中国科学出版集团
新世纪书局

全球外包100强企业文思创新资深副总裁张涛
帮你应对最新开发平台的挑战 Visual Studio 2010 + C# 4.0

专业推荐

.NET 实践之旅

足够“大”又足够“小”的实例
完整开发流程，进行思维训练
结合必备知识与解决问题的方法

C#篇

黄凯波 编著
微软高级项目经理，现任文思创新部门经理

本书帮你解决的问题

- 作为团队的新丁，我该做什么，不该做什么？
- 遇到一个解决不了的技术问题，该怎么办？
- 技术发展得太快，我该怎么学习？
- 我最怕被问及什么时候能完成某个任务！
- 我最烦做计划！

科学出版社

全球外包100强企业文思创新资深副总裁张涛
帮你应对最新开发平台的挑战 Visual Studio 2010 + C# 4.0

专业推荐

.NET 实践之旅

C#篇

黄凯波 编著

科学出版社

内 容 简 介

本书通过一个模拟的实例，逐步介绍解决问题的思路、方法和良好的习惯，帮助刚入行的人员拿起手边简单的武器解决所遇到的问题。同时采用比较的方法介绍.NET Framework 4.0（C# 4.0）的一些重要知识点。

本书分为主辅两大部分：第一部分为主线，讲述一个足够“大”（同时也是足够“小”）的模拟工程；第二部分为辅线，介绍 C#以及.NET Framework 的基础和特性，采用结合实际工程的方式来引入这些重要的知识点，说明为什么用它们以及怎么使用，并且阐述这些技术的限制，以帮助读者形成自己的技术判断能力，这些知识也是公司比较喜欢的面试题。两个部分相互索引，相辅相成，让读者了解实际工作中可能遇到的问题和所需的知识点，也可以反过来学习这些技术在实际工作中是如何选择和应用的。

最重要的是，本书将作者工作时的心得体会穿插在章节之中。书中所有的关键技术术语也会在括号中给出对应的英文单词，以方便读者阅读及搜索外文资料。

本书针对因工作等需要使用 C#（.NET Framework）来完成软件项目的人群，可供 C#编程人员参考，也可作为大中专院校使用 C#进行编程课程的教材。

图书在版编目（CIP）数据

.NET 实践之旅.C#篇/黄凯波编著，—北京：科学出版社，2010.8

ISBN 978-7-03-028653-6

I.①N… II. ①黄… III. ①计算机网络—程序设计 ②C 语言—程序设计 IV. ①TP393.09 ②TP312

中国版本图书馆 CIP 数据核字（2010）第 159214 号

责任编辑：张 鑫 陈 洁 / 责任校对：刘雪连
责任印刷：新世纪书局 / 封面设计：彭琳君

科学出版社 出版
北京东黄城根北街 16 号
邮政编码：100717
http //www sciencep com

中国科学出版集团新世纪书局策划
北京市艺辉印刷有限公司印刷
中国科学出版集团新世纪书局发行 各地新华书店经销
*
2010 年 9 月 第 一 版 开本：16 开
2010 年 9 月第一次印刷 印张：29.25
印数：1—3 000 字数：711 000

定价：59.00 元

（如有印装质量问题，我社负责调换）

序

——实实在在的经验

早就答应给这本书写篇序，但一直忙于别的事情，写序的事就被拖下来。凯波很淡定，也不催我。时间一长，我倒不好意思了，见到凯波就绕开走。终于，凯波的耐心赢了。

五年前我回国，进入了软件外包行业，因此跟凯波成为了同事。我还记得第一次同凯波开会，他跟我谈了很多工程中的事。那时越来越多的微软工程被放到中国来做，每一个工程都需要有经验的软件工程师。虽然国内每年有大量的软件专业的毕业生，但马上就能上手的并不多，即使是那些有几年编程经验的也未必能胜任微软的工作。凯波在如何培训新员工、如何组织项目团队、提高工作效率等方面有很多建议。他是个肯动脑子、很有想法的人。我有时在想，凯波思想活跃应该跟他酷爱摄影有关。摄影能教会人去观察、比较、总结。更不要说，在野外摄影时随时会遇到困难，要能随机应变、迅速解决问题，才能坚持到最后。

