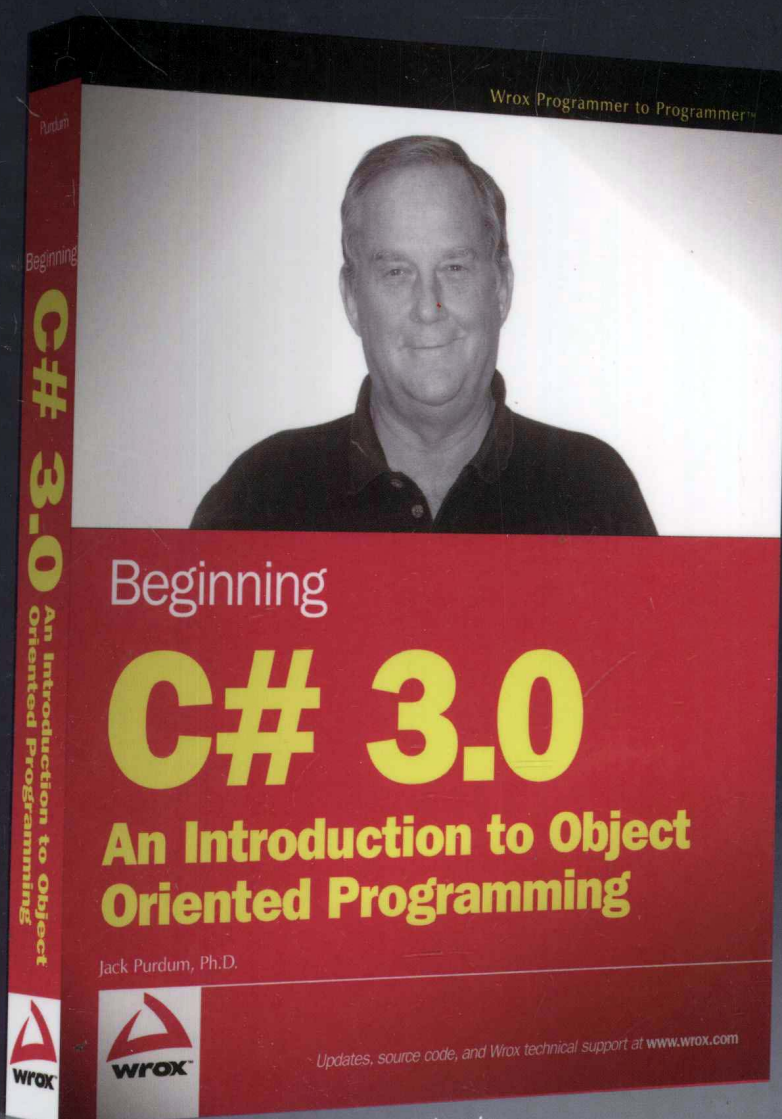


C# 3.0

面向对象编程

(美) Jack Purdum 著 叶雄兵 黄谦 译



Updates, source code, and Wrox technical support at www.wrox.com



Beginning C# 3.0: An Introduction to
Object Oriented Programming

清华大学出版社

国外计算机科学经典教材

TP31
P772

000

C# 3.0 面向对象编程

(美) Jack Purdum 著

叶雄兵 黄谦 译

TP31/2C
P772

清华大学出版社

北 京

Jack Purdum

Beginning C# 3.0: An Introduction to Object Oriented Programming

EISBN: 978-0-470-26129-3

Copyright © 2007 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2009-2125

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

C# 3.0 面向对象编程/(美)普德姆(Purdum, J.)著; 叶雄兵, 黄谦 译. —北京: 清华大学出版社, 2009.9
(国外计算机科学经典教材)

书名原文: Beginning C# 3.0: An Introduction to Object Oriented Programming

ISBN 978-7-302-20953-9

I .C... II. ①普... ②叶... ③黄... III. C 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(2009)第 160375 号

责任编辑: 王 军 于 平

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 何 芊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京鑫丰华彩印有限公司

