

C# Essential

Универсальные шаблоны. Ограничения

C# Essential

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал
на [TestProvider.com](http://testprovider.com)

Универсальные шаблоны. Ограничения

Ограничение

where

Предложение **where** используется в определении универсального типа для указания ограничений типов, которые могут использоваться в качестве аргументов параметра типа, определенного в универсальном объявлении.

```
class MyClass<T> where T : struct
{
    //...
}
```

```
static void Main()
{
    MyClass<int> instance = new MyClass<int>();
    //MyClass<string> instance2 = new MyClass2<string>();
}
```

Ограничения определяются с помощью контекстно-зависимого ключевого слова **where**.

Ограничение

`new`

Ограничение `new()` указывает, что аргумент любого типа в объявлении общего класса должен иметь открытый конструктор без параметров.

```
class MyClass<T> where T : new()  
{  
    public T instance = new T();  
}
```



Использовать ограничение `new()` можно только в том случае, если тип не является абстрактным.

Ограничение

Правила использования

`where T: struct` – аргумент типа должен быть структурного типа, кроме `Nullable`.

`where T : class` – аргумент типа должен иметь ссылочный тип; это также распространяется на тип любого класса, интерфейса, делегата или массива.

`where T : <base class name>` – аргумент типа должен являться или быть производным от указанного базового класса.

`where T : U` – аргумент типа, поставляемый для T, должен являться или быть производным от аргумента, поставляемого для U. Это называется неприкрытым ограничением типа.

Коллекции

ArrayList

ArrayList – коллекция с динамическим увеличением размера до нужного значения.

```
static void Main()
{
    ArrayList arrayList = new ArrayList();

    arrayList.Add(1);
    arrayList.Add((object)2);
}
```

При добавлении элементов в коллекцию **ArrayList** ее емкость автоматически увеличивается нужным образом за счет перераспределения внутреннего массива.



Коллекция **ArrayList** – использует **boxing/unboxing**, по этому не рекомендуется ее использовать в больших коллекциях.

Коллекции

Dictionary<TKey, TValue>

`Dictionary<TKey, TValue>` – класс представляет коллекцию, которая работает по принципу - ключей и значений.

```
static void Main()
{
    Dictionary<int, string> dictionary = new Dictionary<int, string>();

    dictionary.Add(0, "Ноль");
    dictionary.Add(1, "Один");
    dictionary.Add(2, "Два");
    dictionary.Add(3, "Три");
}
```


C# Essential

Q&A

Информационный видеосервис для разработчиков программного обеспечения

