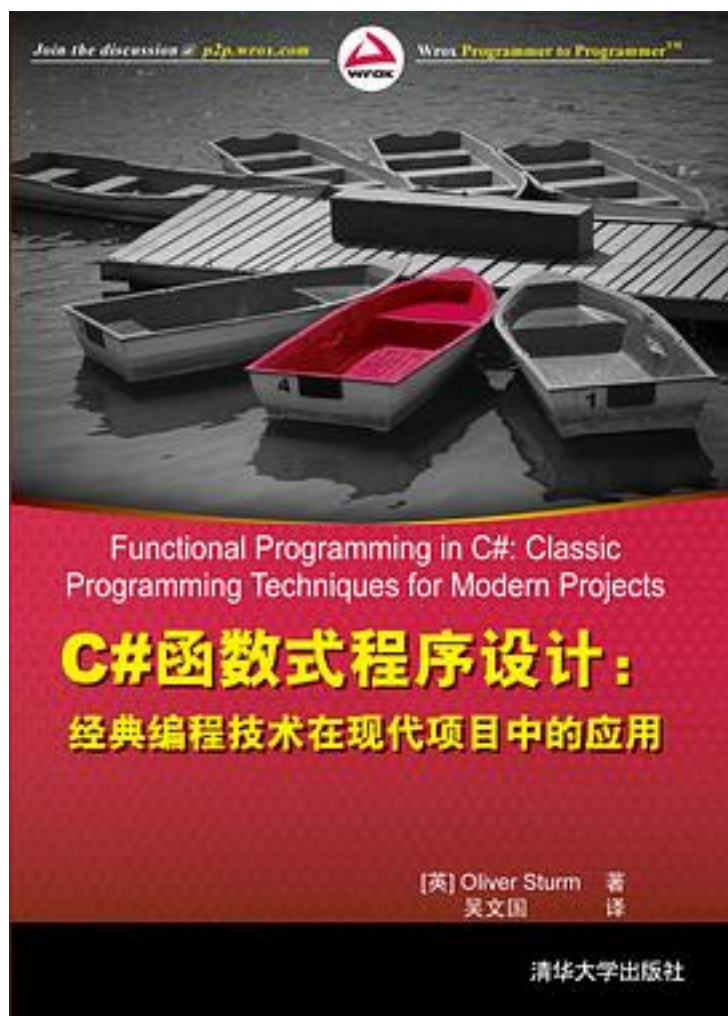


C#函数式程序设计



[C#函数式程序设计_下载链接1](#)

著者:(英)斯图姆(Sturm, O.) 著

出版者:清华大学出版社

出版时间:2013-1

装帧:平装

isbn:9787302302346

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Visual Studio 2010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VB.NET中，这说明了微软公司对函数式设计的认同。

作者介绍:

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100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。以苏格兰为据点，Oliver主要从事自由咨询师和培训师的工作，同时还是国际咨询公司thinktecture的顾问。他的个人博客是www.sturmnet.org/blog，商业网址是www.oliversturm.com，电子邮件地址是oliver@oliversturm.com。