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 28.75 字 数: 700 千字

版 次: 2009 年 9 月第 1 版 印 次: 2009 年 9 月第 1 次印刷

印 数: 1~4000

定 价: 58.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 030094-01

出版说明

近年来,我国的高等教育特别是计算机学科教育,进行了一系列大的调整 and 改革,亟需一批门类齐全、具有国际先进水平的计算机经典教材,以适应我国当前计算机科学的教學需要。通过使用国外优秀的计算机科学经典教材,可以了解并吸收国际先进的教学思想和教学方法,使我国的计算机科学教育能够跟上国际计算机教育发展的步伐,从而培养出更多具有国际水准的计算机专业人才,增强我国计算机产业的核心竞争力。为此,我们从国外多家知名的出版机构 Pearson、McGraw-Hill、John Wiley & Sons、Springer、Cengage Learning 等精选、引进了这套“国外计算机科学经典教材”。

作为世界级的图书出版机构, Pearson、McGraw-Hill、John Wiley & Sons、Springer、Cengage Learning 通过与世界级的计算机教育大师携手,每年都为全球的计算机高等教育奉献大量的优秀教材。清华大学出版社和这些世界知名的出版机构长期保持着紧密友好的合作关系,这次引进的“国外计算机科学经典教材”便全是出自上述这些出版机构。同时,为了组织该套教材的出版,我们在国内聘请了一批知名的专家和教授,成立了专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动,各位委员根据国内外高等院校计算机科学及相关专业的现有课程体系,并结合各个专业的培养方向,从上述这些出版机构出版的计算机系列教材中精心挑选针对性强的题材,以保证该套教材的优秀性和领先性,避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量,我们为这套教材配备了一批经验丰富的编辑、排版、校对人员,制定了更加严格的出版流程。本套教材的译者,全部由对应专业的高校教师或拥有相关经验的 IT 专家担任。每本教材的责编在翻译伊始,就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华,在经过翻译、排版和传统的三审三校之后,我们还请编审委员或相关的专家教授对文稿进行审读,以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限,该套教材在出版过程中很可能还存在一些遗憾,欢迎广大师生来电来信批评指正。同时,也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材,共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

前言

在过去 25 年中，我编写了 15 本书籍，这些书籍涉及各种编程主题。为什么要编写如此多的编程书籍？是否是一开始写得不好？事实并非如此。在 25 年前编写第一本书籍时，面向对象编程(OOP)这一概念还很不明朗，大多数人都不了解它。与其他人一样，我花费不少精力研究 OOP 后才了解到 OOP 的真谛。并且，我知道很多人仍然对 OOP 缺乏了解。

每次上编程课时，我都会尽量采用更新更好的方式来讲解课程内容。如果环视整个教室并发现同学们都静静地坐着，那么我就会自我反省，并且尝试用其他方法介绍正在讲述的概念。我教授的每一堂课都会对教与学提出新的观点。我常常会更换教学示例和讲解编程概念的技巧，我在多年后又回来教书的原因之一就是我怀念那段不断摸索学习的经历。

虽然我编写的一些书籍是关于其他语言而非 C# 语言，但是这些书籍同样丰富了我作为教师的经验。只有在掌握一些不尽完美的语言之后，才能体会到某种语言的优点和弱点。编程语言不断地发展这一事实说明了如下观点：我肯定不是唯一仍然在学习的人。目前绝对可以确信的一件事情是：今天学到的任何知识在将来都有可能过时。

也许您应该询问自己的真正问题是，我为什么要购买这本书籍，而不是购买其他人编写的书籍？购买我所编写书籍的原因肯定有很多，但是真正使我编写的书籍在众多同类书籍中脱颖而出的原因只有一个。我绝不怀疑一些程序员可以编写更好的代码。当拥有自己的软件公司时，我最优秀的程序员 Tim 编写的代码就比我好，他是一名非常出色的程序员。但是，如果要让 Tim 说明这些代码，他就会手足无措。

