<<C#函数式程序设计>>

图书基本信息

书名：<<C#函数式程序设计>>

13位ISBN编号：9787302302346

10位ISBN编号：7302302340

出版时间：2013-1

出版时间：清华大学出版社

作者：(英)斯图姆(Sturm, O.) 著

译者：吴文国

<<C#函数式程序设计>>

内容概要

函数式设计是一种重要的程序设计模式，它可以追溯到很久之前。
函数式程序设计总是与教授程序设计的人们有关．函数式程序设计的整洁而富有逻辑的概念是它特别适合于教学的重要原因。
广泛使用计算机和自己设计程序的行业也发现函数式程序设计是实现其目标最有效的办法。
然而，在许多所谓的“主流”软件公司看来，函数式程序设计一直以来只具有学术研究价值，他们普遍选择传统的指令式设计方法，如面向对象等。

最近几年，在NET平台上把越来越多的函数式成分增加到指令式语言中。
在VisualStudio2010中增加了F#语言，它是用微软主流开发平台开发的第一个混合的函数式语言。
甚至有更多的函数式功能被引入到C#和VBNET中，这说明了微软公司对函数式设计的认同。

<<C#函数式程序设计>>

作者简介

作者：（英国）斯图姆（Oliver Sturm） 译者：吴文国 Oliver Sturm有20多年的专业软件开发经验。
他是应用程序体系结构、程序设计语言和DevExpress开发的第三方.NET工具等多个领域的专家。
自2002年开始，他的主要兴趣在于.NET平台。
Oliver曾在许多国际会议上发表过演说，编写了20多个培训课程，并在杂志上用英语和德语发表了l00多篇文章。
他也曾从事计算机基础编程教学15年之久。
由于他对.NET社区所做的贡献，因此多次获得微软英国最佳C#程序员称号。
吴文国，博士，温州大学物理与电子信息学院副教授。
其研究方向是计算机图形学和地球物理及探测技术，主要从事面向对象程序设计、数据结构等基础课程的教学工作。
他工作之余还从事软件开发和翻译工作，已翻译出版了《交互式计算机图形学——基于OpenGL的自顶向下方法（第4版）》、《UNIX原理与应用（第4版）》等10多本计算机图书。
另外，他还在《计算机辅助设计与图形学学报》、《中国物理快报》、《电子学报》等杂志上发表过多篇文章。

<<C#函数式程序设计>>

书籍目录

章节摘录

版权页： 插图： 可以看出，可以根据受扩展方法支持的字面类型调用扩展方法，也可以根据此类型的变量调用扩展方法。
SecondElement方法甚至扩展了一个泛型接口类型，在示例代码中这个类型用传入的int数组来实现。
但是同一个接口可以由不同的集合类型来实现，因此这个函数具有广泛的应用。
使用一个可以为一大群类型打开一个函数的参数类型是值得推荐的做法，但是利用扩展方法可以使得这样的辅助函数变得容易使用，更加直观。
在本书的后面章节中，读者将会看到有关扩展方法如何成为标准.NET机制的基础等令人感兴趣的细节内容，以及对许多标准扩展方法的详细介绍。
3.5 引用透明 在指令式程序设计中，编写一个计算机程序意味着定义一个为实现某个具体目标而需要的操作序列。
在这个序列中，需要定义状态和状态的转换——A状态如何转换到B状态，A状态和B状态的具体内容，以及什么时候从A状态转换为B状态。
当人们说指令式程序设计都是与状态有关时，就是指这个意思。
从理论上讲，一个序列程序可以逐行编写，程序的执行过程是从顶部到底部并在底部结束。
实际上，即使在CPU级，也总是有工具提高程序设计的效率。
程序设计语言提供了函数、方法以及其他模块，这些都可以用来增加问题的抽象性。
在指令式程序设计中，这些模块的基本作用是防止代码重复，把代码分解成更容易管理的函数级模块。
指令式程序设计的最大问题之一是随着时间的推移，模块会变得越来越大。
这在代码库中情况尤为突出。
由于指令式程序设计把重点放在执行序列上，因此函数和方法的引用总是不透明的。
这是指，即使用相同的一组输入参数（或者没有参数）调用函数，也无法保证每次都得到相同的结果。
函数的实现经常要用更大作用域中的变量（例如类级别的字段），这些变量通常称为全局变量。
正如前面虚构的由无数行代码组成的无穷序列例子中一样，选取的函数和方法必须按某个特定顺序进行调用，这样它们之外的状态才会与它们的算法相符。
引用透明正好与此相反。
这个术语可以应用于任何表达式，它可表示这样的意思：表达式可以用表达式的值取代而不会影响程序，也就是不会影响使用此替换操作的算法的最终结果。
在数学上，表达式总是引用透明的，很容易找到这方面的例子。
例如，在任何数学情形下，表达式3+2都可以用5来代替，而且不会改变其本意。
遗憾的是，在计算机程序设计中，情况并非都如此。
有些表达式或函数绝对不可以透明引用，因为它们的作用就是返回改变后的值，或者每次返回不同的值。
在.NET中，这些例子包括DateTime.Now、File.ReadByte（）或Console.Read（）。
在C#中，返回值类型为void的函数不可以作为表达式，因此它们不能透明引用。
函数式程序设计中一个重要的概念是在函数实现时避免出现副作用。
这个主题与引用透明的主题非常接近。
纯函数是指一个不会有任何副作用的函数，即一个只从一组输入参数计算得到返回值的函数。
这样的函数可以调用其他函数作为其算法的一部分。
它可以访问作用域之外的数值，前提是它们是引用透明而且绝对不会发生变化的。
想象一下，一个算法引用Math.PI值——这是允许的，因为被访问的对象是一个常量。

<<C#函数式程序设计>>

编辑推荐

《C#函数式程序设计:经典编程技术在现代项目中的应用》提供了众多的不同类型实例，这些实例结合了多个方法解决不同领域里的问题。
既包括了并行计算和高性能计算等复杂的问题，也包括了Web服务和尚业逻辑实现等简单的用例。
《C#函数式程序设计:经典编程技术在现代项目中的应用》希望帮助程序员在C#语言里找到问题的解决方案，并向读者介绍C#函数式编程的优点和缺点。
《C#函数式程序设计:经典编程技术在现代项目中的应用》的主要目的是帮助程序员最大限度地利用已知的程序设计技术。

<<C#函数式程序设计>>

版权说明