

专注Xamarin.Android与ASP.NET MVC

组建XAMARIN中国社区与培训机构（希望有更多人的帮助）

[HOME](#)[CONTACT](#)[GALLERY](#)

C#函数式编程之由函数构造函数

2015-03-17 16:30 by y-z-f, 1654 阅读, 1 评论, 收藏, 编辑

在面向对象的编程中，如果我们需要复用其他的类，我们可以通过继承来实现。而在函数式编程中我们也可以采取不同的方式来复用这些函数。今天的教程将会讲述两种方式，其中一个就是组合，将多个函数组合成为一个函数，另一个则是之前我们介绍过的部分应用，当然我们将会讲述如何将其高级化，来符合我们的使用要求。

组合

顾名思义，组合就是将函数A的结果传递给函数B。但是我们并不关注函数A的结果，当然大多数一定会这样去做：

```
1 var r1 = funcA(1);
2 var r2 = funcB(r1);
```

这样显然不是我们希望的那样，假设我们后面需要经常利用到这样的函数。问题就出现了，所以我们就需要利用组合来将他们合成一个新的函数，首先我们先写出两个用来组合的函数：

```
1 public static int FuncA(int x)
2 {
3     return x + 3;
4 }
5
6 public static int FuncB(int x)
7 {
8     return x + 6;
9 }
```

如果我们不借助任何的自动化函数，我们可以通过这样的写法来进行组合：

```
Func<int,int> funcC = x => FuncB(FuncA(x));
```

但是我们这里无法使用var，因为C#的自动推断类型无法推断出这个类型。这样我们就有了一个新的函数funcC，我们可以试着执行这个函数看看最终的结果。上面我们通过手动的方式完成了组合，下面我们将编写一个自动化的函数来完成这个操作：

```
1 public static Func<T1, T3> Compose<T1, T2, T3>(Func<T1, T2> fu
nc1, Func<T2, T3> func2)
2 {
3     return x => func2(func1(x));
4 }
```

About



目前工作：技术经理

工作单位：[江苏斯诺物联科技有限公司](#)

学过语言：C/C++、Objective-c、java、C#、javascript、Perl

读过书籍：《精通ASP.NET Web程序测试》、《SQL SERVER 2012编程入门经典》、《精通ASP.NET MVC 3框架》、《C#高级编程》、《C++ Primary 四》、《C#精髓》、《ASP.NET 4高级程序设计》、《ASP.NET 设计模式》、《JavaScript设计模式》、《JavaScript高级程序设计》等诸多名著

兴趣爱好：巧克力、技术、天文学、写技术博客

相关社区

[专注Xamarin开源社区](#) [专注Mvvmcross开源社区](#)

掌握Backbone.js、Knockout.js、Castle Active Record、Ninject、EF等框架，染指C#、Javascript、SQL Server、Jquery、Jquery unit、Jquery mobil、Jquery ui、EasyUI、ASP.NET MVC、ASP.NET、HTML5、Xamarin.Android等技术，对TDD、BDD、DDD有强热的追求、对代码工整非常苛刻。

Visitors

	365,988		230
	4,148		164
	2,784		111
	2,289		95
	2,032		95
	845		94
	373		94
	319		91
	232		84

FLAG counter

TRANSLATE



昵称：[y-z-f](#)

园龄：[4年11个月](#)

粉丝：[713](#)

关注：[14](#)

[+加关注](#)

最新随笔

[如何在Ubuntu_16_04下使用MySQL的GR](#)

[如何利用Github+Appveyor+Nuget打造自己的.net core开源库](#)

[.NET Core下使用gRpc公开服务（SSL/TLS）](#)

[.Net Core下如何管理配置文件](#)

[如何在ASP.NET 5上搭建基于TypeScript的Angular2项目](#)

[Xamarin.Android之UI Test简单入门](#)

[TypeScript之接口类型](#)

[TypeScript之基本数据类型](#)

[Xamarin.Android经验之谈](#)

[Xamarin.Android之MvvmCross](#)

[MongoDB for C#基础入门](#)

[Xamarin.IOS之多视图](#)

[Xamarin.IOS之快速入门](#)

[为Xamarin更好的开发而改写的库](#)

[C#函数式编程之可选值](#)


```
public static Func<Converter<int, int>, IEnumerable<int>, IEnumerable<int>> MapDelegate<T1, T2>()
{
    return Map<T1, T2>;
}
```

这样我们在调用Curry函数就不需要提供类型参数了：

```
Functional.Curry(Functional.MapDelegate<int, int>());
```

至此，类型推断的问题我们就解决了。在实际开发中部分应用虽然十分有用，但是在某些情形下却十分的麻烦，比如函数Filter需要两个算法，最后一个参数为数据。在实际使用中我们都会将两个算法赋进去，而在后面的使用中仅仅只会改变对应的数据，但是在采用部分应用后就显得麻烦了，下面是Filter函数的实现：

```
1 public static IEnumerable<R> Filter<T,R>(Func<T,R> map,Func<T, bool> compare, IEnumerable<T> datas)
2 {
3     foreach (T item in datas)
4     {
5         if (compare(item))
6         {
7             yield return map(item);
8         }
9     }
10 }
```

具体的功能就是通过compare函数判断是否符合条件，然后通过map函数返回需要的部分。我们可以通过如下的方式来调用这个函数：

```
1 foreach (int x in Filter<int, int>(x => x, x => x
<= 10, new int[] { 2, 3, 1, 4, 5, 3, 34 })))
2 {
3     Console.WriteLine(x);
4 }
5 Console.ReadKey();
```