十几年前我也编写了一本介绍 C 语言的书，在美国出版，我深知写作的艰辛。有时作者一不留神，由着自己的性子，信马由缰，那边读者就不高兴了。记得最初我高估了老美的数学基础，书中用了很多国内中学就学到的数学知识，结果被编辑打回，告知要重写。

凯波比我要幸运多了，他的读者群在数学方面的功底一定比较强。更重要的是，凯波通过模拟的工程，不但把 C# 和 .NET Framework 的基本概念介绍给读者，而且把实际工作中的流程以及团队的协作在书中做了一番演示，以便大家在实际的工作中能很快上手，很快出成绩。

写书、读书是个团队工作（team work），需要团队的协作。作者写完书了，只是完成了工作的一半，剩下的一半要靠读者来完成——通过阅读此书，你们完成一次有挑战的思维训练，也经历一次丰富的模拟实践之旅，学到一些实实在在的经验。

好了，请系上安全带、握紧方向盘、发动车……祝各位有一次愉快的阅读旅行！

文思创新 资深副总裁
Senior VP
张涛

PREFACE

前言

本书尝试解决的问题

现在学校都会开一些很接近商业应用的课程，社会上也有很多实用的培训课程，但是刚参加工作一两年的朋友或者是刚毕业的学生都会发现上述课程所教授的知识如蜻蜓点水，完全抓不到重点，或太肤浅或有遗漏，应付不了第一轮面试。即便加入了公司，也要花很长时间去学习和领悟如何使用合适的方法与技术来解决实际问题，并且花时间逐渐形成一些良好的工作习惯。虽然本书所有的代码都是用C#编写的，但是所要解决的问题不仅仅是帮助新手如何掌握这种语言（市面上有很多好书或者网站[如 MSDN]更详细地介绍了 C#），更重要的是帮助新手在可能的范围内分析、解决工程实践中的各种问题。本书通过模拟实际工程的流程，带领读者一同分析其中的问题和基于C#的解决方案，从而提供一种思维训练的机会，希望通过这些训练让读者了解企业的思维方式，缩短将来适应环境的时间。这个模拟的工程突出C#和 .NET Framework 的基础和特性（包含最新的 .NET Framework 4.0），让读者了解C#是如何与各种工程理论、开发工具、团队协作相结合的，和这些因素之间如何作用，最终解决一些“实际”问题。

C#和 .NET Framework

本书是基于C#编程语言的教程。自从 Java 在 1995 年诞生之日起，各大公司之间的争斗就越发激烈。微软最终按捺不住，在 2000 年末推出自己的商用语言——C#，并且提交给 ECMA 作为一个开放标准的语言（详情请查阅 ECMA-334 的文档）。大家如果留心就会发现这个#拆开很像++，由此推论 C#很希望和C++一样去影响编程世界。正如标准文档 ECMA-334 中所提到的，“C# combines the high productivity of Rapid Application Development（RAD） languages and the raw power of C++.”（C#集合快速应用开发的高效生产力和C++的天生强大能力），C#现在是最“成功”的商用编程语言之一，掌握了它就等于打开了其中一扇技术平台之门。不得不提的是C#语言是建立在强大的 .NET Framework 的基础上的。在本书面世的时候，.NET Framework 4.0 已经发布。.NET Framework 包含一个 CLR（Common Language Runtime，公共语言运行时）和一个庞大的类库。CLR 起到类似 Java 虚拟机（Application Virtual Machine）的作用，是微软对 CLI（Common Language Infrastructure）的实现，让 C#编译后的代码在理论上也可以“一次编写，随处运行”（write-once-run-everywhere）。.NET Framework 虽然支持多种语言，但是C#是为它量身定做的，可以充分发挥 .NET Framework 的威力。而且，C#在微软内外都是在.NET 平台上使用最多的语言。.NET Framework 把源代码编译成中间代码（Intermediate Language，IL）并且提供两套不断

完善中的类库，即 BCL（Base Class Library）和 FCL（Framework Class Library）。详细介绍请参阅 MSDN 相关主题。

本书的出发点

编写本书的出发点有两个：一是认为学习应该从某个工具的缺陷开始，称之为"边际学习法"；二是"The Law of Leaky Abstractions"（抽象缺失定律，由 Joel Spolsky 于 2002 年提出）。这两点会在书中逐渐展开。