目录: 目录

第 I 部分 函数式程序设计引言

第1章 函数式程序设计简史 3

1.1 函数式程序设计简介 3

1.2 函数式程序设计语言 4

1.3 与面向对象程序设计的关系 7

1.4 小结 7

第2章 函数式程序设计思想在现代项目中的应用 9

2.1 控制副作用 10

2.2 敏捷开发方法 11

2.3 声明式程序设计 11

2.4 函数式程序设计的定向思维 11

2.5 用C#实现函数式程序设计的可行性 12

2.6 小结 13

第 II 部分 C#函数式程序设计基础

第3章 函数、委托和Lambda表达式 17

3.1 函数与方法 17

3.2 重用函数 19

3.3 匿名函数与Lambda表达式 22

3.4 扩展方法 25

3.5 引用透明 27

3.6 小结 29

第4章 泛型 31

4.1 泛型函数 32

4.2 泛型类 33

4.3 约束类型 35

4.4 其他泛型类型	36
4.5 协变与逆变	38
4.6 小结	41
第5章 惰性列表工具—迭代器	43
5.1 什么是惰性	43
5.2 用.NET方法枚举元素	44
5.3 迭代器函数的实现	47
5.4 链式迭代器	51
5.5 小结	53
第6章 用闭包封装数据	55
6.1 动态创建函数	55
6.2 作用域存在的问题	56
6.3 闭包的工作机制	56
6.4 小结	60
第7章 代码即数据	61
7.1 .NET中的表达式树	62
7.2 分析表达式	63
7.3 生成表达式	68
7.4 .NET 4.0特性	71
7.5 小结	73
第Ⅲ部分 用C#实现常用的函数式设计技术	
第8章 局部套用与部分应用	77
8.1 参数的解耦	77
8.1.1 手动局部套用	78
8.1.2 自动局部套用	79
8.1.3 调用局部套用函数	81
8.1.4 类上下文	81
8.1.5 FCSlib库的内容	84
8.2 调用函数的各部分	86
8.3 参数顺序的重要性	88
8.4 小结	89
第9章 惰性求值	91
9.1 惰性求值的优点	92
9.2 传递函数	93
9.3 显式的惰性求值	94
9.4 惰性求值方法的比较	98
9.4.1 可用性	98
9.4.2 效率	98
9.5 惰性求值方法的选择	99
9.6 小结	99
第10章 缓存技术	101
10.1 记住以前结果的重要性	101
10.2 预计算	102
10.3 缓存	107
10.3.1 深度缓存	110
10.3.2 缓存的几个考虑因素	113
10.4 小结	114
第11章 递归调用	115
11.1 C#中的递归	115
11.2 尾递归	117
11.3 累加器传递模式	119
11.4 后继传递模式	120
11.5 间接递归	123

- 11.6 小结 126
- 第12章 标准高阶函数 127
 - 12.1 应用运算：Map 127
 - 12.2 使用筛选条件：Filter 128
 - 12.3 累加操作：Fold 129
 - 12.4 LINQ中的Map、Filter和Fold 134
 - 12.5 标准高阶函数 135
 - 12.6 小结 136
- 第13章 序列 137
 - 13.1 何为列表推导 137
 - 13.2 用函数方法实现迭代器 138
 - 13.3 值域 139
 - 13.4 限制 141
 - 13.5 小结 143
- 第14章 由函数构建函数 145
 - 14.1 组合函数 145
 - 14.2 高级的部分应用 148
 - 14.3 各种方法的综合 150
 - 14.4 小结 154
- 第15章 可选值 155
 - 15.1 空值的含义 155
 - 15.2 可选值的实现 156
 - 15.3 小结 161
- 第16章 防止数据变化 163
 - 16.1 变化不总是件好事 163
 - 16.2 错误的假定 164
 - 16.2.1 静态数据受欢迎 165
 - 16.2.2 深度问题 166
 - 16.2.3 克隆 167
 - 16.2.4 自动克隆 168
 - 16.3 实现不可变容器数据类型 172
 - 16.3.1 链表 172
 - 16.3.2 队列 178
 - 16.3.3 非平衡的二叉树 180
 - 16.3.4 红黑树 183
 - 16.4 持久数据类型的替代选择 185
 - 16.5 小结 186
- 第17章 单子 187
 - 17.1 类型类的概念 188
 - 17.2 单子的概念 191
 - 17.3 使用抽象的原因 191
 - 17.4 Logger单子 195
 - 17.5 含糖语法 197
 - 17.6 用SelectMany方法建立绑定 197
 - 17.7 小结 199
- 第IV部分 函数式设计的实际应用
- 第18章 函数式程序设计技术的综合应用 203
 - 18.1 重构 204
 - 18.1.1 用Windows Forms UI实现列表筛选 204
 - 18.1.2 Mandelbrot分形计算 210

- 18.2 编写新代码 217
 - 18.2.1 使用静态方法 217
 - 18.2.2 优先考虑匿名函数 219
 - 18.2.3 优先考虑高阶函数 220
 - 18.2.4 优先考虑不可变数据 221
 - 18.2.5 注意类中行为的实现 222
- 18.3 寻找可以替代函数式设计的方法 222
 - 18.3.1 其他需要考虑的问题 222
 - 18.3.2 使用已有代码 224
- 18.4 小结 225
- 第19章 MapReduce模式 227
 - 19.1 MapReduce的实现 228
 - 19.2 问题的抽象 231
 - 19.3 小结 233
- 第20章 函数模块化思想的应用 235
 - 20.1 在应用程序中执行SQL代码 235
 - 20.2 用部分应用和预计算重写函数 237
 - 20.3 小结 239
- 第21章 函数式技术在现有项目中的应用 241
 - 21.1 .NET Framework 241
 - 21.2 LINQ 243
 - 21.2.1 LINQ to Objects 243
 - 21.2.2 LINQ到查询后台 247
 - 21.2.3 并行化 249
 - 21.3 Google MapReduce及其实现 250
 - 21.4 NUnit 252
 - 21.5 小结 254
 - • • • • (收起)

[C#函数式程序设计_下载链接1](#)

标签

C

#软件开发

计算机技术

函数式编程

设计模式

计算机

quant

paper

评论

C# 函数式编程从入门到放弃

内容尚可，没太多新东西。体会FP最好的方法还是尝试使用一下函数式语言，比如F#或是Scala。但我对在团队项目中使用F#还是不太乐观，一是因为代码必须要让团队都能了解且维护，二是C#实在设计得太好。

好书，非常深入。

[C#函数式程序设计_下载链接1](#)

书评

[C#函数式程序设计_下载链接1](#)