本书不同于其他书籍：我尝试使用各种技术、示例和方法来向数千名学生介绍各种编程概念。我知道什么方法有效，什么方法无效。我遇到过很多令人头痛的问题，并且将它们改造成能够帮助学生在课堂上领悟所学知识的示例。毫无疑问，虽然如今有诸多作者、出色的编码人员，但是他们没有遭遇过令人头疼的示例，所以教导的方法完全不起作用。适合您的方法未必适合其他人。会编写优秀的示例不等于会编写优秀的书籍。有些人具备这两方面的能力(我认为 P.J. Plauger、Jon Bentley、Brian Kernighan 和 Steve McConnell 都是这样的人)，但是这样的人相当少。可以看出，我并没有将自己放到这个列表中。然而，我的教学经验弥补了我在编码技巧方面的欠缺。当然，这需要由读者来判断。

本书内容丰富、示例清晰、叙述风格活泼。我想您会喜欢本书的。然而，更重要的是通过阅读本书，您可以了解面向对象编程和 C# 能完成的所有工作。

本书读者对象

本书不要求读者有编程经验。然而，这并不表示本书十分简单。本书的介绍进度非常缓慢，在介绍下一个主题之前会先帮助读者打下牢固的基础。本书鼓励读者编写自己的程序，并完成每章末尾的习题。如果试图走捷径，实际上就是自欺欺人。读者应该亲自输入本书的每行代码，并用这些代码进行试验。这样做可以使 C# 语言的学习事半功倍，也更容易使自己理解书中介绍的概念。可以从 Wrox 的 Web 站点下载代码，在后面将提供具体下载方式。可以只通过编写程序来学习编程，我鼓励您只要有机会就这样操作。

如果您有一些编程经验，很可能更容易读懂本书。并在阅读本书的过程中学到一些知识。如果您以前从来没有编写过程序，则非常适合于学习本书，因为这样就不会将太多的不良习惯带到新语言的学习中。

我认为本书非常适合于已经掌握一种编程语言，然而是在面向对象编程技术出现之前学习该语言的人。我想有很多像我这样的“老资格”的人愿意让自己接受 OOP 技术的再教育，并充分享受 OOP 带来的好处。我相信本书非常适合于这种需要。

与其他同类书籍不同的是，如果您已经掌握一种编程语言，或者以前有过其他方面的编程经验，我建议阅读本书时不要采用跳越章节的方式阅读。如果您熟悉符号表、lvalues、rvalues 和 Bucket Analogy，那当然很好——但是从不同的角度再读一遍也不无益处。记住，学习编程语言没有捷径。

本书内容简介

本书使用了一种非编程语言的方式来介绍面向对象编程。相应章节的目标是先提出一些概念，然后平滑地过渡到在 C# 中处理对象的方式。

全书将使用常见且容易理解的示例来介绍新的编程概念。其中涵盖了大部分 C# 编程语言的主题，然而仍然有少量相对不常见的主题没有包括进来。我觉得这些主题对手边的任务(即使用 C# 教导 OOP)并不重要。当读者学习完本书后，应该能够编写自己设计的复杂 OOP 程序。

本书组织结构

本书的章节顺序就是我向新生授课的顺序。这种顺序具有逻辑意义，每一章构建在前面几章包含信息的基础上。虽然读者可以跳过一些章节，但我认为这样对学习无益。我有自己的解释概念的方式，有些方式与众不同，但它们确实有用。

每一章最后都有几道习题。希望读者在学习下一章之前先完成这些习题。前面提到过，可以只通过编写程序来学习编程。只是阅读容易使读者获得一种虚假的安全感，以为自己学会了。其实也许并没有学会。编写自己的程序是真正学会编程的唯一途径。

使用本书所需的软硬件

本书的所有示例都是用 Visual Studio 2008 的 C# Express 版本编写的。第 1 章说明了如何从 Microsoft 免费下载和安装 C# Express。C# Express 是一款非常出色的软件，几乎具有完整 Visual Studio 的大部分功能。虽然这个 C# Express 版本缺少一些优秀工具，但是用来练习使用 Visual Studio 和 C# 提供的大部分功能的工具已经足够。当您用了 C# 一段时间以后，可能会购买完整的 Visual Studio 包。到时候您会发现那样做是值得的。