本书结构和内容简介

本书分为主辅两大部分：第一部分为主线，讲述一个足够"大"（同时也是足够"小"）的模拟工程；第二部分为辅线，讲述 C#以及 .NET Framework 的基础和特性。两个部分相互索引，相辅相成，让读者可以知道实际工作中可能遇到的问题和所需的知识点，也可以反过来了解到技术在实际工作中是如何选择和应用的。这个布局便于读者做手册式的查阅，也便于在复习的时候，可以快速地翻阅知识点或者是工程相关的内容。

PART 01 是工程实战。这个模拟的工程一开始会非常小，只有几行代码，但足以演示 C#最基本的功能，而又不会像 Hello World 那样没有延续性。随着环节的步进和问题的深入，内容会逐渐丰满。整个工程环节中，比较重要的内容简述如下。

- 关键技术调研是立项的先决条件，本书会详细介绍这些关键部分所会遇到的困难和解决思路。Chapter 01 中另一个关键部分是带出了本书第一个例程。这个 C#的例程虽然有很多缺陷，但已经是独立并可实际使用的小程序。
- 计划制定从来都是个难点。如何在有限的、不充分的、甚至不可靠的事实基础上，估计什么时候能够完成一个功能是非常考验整体判断力的。Chapter 02 中介绍一些实用的经验和一些可操作的分析方法，如"关键路径法"。在制定计划时还要遵守相关规范与约定，在 Chapter 03 中介绍。
- 需求变更是工程中最重要的内容之一。软件工程中需求和技术都几乎无法避免地会因为某些原因发生改变。Chapter 04 会带读者去"体现"其中的冲突和应对方案。
- 架构设计和重构（Chapter 05 和 Chapter 06）是一个重要的编程思维训练。即便是新手，也需要理解所使用的架构才能够更好地做份内的事情。在工程进行当中进行重构是一把双刃剑。合适范围和深度的重构会积极地推动工程；否则，可能会因为低估工程的复杂度而让工程难以收拾，或者中后期的过度重构让工程无限延期。所以重构非常值得仔细斟酌推敲，同时它也是很好的思维训练。
- 软件调试以及相关的内容。随着工程规模的逐渐加大，软件调试变得越来越重要，即便是开发人员也开始需要了解和测试相关的内容。调试不仅仅影响开发，也影响到测试。Chapter 07 重点介绍测试的思维方式和部分测试方法，如何利用 .NET Framework 的工具类输入调试信息，并且定制调试信息和输出的方式。

在每个章节之前，都会列举此章所用的 C#和 .NET Framework 的技术知识，并给出其所在

PART 02 的位置。

PART 02 是.NET Framework 基础。所有的知识点都很重要，而且是经过实践证明比较难理解和掌握的，也就是说在抛开工程实践的情况下，难以通过单纯看书和资料来理解和掌握。本书中会结合实际工程来引入这些重要的知识点，说明为什么用它们，怎么使用，并且阐述这些技术的限制，以帮助读者形成自己的技术判断能力。同时，这些知识也是各大公司比较喜欢的面试题。例如，垃圾回收器（Garbage Collection）、异常处理（Exception Handling）、Object 的生命周期（Object Life Cycle）、应用程序域（Application Domain）、反射（Reflection）、特性（Attribute）、装箱/拆箱（Boxing/Unboxing）、XML 序列化（XML Serialization）、委托（Delegate）和 .NET 多线程（.NET Multi-Thread）。其中，垃圾回收器、异常处理和 Object 的生命周期对于了解 .NET Framework 运行机制很有帮助；反射和特性是可维护设计的重要技术基础；应用程序域、装箱/拆箱和委托是 .NET 重要的基础常识，影响编程的方方面面；XML 现在已经是最流行的基础技术，如何有效地使用还是有很多需要注意的地方。书中还提到不少 C# 和.NET Framework 从 2.0 到 4.0 的新特性，如语言集成查询（LINQ）、匿名类型（Anonymous Types）、匿名方法（Anonymous Method）、Lambda 表达式、自动属性（Automatic Properties）、.NET 并行扩展（.NET Parallel Extensions）。C#和 .NET Framework 的基础内容中还有一些容易理解和掌握的，没有在本书中单独介绍，而是在需要介绍的时候，介绍其需要注意的环节。

