Сошников Д. В.

Функциональное программирование на F#



Москва, 2011

УДК 004.438F#
ББК 32.973.26-018.1
С54

Сошников Д. В.

С54 Функциональное программирование на F#. – М.: ДМК Пресс, 2011. – 192 с.: ил.

ISBN 978-5-94074-689-8

Автор этой книги имеет богатый опыт преподавания курсов функционального программирования на базе F# в ведущих российских университетах, в то же время, будучи технологическим евангелистом Майкрософт, он умеет доходчиво объяснить концепции функционального программирования даже начинающему разработчику ПО, не прибегая к сложным понятиям лямбда-исчисления.

Книга содержит много полезных примеров использования F# для решения практических задач: доступа к реляционным или слабоструктурированным XML-данным, использование F# для веб-разработки и веб-майнинга, визуализация данных и построение диаграмм, написание сервисов для облачных вычислений и асинхронных приложений для Windows Phone 7. Используя фрагменты кода, рассмотренные в книге, читатели могут немедленно приступить к решению своих практических задач на F#.

УДК 004.438F#
ББК 32.973.26-018.1

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но поскольку вероятность технических ошибок все равно существует, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-5-94074-689-8

© Сошников Д. В., 2011
© Оформление, издание, ДМК Пресс, 2011

Содержание

Предисловие Дона Сайма 6

Предисловие автора 8

0. Введение 10
0.1. Зачем изучать функциональное программирование 10
0.2. О чем и для кого эта книга 12
0.3. Как установить и начать использовать F# 13

1. Основы функционального программирования 17
1.1. Применение функций vs. Присваивание 17
1.2. Упорядоченные кортежи, списки и вывод типов 19
1.3. Функциональные типы и описание функций 20
1.4. Каррирование 22
1.5. Условный оператор и опциональный тип 23
1.6. Типы данных, размеченное объединение и сопоставление с образцом 25
1.7. Рекурсия, функции-параметры и цикл for 27
1.8. Конструкции >>, |> 28
1.9. Пример – построение множества Мандельброта 29
1.10. Интероперабельность с .NET 31

2. Рекурсивные структуры данных 34
2.1. Списки и конструкторы списков 34
2.2. Сопоставление с образцом 35
2.3. Простейшие функции обработки списков 36
2.4. Функции высших порядков 37
2.4.1. Отображение 37
2.4.2. Фильтрация 39
2.4.3. Свертка 41
2.4.4. Другие функции высших порядков 43
2.5. Генераторы списков 44
2.6. Хвостовая рекурсия 45
2.7. Сложностные особенности работы со списками 47
2.8. Массивы 50
2.9. Многомерные массивы и матрицы 52
2.9.1. Списки списков, или непрямоугольные массивы (Jagged Arrays) 52
2.9.2. Многомерные массивы .NET 53
2.9.3. Специализированные типы для матриц и векторов 54
2.9.4. Разреженные матрицы 55

2.9.5. Использование сторонних математических пакетов ... 56
2.10. Деревья общего вида ... 56
2.11. Двоичные деревья ... 59
2.11.1. Определение ... 59
2.11.2. Обход двоичных деревьев ... 59
2.11.3. Деревья поиска ... 60
2.11.4. Деревья выражений и абстрактные синтаксические деревья (AST) ... 62
2.12. Другие структуры данных ... 63
2.12.1. Множества (Set) ... 63
2.12.2. Отображения (Map) ... 63
2.12.3. Хеш-таблицы ... 64

3. Типовые приемы функционального программирования ... 66
3.1. Замыкания ... 66
3.2. Динамическое связывание и mutable-переменные ... 67
3.3. Генераторы и ссылочные переменные ref ... 68
3.4. Ленивые последовательности (seq) ... 71
3.4.1. Построение частотного словаря текстового файла ... 73
3.4.2. Вычисление числа π методом Монте-Карло ... 74
3.5. Ленивые и энергичные вычисления ... 76
3.6. Мемоизация ... 79
3.7. Продолжения ... 81

