

احتمالا در بیشتر مقالات (فارسی/انگلیسی) عبارات هایی مثل نمونه‌های زیر را دیده اید :

```
where T:clas
where T:struc
...
```

در این مقاله قصد داریم بپردازیم به «مقید سازی پارامترهای نوع جنریک» و اینکه چه کاربردی دارند و در چه زمانی بهتر است از آن‌ها استفاده کنیم و نحوه استفاده از آنها چگونه است. فرض میکنیم که خواننده‌ی محترم با مفاهیم جنریک آشنایی دارد. در صورتیکه با جنریک‌ها آشنا نیستید ابتدا مروری داشته باشید بر ج [نریک‌ها](#) و بعد این مقاله را مطالعه فرمایید؛ به این دلیل که موضوع مورد بحث بر پایه‌ی جنریک‌ها می‌باشد.

همانطور که مطلع هستید هر عنصری جنریکی را که تعریف میکنید حداقل دارای یک پارامتر نوع هست و در زمان بکارگیری آن جنریک باید نوع آن را مشخص نمایید. برای نمونه مثال زیر را در نظر بگیرید :

```
public class MyCollection<T>
{
    private List<T> collections = new List<T>();
    public void Add(T value)
    {
        collections.Add(value);
    }
}
```

کلاس فوق یک کلاس جنریک است که در هنگام ساخت نمونه‌ای از آن، باید ابتدا data type نوعی را که می‌خواهیم با آن کار کنیم، تعیین کنیم. برای مثال در کد فوق در هنگام ساخت نمونه‌ای از آن، نوع int را برای آن مشخص میکنیم و هر وقت بخواهیم متد Add آن را فراخوانی کنیم، فقط نوعی را قبول خواهد کرد که در ابتدا برای آن تعیین کرده ایم (int):

```
MyCollection<int> myintObj = new MyCollection<int>();
myintObj.Add(12);
myintObj.Add(33);
myintObj.Add(33.3); // ERROR z
```

سؤال: می‌خواهیم فقط نوع‌هایی را بتوان به T نسبت داد که از نوع ارجاعی (reference type) هستن و یا فقط نوع هایی را به T نسبت داد که یک سازنده دارند؛ چگونه؟

ایجاد قیدها یا محدودیت‌ها بر روی پارامترهای جنریک‌ها شامل پنج حالت می‌باشد:

حالت اول : Where T:struct

در این حالت T باید یک ساختار باشد .

حالت دوم : where T:class

T باید یک نوع ارجاعی باشد. اگر در مثال فوق این قید را به آن اضافه کنیم، در هنگام ساخت نمونه‌ای از کلاس فوق، اگر یک نوع value type را به T نسبت دهیم، در هنگام وارد کردن یک نوع value type با خطا مواجه خواهیم شد. مثال:

```
public class MyCollection<T> where T:class
{
    private List<T> collections = new List<T>();
    public void Add(T value)
    {
        collections.Add(value);
    }
}
```

و برای استفاده :

```
MyCollection<int> myintObj = new MyCollection<int>(); // ERROR , int is value type
```

حالت سوم : where T:new()

نوعی که به T نسبت داده می‌شود باید یک سازنده‌ی پیش فرض داشته باشد.

داخل پرانتز : سازنده‌ی پیش فرض: زمانی که شما یک کلاس می‌نویسید اگر آن کلاس دارای هیچ سازنده‌ای نباشد، کامپایلر یک سازنده‌ی بدون پارامتر را به کلاس فوق اضافه می‌کند که کار آن مقدار دهی به فیلدهای کلاس است. در اینجا از مقادیر پیش فرض استفاده می‌شود. مثلاً برای int مقدار صفر و برای string مقدار "" و به همین ترتیب. اگر از مقدار دهی پیش فرض توسط کامپایلر خرسند نیستید، می‌توانید سازنده پیش فرض را تغییر داده و مطابق میل خود فیلدها را مقدار دهی اولیه کنید .

حالت چهارم : where T:NameOfBaseClass

نوعی که به T نسبت داده می‌شود باید از کلاس NameOfBaseClass ارث بری کرده باشد.

حالت پنجم : where T:NameOfInterface

همانند حالت چهارم می‌باشد؛ با این تفاوت: نوعی که به T نسبت داده می‌شود باید واسط NameOfInterface را پیاده سازی کرده باشد.

پنج حالت فوق نمونه‌هایی از ایجاد محدودیت بر روی پارامتر نوع اعضای جنریک بودند و اما در ادامه قصد داریم نکاتی را در این باب، بیان کنیم:

نکته اول : می‌توانید محدودیت‌های فوق را با هم ترکیب کنید برای اینکار آنها را با کاما از هم جدا کنید :

```
public class MyCollection<T> where T:class,IDisposable,new()
{
    //content
}
```

نوعی که به T نسبت داده می‌شود

باید از نوع ارجاعی باشد.

باید واسط IDisposable را پیاده سازی کرده باشد.

باید یک سازنده‌ی پیش فرض داشته باشد.

نکته دوم : زمانیکه از چندین محدودیت استفاده می‌کنید مثل مثال فوق، باید محدودیت new() در آخرین جایگاه محدودیت‌ها قرار گیرد؛ در غیر اینصورت با خطای زمان ترجمه روبه رو خواهید شد .

نکته سوم : می‌توان محدودیت‌های فوق را علاوه بر کلاس، بر روی متدهای جنریک نیز اعمال کنید:

```
public void Swap<T>(ref T val1,ref T val2) where T:struct
{
    //content
}
```

نکته چهارم : زمانیکه کلاس و یا متدهای شما بیش از یک نوع پارامتر از نوع جنریک را دریافت می‌کنند، باید محدودیت‌های مورد نظر را برای هر کدام به صورت جداگانه قید کنید. به طور مثال به کلاس زیر که دو پارمتر T و K را دارد، باید برای هر کدام جداگانه محدودیت‌های مورد نظر را اعمال کنیم (در صورت نیاز):

```
public class MyCollection<T,K> where T:class where K:IDisposable,new()
```

```
{  
//content  
}
```