 public static BigInt operator +(BigInt a, BigInt b) {  
 return new BigInt  
 {Value = a.Value + b.Value };  
 }  
 public static bool operator >)BigInt a, BigInt b) {  
 if (a.Value > b.Value) return true;  
 else return false;  
 }  
} …  
BigInt x = new BigInt { Value = 50 };  
BigInt y = new BigInt { Value = 105 };  
Console. WriteLine(x>y); //false  
Console.WriteLine((a+b).Value); //155*

Подлежат перегрузке:  
+, -, !, ( ++, -- парами) унарные  
true, false (попарно)  
+, -, \*, /, %, &, |, ^, <<, >> бинарные  
==, !=, <, >, <=, >= (перегрузка парами) сравнения

4. Как используется ключевое слово operator?

Ключевое слово operator используется для перегрузки операторов.

5. Какие операции нельзя перегружать в C#?

[] (но есть индексатор)  
() (можно определить новые операторы преобразования)  
+=, -=, \*=, /=… (получаем автоматически при перегрузке бинарной операции)  
&&, || логические  
=, ., ?:, ??, ->, =>, f(x), as, checked, unchecked, default, delegate, is, new, sizeof, typeof

6. Можно ли перегрузкой отменить очередность выполнения операции?

Нет, перегрузка операторов в C# не позволяет отменить стандартную очередность выполнения операций.

Например, при перегрузке оператора **+** для пользовательского класса, вы можете определить, как будет выполняться сложение объектов этого класса, но порядок выполнения сложения (слева направо) останется неизменным, так как это стандартное поведение для оператора **+**.

Приоритет операций в общем случае не имеет никакого отношения к последовательности выполнения операторов. Но именно для перегруженных операторов такая связь имеется - перегруженные операторы выполняются именно в порядке их приоритета

7. Истинно ли след. утверждение: операция >= может быть перегружена

да

8. Сколько аргументов требуется для определения перегруженной унарной операции?

1 – объект класса, в которой данный оператор определен. Унарные операции выполняются над одним операндом, поэтому метод, перегружающий унарный оператор, должен принимать ровно один параметр, представляющий операнд, над которым выполняется унарная операция.

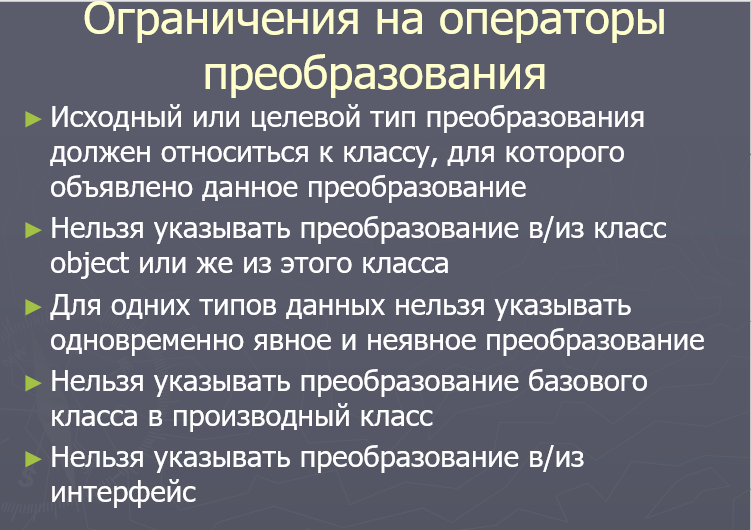
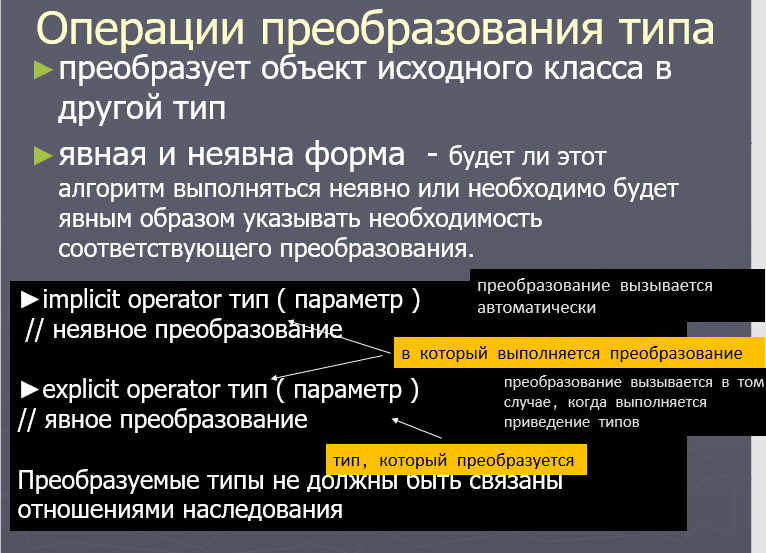
9. Можно ли перегружать операцию []?