4. Императивные и объектно-ориентированные возможности F# ... 84
4.1. Мультипарадигмальность языка F# ... 84
4.2. Элементы императивного программирования на F# ... 85
4.2.1. Использование изменяемых переменных и ссылок ... 85
4.2.2. Цикл с предусловием ... 86
4.2.3. Условный оператор ... 87
4.2.4. Null-значения ... 87
4.2.5. Обработка исключительных ситуаций ... 87
4.3. Объектно-ориентированное программирование на F# ... 89
4.3.1. Записи ... 89
4.3.2. Моделирование объектной ориентированности через записи и замыкания ... 90
4.3.3. Методы ... 91
4.3.4. Интерфейсы ... 92
4.3.5. Создание классов с помощью делегирования ... 93
4.3.6. Создание иерархии классов ... 94
4.3.7. Расширение функциональности имеющихся классов ... 97
4.3.8. Модули ... 97

5. Метапрограммирование ... 99
5.1. Языково-ориентированное программирование ... 99
5.2. Активные шаблоны ... 102
5.3. Квотирование ... 103

5.4. Конструирование выражений, частичное применение функции и суперкомпиляция 106
5.5. Монады 107
5.5.1. Монада ввода-вывода 108
5.5.2. Монадические свойства 110
5.5.3. Монада недетерминированных вычислений 111
5.6. Монадические выражения 112

6. Параллельное и асинхронное программирование 115
6.1. Асинхронные выражения и параллельное программирование 115
6.2. Асинхронное программирование 116
6.3. Асинхронно-параллельная обработка файлов 118
6.4. Агентный паттерн проектирования 120
6.5. Использование MPI 122

7. Решение типовых задач 127
7.1. Вычислительные задачи 127
7.1.1. Вычисления с высокой точностью 127
7.1.2. Комплексный тип 128
7.1.3. Единицы измерения 128
7.1.4. Использование сторонних математических пакетов 129
7.2. Доступ к данным 131
7.2.1. Доступ к реляционным базам данных (SQL Server) 131
7.2.2. Доступ к слабоструктурированным данным XML 136
7.2.3. Работа с данными в Microsoft Excel 139
7.3. Веб-программирование 143
7.3.1. Доступ к веб-сервисам, XML-данным, RSS-потокам 144
7.3.2. Доступ к текстовому содержимому веб-страниц 144
7.3.3. Использование веб-ориентированных программных интерфейсов на примере Bing Search API 147
7.3.4. Реализация веб-приложений на F# для ASP.NET Web Forms 148
7.3.5. Реализация веб-приложений на F# для ASP.NET MVC 150
7.3.6. Реализация веб-приложений на F# при помощи системы WebSharper 152
7.3.7. Облачное программирование на F# для Windows Azure 156
7.4. Визуализация и работа с графикой 158
7.4.1. Двухмерная графика на основе Windows Forms API 159
7.4.2. Использование элемента Chart 160
7.4.3. 3D-визуализация с помощью DirectX и/или XNA 164
7.5. Анализ текстов и построение компиляторов 171
7.4.1. Реализация синтаксического разбора методом рекурсивного спуска 171
7.4.2. Использование fslex и fsyacc 174
7.5. Создание F#-приложений для Silverlight и Windows Phone 7 179

Вместо заключения 185

Рекомендуемая литература 190

Предисловие Дона Сайма

F# является достаточно молодым языком программирования – его разработку мы начали в MicrosoftResearch в Кембридже 7 лет назад. С тех пор язык стал богатым и выразительным инструментом индустриального программирования, использующимся в различных областях, в особенности после того, как он был включен в стандартную поставку VisualStudio 2010.

В прошлом году мне довелось выступать с несколькими виртуальными докладами на конференциях по разработке ПО в России, и я убедился, что здесь есть значительный интерес к F#, как среди разработчиков ПО, так и в кругу любителей функционального программирования. Поэтому мне особенно приятно, что Дмитрий взял на себя тяжелый труд по написанию первой русскоязычной книги по F#.

