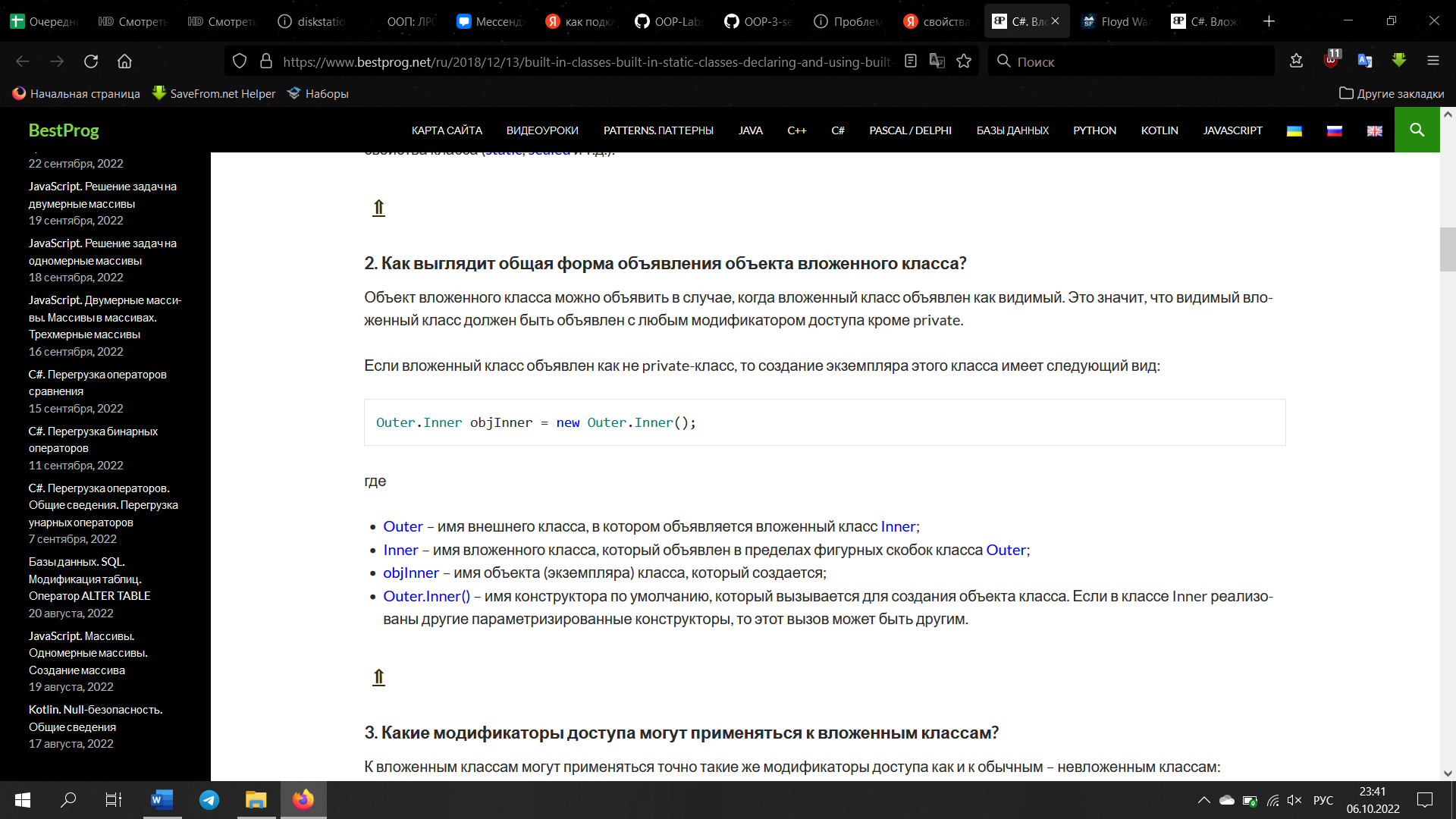
**Вопросы к лабораторной работе №4:**

1.Перечислите свойства внутренних и вложенных классов?

<https://www.bestprog.net/ru/2018/12/13/built-in-classes-built-in-static-classes-declaring-and-using-built-in-classes-examples-ru/>



► класс может быть открытым (public) или **внутренним (internal)**

**модификаторы доступа могут применяться к вложенным классам?**

* private. В этом случае вложенный класс есть видимым в пределах фигурных скобок { } внешнего класса;
* public. В этом случае вложенный класс доступен в пределах внешнего класса, из экземпляра внешнего класса, из унаследованного класса а также за пределами сборки;
* protected. В этом случае класс есть доступен в пределах внешнего класса а также из методов унаследованных классов;
* internal. Вложенный класс есть доступен в пределах сборки и недоступен за пределами сборки;
* protected internal. Вложенный класс может быть доступен из методов внешнего класса, методов унаследованных классов или недоступен из методов за пределами текущей сборки.