除了上述内容，本书还会在章节中提及一些其他扩展资源，如网站、博客、书籍等供读者参考，书的最后有全部索引。最重要的是，还有工作时的心得体会也穿插在章节之中。书中所有的关键技术术语在括号中给出对应的英文单词，以方便读者阅读、搜索外文资料。

使用本书所需要的其他资源

使用本书的读者最少需要一台可以运行 Windows XP 操作系统的计算机。比较好的 Windows 操作系统平台是 Windows Vista、Windows 7、Windows Server 2003/2008，Q（Home Server）也可以胜任。

本书使用的主要演示工具是 Microsoft Visual C# 2010 Express Edition（微软称 Express Edition 为速成版，可以到微软的官网上免费下载和免费注册）、Microsoft Visual Studio Ultimate 2010 和 SharpDevelop（非常好用的开源 IDE，现已支持最新的 .NET Framework 4.0。下载安装的链接为 http://www.sharpdevelop.com/OpenSource/SD/Default.aspx）。这三个工具各有特点，Express 版提供绝大部分开发所需要的基本功能（通过 TestDriven.NET 第三方工具，集成单元测试功能），并和 Ultimate 版有类似的使用界面，以方便用户从 Express 版升级到商业版。SharpDevelop 集成开源的单元测试工具 NUnit、代码管理工具 subversion、Code Coverage，并支持托管代码反汇编工具 "Red Gate's .NET Reflector"（读者可以通过下面的网站链接下载此工具：

http://www.red-gate.com/products/reflector/)，还支持 Ultimate 版才提供的类图功能、FxCop 和 StyleCop。SharpDevelop 的操作和 Ultimate、Express 版差别不大。本书的范例大部分都可以无缝地被这三个工具打开、编辑、编译和运行。

本书约定

- 凡在书中写参阅 MSDN 的"XXX"主题的表示读者可以用双引号的内容（不包含双引号）在 MSDN 中查询。
- 本书中除特殊声明外，IDE（Integrated Development Environment）指的是微软的集成开发环境，包括 VSTS 2008、Express 版、VS 2010 Ultimate。
- 本书中出现类似这样的内容："视图"（View）→"类视图"（Class View），表示先选中"视图"（View）菜单，然后选中"类视图"（Class View）菜单。引号中的内容表示中文版显示的内容，括号中的内容表示英文版显示的对应内容。
- 本书中，"Tips"表示一些小技巧；"小知识"介绍一些专业词汇或者常识。
- 本书的范例大部分都是没有注释的。因为解释都在书中表述过，为了内容紧凑就不再重复。而且，本书范例代码都是从"零"行开始的。
- 本书会直接使用某些技术术语，如 Handle（句柄）。
- 类成员变量被称为类成员字段（field）。
- 本书把"泛型"作为一种编程思想处理。为了区别，C#对泛型的支持被称为 C#对模板的支持。

不包含的内容

这里无法一一列举不包含的内容，只能提到部分大方向的内容。本书不介绍 ASP.NET、ADO.NET、WCF、WF 和 Profile API，这些待有机会时另外出书，分别进行介绍。一些内容只做了基本的介绍或者小范围的举例介绍，如 WinForm、WPF、设计模式，还有不少内容因为篇幅的问题只做初级介绍，不再列举。希望读者可以根据这里的介绍更好地选择自己需要的书籍。本书包含所有 .NET Framework 4.0 的新特性，选择性地介绍其中一些内容，比较重要的有 .NET 并行扩展（.NET Parallel Extensions）、动态语言运行时（Dynamic Language Run-Time），还有一些 .NET Framework 4.0 的重要修订。有兴趣的读者可以在 MSDN 中查阅其他的更新部分。

如何使用本书

本书针对因工作或其他需要使用 C#（.NET Framework）来完成软件工程的人群，也可作为大中专院校的辅助读物。本书假设读者正在学习或者学过一门编程语言，即对赋值和编程逻辑（条件转移[Conditional Logical]，循环[Iteration]等）有一定的认识。

本书提供所有源代码，请从网站（www.ncpress.com.cn）下载，或致电 010-64865699-8034/8067，也可以发送 E-mail 至 bookservice@126.com 获取。

