

Как работает ограничение where T:new()?

where T:new() – механизм ограничения, используемый при работе с обобщениями. Обобщения используются для создания классов или методов, работающих с разными типами данных. То есть метод или класс определяется один раз, после чего реализованные механизмы могут быть применены к различным типам данных.

Ограничения накладываются при создании обобщений при их объявлении:

```
class Example<T> where T: (...) {  
    ...  
}
```

Так как в C# реализованы несколько механизмов ограничений, в круглых скобках еще должен быть указан конкретный тип ограничения.

Рассмотрим подробнее ограничение where T: new().

При задании этого ограничения тип данных T должен иметь public конструктор без параметров. Тогда в обобщенном классе можно реализовать, например, следующее:

```
class Program  
{  
    class A1 {  
        public int a { get; set; } = 3;  
        public string str { get; set; } = "test";  
        public A1() { }  
        public override string ToString()  
        {  
            return a.ToString() + " " + str;  
        }  
    }  
  
    class Example<T> where T : new() {  
        public T b = new T();  
        public T output() {  
            return new T();  
        }  
    }  
}
```

```

static void Main(string[] args)
{
    Example<A1> ob2 = new Example<A1>();
    Console.WriteLine( ob2.output().ToString() );
    Console.WriteLine(ob2.b.ToString());
}
}

```

В результате будет выведено:

3 test

3 test

Если нарушить одно из условий ограничения, например изменить конструктор класса A1 на

```

public A1(int param) {
    a = param;
}

```

то возникнет ошибка при создании объекта класса Example.

Может быть задано сразу несколько ограничений. Тогда new() указывается последним. Так, комбинация where T: class, new() потребует использование ссылочного типа данных с открытым конструктором без параметров. В этом случае нельзя использовать в качестве T типы int, double, float или structure.