F# – это интересный язык, поскольку он является не только эффективным инструментом для разработки коммерческого ПО на платформе .NET, но и очень удобным академическим языком для преподавания функционального программирования в университетах (использование для этой цели в качестве базового языка, который имеет индустриальную поддержку, имеет много преимуществ), а также исследовательским языком (на самом деле многие идеи для следующего поколения .NET-языков приходят из F#, взять хотя бы асинхронные блоки, агенты и типо-образующие классы). Дмитрий Сошников, автор этой книги, имеет богатый опыт преподавания курсов функционального программирования на базе F# в ведущих российских университетах, в то же время, будучи технологическим евангелистом Майкрософт, он умеет доходчиво объяснить концепции функционального программирования даже начинающему разработчику ПО, не прибегая к сложным понятиям лямбда-исчисления.

Эта книга в первую очередь ориентирована на практикующих программистов, которые найдут в ней хорошее введение в функциональное программирование. Затем в книге следует более глубокое изложение приемов функционального программирования, полезные главы по объектно-ориентированному, асинхронному и параллельному программированию на F#. Книга также содержит много полезных примеров использования F# для решения практических задач: доступа к реляционным или слабоструктурированным XML-данным, использование F# для веб-разработки (как на ASP.NETWebForms/MVC, так и с применением WebSharper) и веб-майнинга, визуализация данных и построение диаграмм, написание сервисов для облачных вычислений и асинхронных приложений для WindowsPhone 7. Используя фрагменты кода, рассмотренные в книге (а также доступные для скачивания со странички автора из Интернет), читатели могут немедленно приступить к решению своих практических задач на F#.

Я искренне надеюсь, что эта книга поможет раскрыть красоту, богатство и мощь языка F# для многих разработчиков ПО из России и других русско-говорящих стран.

Дон Сайм,
ведущий исследователь, MicrosoftResearch Cambridge,
создатель языка F#

Предисловие автора

Решение взяться за написание книги по F# было очень непростым. Во-первых, не было понятно, насколько такая книга будет востребована. Во-вторых, сложно конкурировать с уже появившейся полудюжиной англоязычных книг по этому языку. И наконец, написание книги – это очень трудоемкий процесс, учитывая, что он является далеко не единственным делом моей жизни.

Однако, встречаясь со многими студентами и разработчиками и выступая в вузах, я понял, что интерес к языку огромен, а необходимость в русскоязычных книгах имеется, поскольку не все начинающие изучать программирование хорошо владеют английским языком, – а хотелось показать прелести функционального программирования не только профессионалам, но и **всем**. Поэтому в какой-то момент решение было принято, а результатом стало то, что вы держите в руках.

Тем не менее мне не хотелось дублировать существующие англоязычные книги. Основной целью было создать небольшую книгу, которая позволит практикующим программистам и начинающим за короткий срок овладеть как основами функционального программирования, так и базовым синтаксисом языка F#. Для удобства в книге сначала рассматриваются базовые понятия, которые позволят вам быстро (после прочтения первых трех глав) начать писать на F# весьма нетривиальные программы и понимать чужой код. Последняя, седьмая глава содержит в себе множество коротких и лаконичных примеров и фрагментов кода, многие из которых вы можете использовать с минимальными изменениями для решения своих задач. Таким образом, после прочтения книги вы не только «расширите свое сознание», изучив еще один подход к программированию, но и пополните свой арсенал чрезвычайно мощным средством для решения различных задач обработки данных.

Создание этой книги было бы невозможно без участия многих моих друзей и коллег. Первоначальный интерес к функциональному программированию появился у меня в результате бесед с профессором В. Э. Вольфенгагеном и другими моими знакомыми и коллегами из «кибернетической школы» МИФИ, в частности А. В. Гавриловым и С. В. Зыковым; в дальнейшем он был подкреплен знакомством с Саймоном Пейтоном-Джонсом, одним из создателей языка Haskell, ныне работающим в Microsoft Research в Кембридже. После того как благодаря в первую очередь усилиям Дона Сайма F# был включен в состав Visual Studio 2010, я стал более плотно заниматься вопросами продвижения функционального программирования в обучение в рамках своих обязанностей как технологического евангелиста Майкрософт. Я хотел бы поблагодарить коллег из Новосибирска (ИСИ СО РАН, НГУ и НГТУ): А. Г. Марчука, Л. И. Городнюю и Н. В. Шилова за

плодотворную дискуссию на тему использования F# для преподавания в рамках семинара ИСИ СО РАН, окончательно убедившую меня в том, что F# может помочь решить благую задачу внедрения в учебный процесс курсов функционального программирования, которые при этом будут иметь значительную практическую направленность.