CONTENTS

目录

PART 01 工程实战

PART

01

工程实战

引

近年来软件技术的广度、深度和集成度在飞速发展。最近 4 年中，微软连续升级 4 次。.NET Framework 这个基础框架就是其中一个的小小例证，同时也证明了竞争是多么的激烈。如今，.NET Framework 不再是“吴下阿蒙”，已开始适应企业级的应用需求。技术飞速的增长和知识量也越来越庞大，让个人在有限的时间内全部掌握知识再开始工作变得越来越不现实。因此，正确分析问题、分解问题，依据问题去查找、搜索现有的解决方案、技术要点，并迅速学习、运用到自己的方案中尤为关键。在实践中发现不少同事，甚至一些有经验的专业人士在这方面也有所缺憾。就像电影“I，Robot”中的一句经典语句“You must ask the right questions”正确的问题往往能迅速打开答案之门。下面就让我们和 Mojo 一起面对他遇到的实际困难与挑战吧！

.NET

Chapter 01
工程开始（Project Kickoff）

在实际的工程中从来不存在“HelloWorld”这类温室的花朵（范例），Mojo 一开始就遇到很多初学者都觉得很难的 Windows API。让我们看看他是如何查找、搜寻资料，并解决他遇到的这些纸老虎吧！

1.1 一个工作上的小问题

Mojo 和很多人一样刚刚毕业就进入一家软件公司。公司规定在上班时间不得用 QQ、WLM、AIM 等即时聊天工具（Instant Message，IM）。但是 Mojo 所在的工程小组的成员又都坐得比较远，有时沟通不是很方便。Mojo 是个实干派，打算自己动手写个简单即时聊天工具。工程有了，但是从哪里开始？ Mojo 刚毕业，毫无经验。既然懂得很少，那就先从懂的地方开始吧！

1.2 问题的快速分析

Mojo 虽然不知道技术上如何实现，但是起码知道这个工具需要具备什么功能。Mojo 在公司内悄悄地侦查了一下，发现公司绝大部分的机器运行微软的各种操作系统，从 Windows XP、Windows 2000 到 Windows Vista、Windows Server 2008 甚至 Windows 7 都有。也难怪，公司的产品就要求要运行在各种微软平台上。Mojo 随手写下这些功能要求。

- 能够在局域网内发送消息。
- 能够列举局域网内的其他消息接收者。

- 能够群发消息。
- 不需要接收方也运行此软件。
- 保存收发的信息。

除此之外，Mojo 还列出一些非功能要求。

- 能够在 Windows XP、Windows Vista 和 Windows 7 上使用。
- 用最新的 C#和 .NET Framework 技术来编写所有的代码，顺便练习。
- 尽量借助已有的模块或者系统功能，做到快速实现。
- 程序要有一定的维护性和扩展性。

1.3 关键技术调研

1.3.1 查找已存在的方案

既然要用微软的技术，那么首先去微软的官方网站上找找灵感。Mojo 打开 Firefox 浏览器，输入 http://msdn.microsoft.com，然后按回车键。MSDN 是一个浩瀚的 MS 知识海洋，想逐条翻看是不太现实的，因为这个知识库每天都有成千上万的人在贡献他们的知识。所以，用搜索是最明智的方法。现在关键是输入什么来搜索，想想上面列举的条目，Mojo 输入“intranet send message C# networking”（局域网 发送 消息 C# 网络）。不过很让 Mojo 失望，翻了好几页也没有什么有价值的条目。Mojo 觉得 MSDN 在跟他开玩笑。于是，他打开 http://www.google.com，继续输入刚才的关键字。结果在第二个条目（笔者无法保证永远都出现在第二条）看到如下内容。

```
Send Messages Using The Net Send Command | C# Download | www ...
1 Apr 2007 ... 5, Send Messages Using The Net Send Command - C#. This sample program
demonstrates how to send messages in a network using the Windows ...
www.cy2online.net/Downloads.php?Language=CSharp... - Cached - Similar
```