Вложенный (внутренний) тип может получить доступ к вмещающему (внешнему) типу. Чтобы получить доступ к внешнему типу, передайте его в качестве аргумента в конструктор вложенного типа.

Вложенный тип имеет доступ ко всем членам, которые доступны внешнему типу. Он может получать доступ к закрытым и защищенным членам вмещающего типа, включая любые наследуемые защищенные члены.

Вложенные типы класса могут иметь модификаторы *public, protected, internal, protected internal, private или private protected.*

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!**

Особенности:

►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

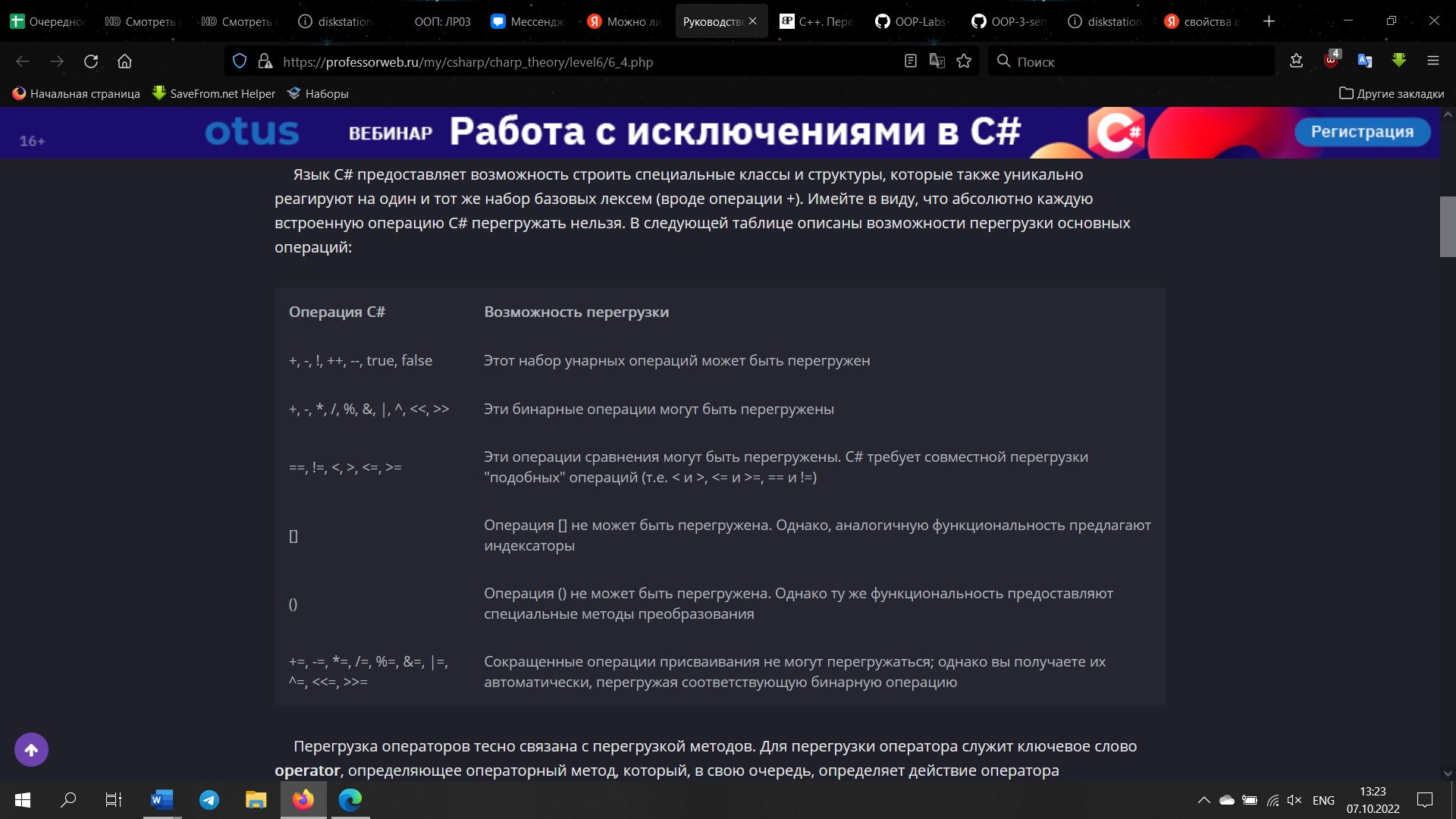
!!! public static возвращающий\_тип operator оператор(параметры) {..}

Перегруженные операторные методы должны быть:

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)