Мне посчастливилось поставить и прочитать такие курсы на базе F# в ведущих московских вузах: МФТИ и ГУ ВШЭ – за эту возможность я благодарен В. Е. Кривцову и С. М. Авдошину. Многие материалы книги основаны на этих курсах, которые я вел совместно с С. Лаптевым и С. В. Косиковым. Некоторые примеры были разработаны студентами Т. Абаевым (ГУ ВШЭ), А. Брагиным (МАИ), А. Мыльцевым (МФТИ). Благодаря заинтересованности и поддержке А. В. Шкреда видеокурс доступен в рамках интернет-университета информационных технологий ИНТУИТ.РУ.

За идею и за возможность издать книгу по языку F# я благодарен Д. А. Мовчану и сотрудникам «ДМК Пресс», причастным к подготовке книги. Я также благодарен моим друзьям и коллегам: С. Звездину (ЮУрГУ), В. Юневу, Ю. Трухину (ТвГТУ), которые любезно согласились прочитать рукопись и высказать свои пожелания и дополнения. Многие вопросы о целесообразности издания такой книги мы обсуждали с В. Е. Зайцевым (МАИ), а проблемы изложения основ функционального подхода – с моим другом Р. В. Шаминым (РУДН). Мне очень важна была также идеологическая поддержка создателя языка F# Дона Сайма (Microsoft Research Cambridge), который любезно согласился написать предисловие.

Наконец, хочу поблагодарить мою дочь Вики, которая регулярно терпеливо недополучала внимание отца, уходящее на написание этой книги. Очень хотел бы надеяться, что потраченные на книгу усилия того стоят и помогут зародить любовь к функциональному программированию и к языку F# в сердцах многих начинающих и уже профессионально практикующих разработчиков и архитекторов.

0. Введение

0.1. Зачем изучать функциональное программирование

Вы держите в руках книгу по новому языку программирования F#, которая также для многих будет путеводителем в новый мир функционального программирования. Как вы узнаете из этой книги, функциональное программирование – это вовсе не стиль программирования, в котором используются много функций, это – другая парадигма программирования, где нет переменных, где не может возникнуть побочный эффект и в которой можно писать более короткие программы, требующие меньшей отладки.

Для начала хотелось бы немного пояснить, зачем же изучать F# и функциональное программирование вообще. До недавнего времени считалось, что функциональные языки используются в основном в исследовательских проектах, поскольку для реальных задач они недостаточно производительны. Действительно, в таких языках в обилии используются сложные структуры данных с динамическим распределением памяти, применяется сборка мусора, реализован более сложный (но и более гибкий) механизм вызова функций и т. д. Кроме того, есть мнение, что только специалисты с ученой степенью способны в них разобраться.

Действительно, функциональные языки представляют из себя очень удобный аппарат для научных исследований в области теоретического программирования, а также инструмент быстрого прототипирования систем, связанных с обработкой данных. Однако можно привести и несколько больших и известных программных систем широкого назначения, реализованных на функциональных языках: среди них графические системы компании Autodesk (использующие диалект языка LISP), текстовый редактор GNU emacs и др. Однако подавляющее большинство промышленных программных систем остаются написанными на «классических» императивных языках типа C#, Java или C++.

Однако в последнее время наблюдается тенденция все большего проникновения функционального подхода в область индустриального программирования. Современные функциональные языки – такие как Haskell, Ocaml, Scheme, Erlang – приобретают все большую популярность. В довершение всего в недрах Microsoft Research на базе OCaml был разработан язык F# для платформы .NET, который было решено включить в базовую поставку Visual Studio 2010 наравне с традиционными языками C# и Visual Basic.NET. Это беспрецедентное решение открывает возможности функционального программирования для большого круга разработ-