

Specialization with generics

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2005-03/1689.html>

From: Michael Sparks (*mike_at_remove.this.sparks.name*)

Date: 03/06/05

Date: Sun, 06 Mar 2005 16:06:16 -0600

I started writing some code with the 2.0 beta compiler, and found out quickly that specialization is not supported with generics.

For example, I want to do something like this:

```
class number ...
class integer : number ...
class real : number ...
class multiplier<_number> where _number : number ...
class multiplier<_number> where _number : integer ...
```

Here, the first implementation of 'multiplier' will work with 'number' or anything that derives from 'number'. The second implementation of 'multiplier' will work with 'integer' or anything that derives from 'integer'. Since 'integer' derives from 'number', it is a specialization. At least that's what I'm calling it – there may be a more appropriate name in this exact context.

This may be preaching to the choir – I haven't been in this newsgroup for a while, but here is an example of where it would be useful to have the above feature:

```
class fancy_computation<_number> where _number : number
{
    public _number doit(_number x,_number y)
    {
        multiplier<_number> mult=new multiplier<_number>;
        return mult.doit(x,y);
    }
}
```

The advantage here is that I can type 'fancy_computation<integer>' and the compiler will automatically select the correct implementation of 'multiplier'.

Anyway, it look like this is not supported. I am wondering what specific limitation caused the design team to leave out support for

microsoft.public.dotnet.languages.csharp: Specialization with generics

specialization.

Thanks for any insight you can provide.

Mike