在采用部分应用前，我们先写出这个函数的Delegate版本，这样我们就可以利用类型推断了：

```
1 public static Func<Func<T, R>, Func<T, bool>, IEnumerable<T>, IEnumerable<R>> FilterDelegate<T, R>()
2 {
3     return Filter<T, R>;
4 }
```

然后我们就可以轻松的使用Curry函数将其部分应用了，这里笔者直接自己实现了一个Curry函数，并没有使用FCSLib中提供的。读者可以参考下：

[5. ASP.NET MVC学习之Ajax \(完结 \) \(18\)](#)

[6. ASP.NET MVC学习之路由篇 \(1 \) \(18\)](#)

[7. Xamarin.Android之UI Test简单入门\(17\)](#)

[8. MongoDB for C#基础入门\(13\)](#)

[9. ASP.NET MVC学习之模型验证篇 \(13\)](#)

[10. .NET Core下使用gRpc公开服务 \(SSL/TLS \) \(10\)](#)

阅读排行榜

[1. C#之玩转反射\(41405\)](#)

[2. javascript库之Mustache库使用说明 \(26881\)](#)

[3. Xamarin.Android快速入门\(26405\)](#)

[4. Android开发之使用HttpURLConnection进行POST请求 \(26169\)](#)

[5. 使用 jQuery 进行前端验证 -- 1 \(17555\)](#)

[6. MongoDB for C#基础入门\(16746\)](#)

[7. ASP.NET MVC学习之路由篇 \(1 \) \(15063\)](#)

[8. ASP.NET MVC学习之Ajax \(完结 \) \(14819\)](#)

[9. Android开发之使用Handler刷新UI控件\(14214\)](#)

[10. Android开发之底部导航栏标准 \(12884\)](#)

```

1         public static Func<T1,Func<T2,Func<T3,R>>> Currey<T1,T
2,T3,R>(Func<T1,T2,T3,R> func)
3         {
4             return x => y => z => func(x, y, z);
5         }

```

最后我们通过实际的使用来看看：

```

1         var f = Currey(FilterDelegate<int, int>());
2         foreach (int x in f(x => x) (x => x <= 10) (new int
3 [] { 2, 3, 1, 4, 5, 3, 34 })))
4         {
5             Console.WriteLine(x);
6         }
7         Console.ReadKey();

```

即使这样也很繁琐，所以我们需要进行更高级的部分应用，这里我们需要另一个自动化函数来帮助我们实现：

```

1         public static Func<T3,R> Apply<T1, T2, T3, R>(Func<T1,
2 Func<T2, Func<T3, R>>> func,T1 arg1,T2 arg2)
3         {
4             return x => func(arg1) (arg2) (x);
5         }

```

这个函数的作用就是将原本的部分应用的函数变成一个接收两个参数并返回一个只接收一个参数的函数，因为算法部分不会变动，但是数据会经常的变动。下面我们通过一个实际的运用来展示：

```

1         var f = Apply(Currey(FilterDelegate<int, int>()),
2 x => x, x => x <= 10);
3
4         foreach (int x in f(new int[] { 2, 3, 1, 4, 5,
5 3, 34 })))
6         {
7             Console.WriteLine(x);
8         }
9         foreach (int x in f(new int[] { 1, 2, 3, 4, 5,
10 6, 7, 8, 9, 10, 11, 12, 13 })))
11         {
12             Console.WriteLine(x);
13         }
14         Console.ReadKey();

```

通过这样一番折腾后，我们就得到的我们真正需要的函数了，我们在一开始的时候确定算法。然后在后面的使用中我们就可以只传递数据即可。

Xamarin.Android -> Xamarin.iOS -> 混合 -> Xamarin.Forms

好文要顶

关注我

收藏该文





y-z-f

关注 - 14

粉丝 - 713

0

0

+加关注

« 上一篇: [C#函数式编程之序列](#)

» 下一篇: [C#函数式编程之可选值](#)

分类: [函数式编程](#)

#1楼 赵御辩

[ADD YOUR COMMENT](#)

2015-03-21 08:58

读者会发现类型参数就占据的一半,上面我们也介绍了如何解决这个问题,所以我们可以写个已经显式声明过类型的函数来封装下Map函数:

在这里有个笔误:

MapDelegate应该使用泛型

```
public static Func<Converter<T1, T2>, IEnumerable<T1>,
IEnumerable<T2>> MapDelegate<T1, T2>()
{
    return Map<T1, T2>;
}
```

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论,请 [登录](#) 或 [注册](#), [访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】腾讯云 十分钟定制你的第一个微信小程序

【推荐】报表开发有捷径: 快速设计轻松集成, 数据可视化和交互



最新IT新闻:

- 荷兰宣布2030年终结煤炭发电厂! 淘汰所有燃油车
 - 美国最权威《消费者报告》: 三星S8比iPhone 8更棒
 - 真·稳如狗: 中国团队推出四足机器人 对标波士顿动力
 - 小冰向虚拟歌姬洛天依粉道歉 并没想diss你们心中的偶像
 - 360安全门锁淘宝众筹上线: 0.5秒开锁/被盗最高赔10万
- » 更多新闻...



最新知识库文章:

- 实用VPC虚拟私有云设计原则
 - 如何阅读计算机科学类的书
 - Google 及其云智慧
 - 做到这一点, 你也可以成为优秀的程序员
 - 写给立志做码农的大学生
- » 更多知识库文章...