看起来很像那么回事。顺着这个链接，最终 Mojo 下载到一个 C#例程。我们不用关心这个例程是怎么写的。关键是这个例程其实仅仅是利用 Windows 系统的一个 Shell 上运行的网络命令“Net Send”。既然这个命令是系统实现的，那么应该有对应的编程接口。于是把字符串改为“Net send message c# Win32 API intranet”继续搜索。查到在 www.CodeProject.com 有个“Batch Net Send”的工程。同样的，Mojo 从中找到一个关键的 Windows 系统 API（Application Programming Interface，应用程序编程接口）——NetMessageBufferSend。不过怎么用还是个问题，这时还要去 MSDN 上搜。在“Win32 and Com Development”根节点下找到相关说明。但是在 C#中怎么用呢？Mojo 现在毫不犹豫地用“NetMessageBufferSend C#”在 Google 里搜索，得到 C#调用的函数原型（大家可能会看着很晕，后面的章节会继续解释，还可以参考 9.2 节“平台调用服务”）。

```
[DllImport("netapi32.dll", CharSet = CharSet.Unicode)]
static extern NetReturnCode NetMessageBufferSend(
    string serverName,
    string msgName,
    string fromName,
    string buffer,
    int bufferLength);
```

Tips

搜索已有的方案往往是我们开始任何工程，甚至是任何一段完整功能的第一步。Google 是技术类搜索暂时的领头羊。先从 Google 开始，然后可以尝试 bing、Baidu，如果是微软的技术，最后可以试试 MSDN。也可以去一些代码网站，如 CodeProject、CodeGuru、GotDotNet 或者 CSDN 论坛。

1.3.2 动手写第一个程序（第一个原型）

这里不会详细介绍工具的所有功能，只是简单介绍用到的步骤。在启动 Express 版的 IDE（集成开发环境）后，选择“File”（文件）菜单中的“New Project”（新建工程）命令，如图 1-1 所示，或者按快捷键“Ctrl+Shift+N”（这和 Ultimate 版的快捷键是一样的）。

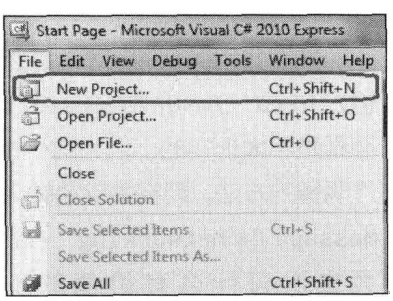

图 1-1

弹出“New Project”（新建工程）对话框，如图 1-2 所示。选择“Console Application”（控制台应用程序）选项，并且把工程名字改为合适的名称（例如“FirstPrototype”），然后单击“OK”（确定）按钮完成工程的新建。

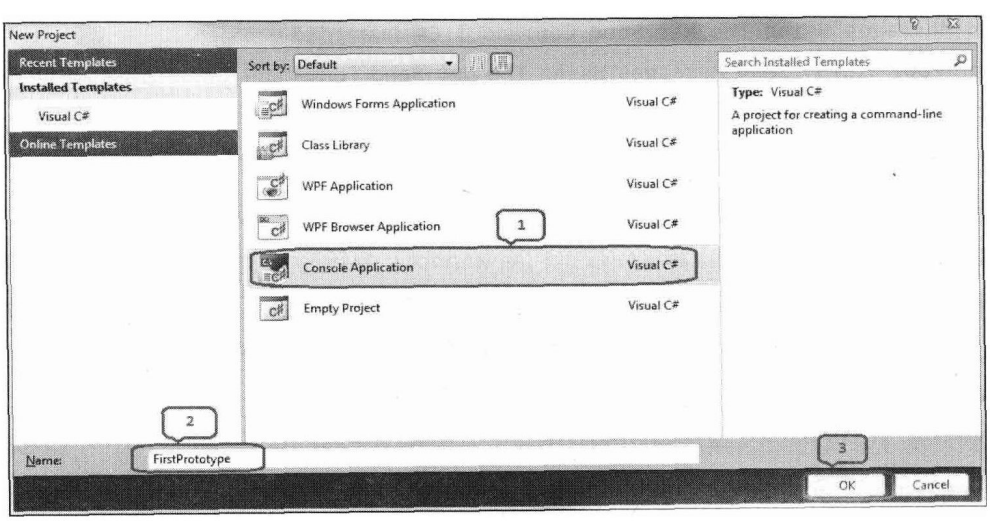

图 1-2

双击“Solutions Explorer”（工程管理器）中的 Program.cs 文件，打开后可以看到向导设定的最基本代码：