нет, но есть индексатор

10. Можно ли перегружать операцию ->?

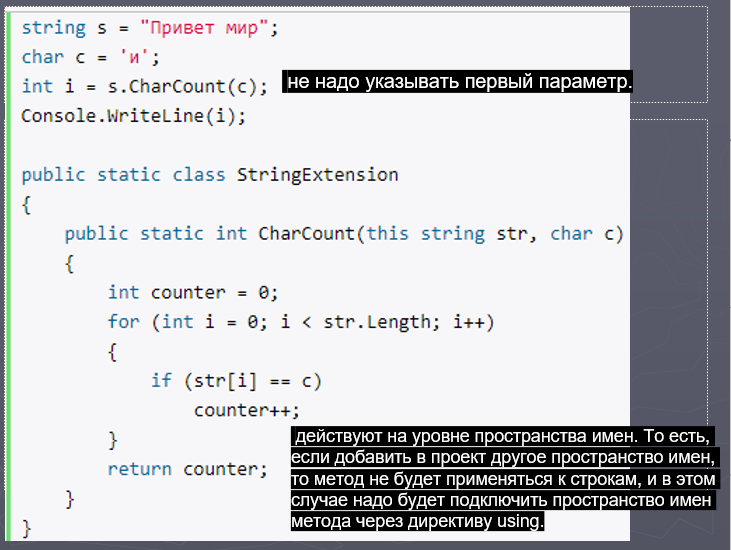
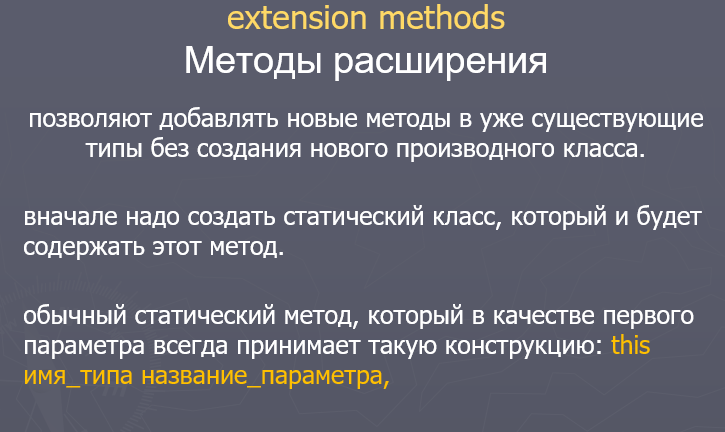
Нет. Оператор -> в языке C# используется исключительно для доступа к членам управляемых указателей (^ - управляемый указатель). Он предоставляет доступ к членам объекта, на который указывает управляемый указатель, и это поведение не может быть изменено или перегружено.

11. Приведите пример оператора приведения типа



12. Что такое метод расширения? Как и где его можно использовать?

Метод расширения - это статический метод, который позволяет добавлять новые методы в уже существующие типы без создания нового производного класса.



14. Верно? Метод расширения может:  
1) получать доступ к public членам расширяемого класса  
2) получать доступ к protected членам расширяемого класса  
3) получать доступ к internal членам расширяемого класса  
4) быть объявлен в любом классе  
5) быть без параметров

15. Неверное правило перегрузки операторов для C#:  
1) префиксные операции ++ и – перегружаются парами  
2) операции сравнения перегружаются парами: == и !=; < и >; <= и >=  
3) перегруженные операции обязаны возвращать значения

4) должны объявляться как ~~protected~~ **public и static**  
5) true и false можно перегружать

13.Пусть дан фрагмент кода определения оператора преобразования типа. Определить форму преобразования. public static implicit operator Point2D(Point3D a) {/\* код\*/;}

В данном фрагменте кода определен оператор неявного (implicit) преобразования типа из типа Point3D в тип Point2D. Это означает, что преобразование выполняется неявно, без явного использования оператора приведения, если оно безопасно (не вызывает потери данных или ошибок).

