**Ответы на вопросы к лабораторной работе №4.**

**1.Каково назначение перегрузки операторов?**

Чтобы определить выбранный оператор, задав ему нужное вам поведение, нужно его перегрузить. При этом используется синтаксис, похожий на синтаксис метода, с возвращаемым значением и параметрами, но в качестве имени метода используются ключевое слово operator и символ объявляемого оператора.

struct Hour

{ ... public static implicit operator int (Hour from)

{

return from.value;

}

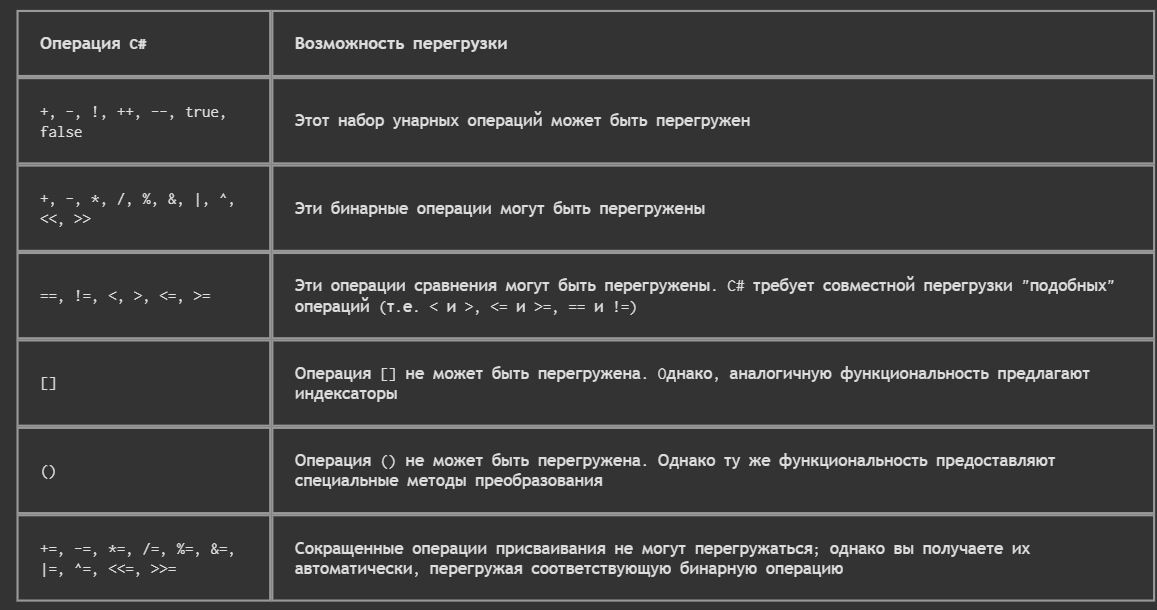
private int value; }

Оператор преобразования должен быть открытым (public), а также статическим (static). Тип, из которого осуществляется преобразование, объявляется в качестве параметра (в данном случае это Hour), а тип, в который осуществляется преобразование, объявляется в качестве названия типа после ключевого слова operator (в данном случае это int). Возвращаемого значения, указываемого перед ключевым словом operator, здесь нет.

**2.Как используется ключевое слово operator?**

Перегрузка операторов тесно связана с перегрузкой методов. Для перегрузки оператора служит ключевое слово operator, определяющее операторный метод, который, в свою очередь, определяет действие оператора относительно своего класса. Существуют две формы операторных методов (operator): одна - для унарных операторов, другая - для бинарных.

**3. Какие операции нельзя перегружать в C#?**



**4. Можно ли перегрузкой отменить очередность выполнения операции?**

Можно.

**5. Истинно ли следующее утверждение: операция >= может быть перегружена.**

Истинно, но должна быть также перегружена совместная операция <=.

**6. Сколько аргументов требуется для определения перегруженной унарной операции?**

Один аргумент. Для бинарной – два.

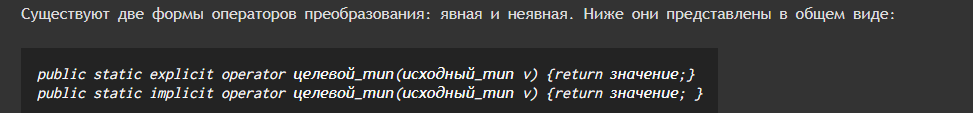
**7. Можно ли перегружать операцию []?**

Нет. Однако аналогичную функциональность предлагают индексаторы.

**8. Можно ли перегружать операцию ->?**

Нет.

**9. Приведите пример оператора приведения типа.**



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

Методы расширения (extension methods) позволяют добавлять новые методы в уже существующие типы без создания нового производного класса. Эта функциональность бывает особенно полезна, когда нам хочется добавить в некоторый тип новый метод, но сам тип (класс или структуру) мы изменить не можем, поскольку у нас нет доступа к исходному коду. Либо если мы не можем использовать стандартный механизм наследования, например, если классы определенны с модификатором sealed.

**11. Пусть дан фрагмент кода определения оператора преобразования типа.**

Определить форму преобразования.

public static implicit operator Point2D(Point3D a) {/\* код\*/;}

Ответ: здесь используется неявная форма преобразования.

**12.Выберите верное утверждение.**

**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**

Ответ:1,3,4.

**13.Выберите из списка 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**

**5) true и false можно перегружать**

Ответ:4.