Ответы на 4 лабу ООП

1. Перечислите свойства внутренних и вложенных классов?

**internal**: класс и члены класса с подобным модификатором доступны из любого места кода в той же сборке, однако он недоступен для других программ и сборок (как в случае с модификатором public).

Тип, определенный внутри [класса](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/class), [структуры](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/builtin-types/struct) или [интерфейса](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/interface), называется вложенным типом. Пример

C#

public class Container

{

class Nested

{

Nested() { }

}

}

Независимо от того, является ли внешний тип классом, интерфейсом или структурой, вложенным типам по умолчанию присваивается модификатор [private](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/private), из-за чего они доступны только из содержащего их типа. В предыдущем примере класс Nested недоступен для внешних типов.

Также можно указать [модификатор доступа](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/access-modifiers), определяющий доступность вложенного типа, как показано ниже:

* Вложенные типы **класса** могут иметь модификаторы [public](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/public), [protected](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/protected), [internal](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/internal), [protected internal](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/protected-internal), [private](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/private) или [private protected](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/private-protected).

Тем не менее при определении вложенного класса protected, protected internal или private protected внутри [запечатанного класса](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/sealed) возникает предупреждение компилятора [CS0628](https://docs.microsoft.com/ru-ru/dotnet/csharp/misc/cs0628) "Новый защищенный член объявлен в запечатанном классе".

* Вложенные типы **структуры** могут иметь модификаторы [public](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/public), [internal](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/internal) или [private](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/private).

В следующем примере класс Nested определяется как открытый:

C#Копировать

public class Container

{

public class Nested

{

Nested() { }

}

}

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

C#Копировать

public class Container

{

public class Nested

{

private Container parent;

public Nested()

{

}

public Nested(Container parent)

{

this.parent = parent;

}

}

}

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

В предыдущем объявлении полным именем класса Nested является Container.Nested. Это имя используется для создания нового экземпляра вложенного класса, как показано ниже:

C#

Container.Nested nest = new Container.Nested();

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

Как и нестатический класс, но нельзя создавать экземпляры статического класса. Нельзя использовать оператор new для создания переменной типа класса. Доступ к членам статического класса осуществляется с использованием самого имени класса

3. Каково назначение перегрузки операторов?

Нужно для удобства программиста .

4. Как используется ключевое слово operator?

Ключевое слово operator используется для перегрузки операторов

5. Какие операции нельзя перегружать в C#?

[], ->, (), += -= \*= /= и тд.

6. Можно ли перегрузкой отменить очередность выполнения операции?

Нет

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

Да

8. Сколько аргументов требуется для определения перегруженной унарной операции?

1

9. Можно ли перегружать операцию []?

Индексатор (нет)

Доступ к элементам не считается перегружаемым оператором, но вы можете определить [индексатор](https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/indexers/).

10. Можно ли перегружать операцию ->?

нет

11. Приведите пример оператора приведения типа

*public static MyArr operator -(MyArr obj1, MyArr obj2)*

*{*

*MyArr arr = new MyArr();*

*arr.x = obj1.x - obj2.x;*

*arr.y = obj1.y - obj2.y;*

*arr.z = obj1.z - obj2.z;*

*return arr;*

*}*

12. 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.

13. Пусть дан фрагмент кода определения оператора преобразования типа. Определить форму преобразования.

public static implicit operator Point2D(Point3D a)

{/\* код\*/;}

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

5) true и false можно перегружать