1. Перечислите свойства внутренних и вложенных классов?

internal - внутренний

internal: компоненты класса или структуры доступен из любого места кода в той же сборке, однако он недоступен для других программ и сборок.

В языке C# любой класс в своей реализации может содержать объявление другого класса. Класс, который объявляется в пределах фигурных скобок другого класса, называется вложенным классом.

Вложенные классы имеют доступ ко всем членам, видимым содержащему их классу, даже если эти члены являются приватными.

Вложенные классы, объявленные как public, позволяют создавать экземпляры коду, внешнему по отношению к охватывающему классу.

Объект вложенного класса можно объявить в случае, когда вложенный класс объявлен как видимый. Это значит, что видимый вложенный класс должен быть объявлен с любым модификатором доступа кроме private.

*Вложенный класс должна иметь public*

*Вложенная структура это структура определенная внутри класса.*

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

Класс считается статическим, если при его создании, перед ключевым словом **class** указывается ключевое слово **static**. Определение класса статическим, означает то, что этот класс является набором только статических сущностей (полей, методов, свойств и т.п.), и, **внимание**, создавать объекты такого класс нельзя!



►прямой потомок System.Object

►экземпляры такого класса создавать запрещено

►не должен реализовывать никаких интерфейсов (не вызвать)

►нельзя использовать в качестве поля, параметра метода или локальной переменной

►от него запрещено наследовать

►все элементы такого класса должны явным образом объявляться с модификатором static

►может иметь статический конструктор

►Компилятор не создает автоматически конструктор по умолчанию

3. Каково назначение перегрузки операторов?

Перегрузка операций – способ объявления новых операций для типа.  
**public static возвр\_тип operator оператор(параметры) {..}**

**Перегрузка операторов позволяет добавлять удобный синтаксис для работы с классом в среду программирования. Это возможность одна из сильных сторон объектно-ориентированного языка программирования С#.**

**Когда в классе присутствует перегруженный оператор, компилятор сначала выполняет поведение определенное пользователем, если не определено, то осуществляется стандартная реализация.**

Перегруженные операторные методы дб:  
1) открытыми и статическими  
2) тип одного из парам. или возвр. значения совпадал с типом, в котором   
определен операторный метод

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





4. Как используется ключевое слово operator?

**public static возвр\_тип operator оператор(параметры) {..}**

**Ключевое слово operator добавляется, чтобы сообщить компилятору, что следующий символ двоичного оператора является оператором, а не обычным методом. Затем этот оператор обработает два параметра операнда, каждый из которых имеет свой тип данных (op-type и op-type2). По крайней мере один из этих операндов должен быть того же типа, что и содержащий его класс.**

5. Какие операции нельзя перегружать в C#?

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



6. Можно ли перегрузкой отменить очередность выполнения операции?

Приоритет операций диктуется грамматикой языка и наследуется из грамматики языка. Грамматике языка не интересно, встроенные ли это операторы или перегруженные.

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

9. Можно ли перегружать операцию []?

нет, но есть индексатор

10. Можно ли перегружать операцию ->?

нет

11. Приведите пример оператора приведения типа

преобразует объект исходного класса в другой тип

*class Point3D {  
 public int x, y;  
 //НЕЯВНОЕ преобр. 2D -> 3D:  
 public static* ***implicit*** *operator Point3D (Point2D p2d) {  
 Point3D p3d = new Point3D();  
 p3d.x = p2d.x;  
 p3d.y = p2d.y;  
 return p3d;  
 }  
….. p2d = (Point2D)p3d;*

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

позвол. добавлять новые методы в уже существующие типы   
без создания нового производного класса

1) создать статич. класс  
2) в него статич. метод

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