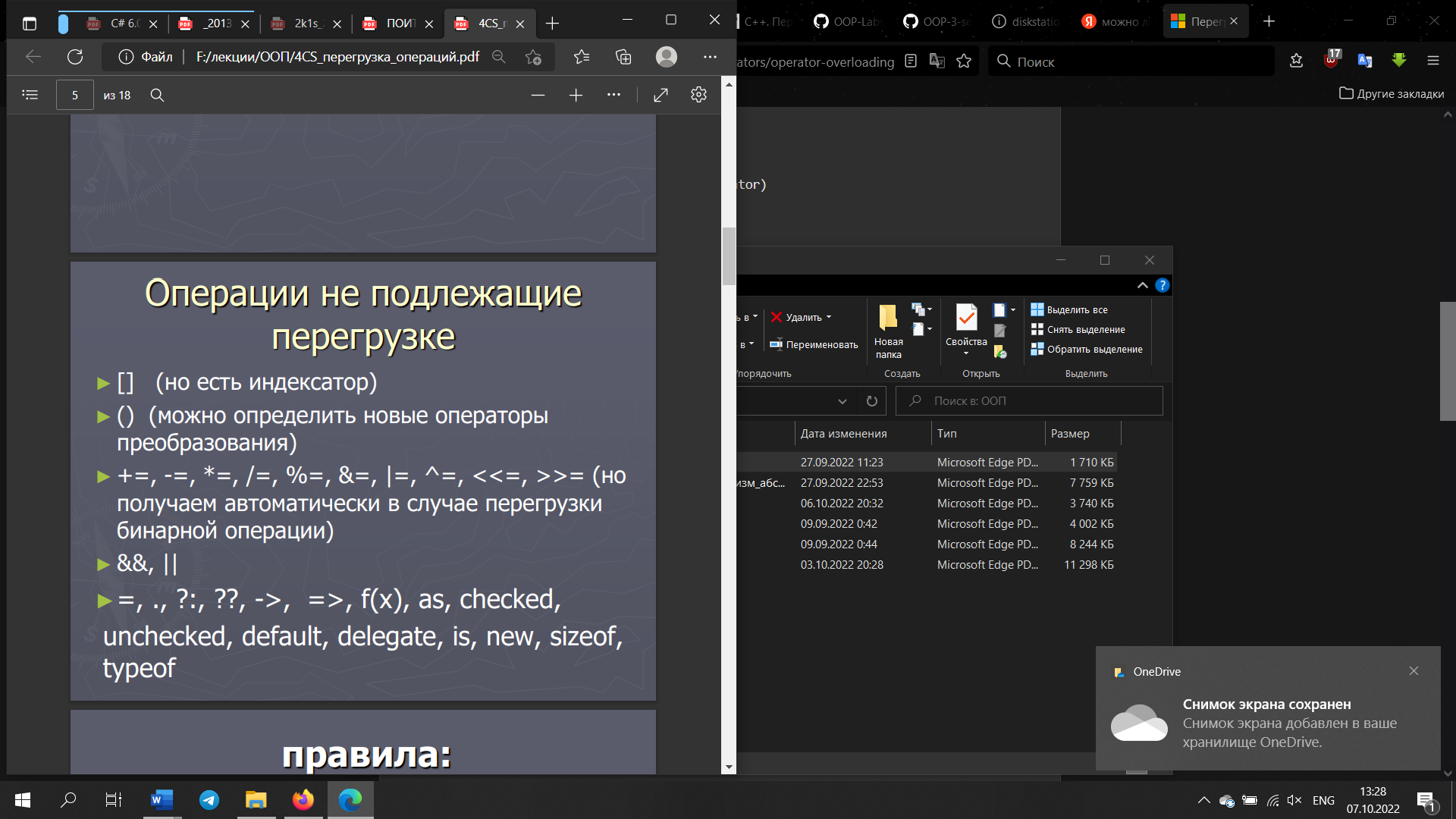


4. Как используется ключевое слово operator?

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.

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

5. Какие операции нельзя перегружать в C#?



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 следующее 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.

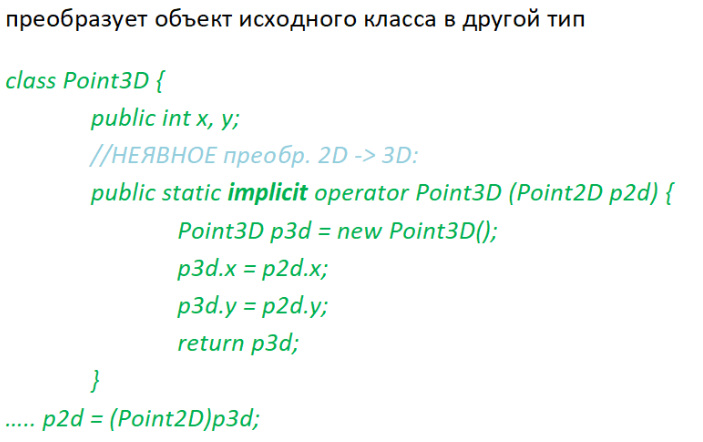
11.Приведите пример оператора приведения типа

public static implicit operator int(Array array) // неявное преобразование

{

return array.Length;

}



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

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

1) создать статический класс

2) в него статический метод

13.Пусть дан фрагмент кода определения оператора преобразования типа.

Определить форму преобразования.

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

*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;*

14.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

15.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~~ public и static

5) true и false можно перегружать