应该使用基于 Pentium 的系统(虽然竞争者也存在，比如 Mac 和 Linux，而且它们似乎也能很好地工作)，至少需要 1GB 内存和 1GB(或更多)硬盘空间。满足这些需求以后，就可以使用您的系统开始学习本书。

源代码

在读者学习本书中的示例时，可以手工输入所有的代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 <http://www.wrox.com/> 或 <http://www.tupwk.com.cn/download> 上下载。登录到站点 <http://www.wrox.com/>，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有的源代码。

注释：

由于许多图书的标题都很类似，所以按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-0-470-26129-3。

在下载了代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但错误总是难免的。如果您在本书中找到了错误(例如拼写错误或代码错误)，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫。当然，这还有助于我们提供更高质量的信息。

请给 wkservice@vip.163.com 发电子邮件，我们就会检查您的反馈信息，如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书英文版的勘误表，可以登录 <http://www.wrox.com>，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已确认并提交的所有勘误项。完整的图书列表还包括每本书的勘误

表, 网址是 <http://www.wrox.com/misc-pages/booklist.shtml>。

P2P.WROX.COM

如果要与作者和同行讨论, 请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统, 便于您发布与 Wrox 图书相关的消息和相关技术, 与其他读者和技术用户交流心得。该论坛提供了订阅功能, 当论坛上有新的消息时, 它可以给您发送感兴趣的话题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 p2p.wrox.com 上, 有许多不同的论坛, 它们不仅有助于阅读本书, 还有助于开发自己的应用程序。要加入论坛, 可以遵循下面的步骤:

- (1) 进入 p2p.wrox.com, 单击 Register 链接。
- (2) 阅读使用协议, 并单击 Agree 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息, 单击 Submit 按钮。
- (4) 您会收到一封电子邮件, 其中的信息描述了如何验证账户, 完成加入过程。

注释:

不加入 P2P 也可以阅读论坛上的消息, 但如果要发布自己的消息, 就必须加入该论坛。

加入论坛后, 就可以发布新消息, 回复其他用户发布的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息, 可以单击论坛列表中该论坛名旁边的 Subscribe to this Forum 图标。

关于使用 Wrox P2P 的更多信息, 可阅读 P2P FAQ, 以了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ, 可以在任意 P2P 页面上单击 FAQ 链接。

目 录

第 I 部分 C# 入门

第 1 章 安装 C#	3
1.1 面向对象编程(OOP)简史	4
1.2 安装 C#	4
1.2.1 下载 C# Express	5
1.2.2 安装 C# Express	5
1.3 使用 C# Express 编写测试程序	7
1.3.1 创建项目	7
1.3.2 C#集成开发环境	8
1.3.3 向 Windows 窗体中添加对象	10
1.3.4 运行程序	13
1.4 小结	13
第 2 章 对象	15
2.1 对象	15
2.1.1 对象在日常生活中的使用	16
2.1.2 使用的属性和方法的数量	18
2.1.3 定义类之后的工作	18
2.2 开发程序计划	25
2.3 使用对象创建简单的应用程序	27
2.4 使用 C#实现程序计划	29
2.4.1 向项目中添加程序引用	30
2.4.2 向项目中添加新条目	31
2.4.3 设置项目属性	33
2.4.4 向 frmMain 对象中添加对象	34
2.4.5 为处理步骤添加程序代码	36
2.4.6 构造用户界面对象	37

2.4.7 btnDisplayOutput 单击事件 代码	38
2.4.8 分析 btnDisplayOutput 单击事件代码	39
2.5 小结	40

第 II 部分 C# 语法

第 3 章 数据类型	43
3.1 整型数据类型	43
3.1.1 整型数据类型的取值范围	44
3.1.2 使用各种整数的场合	45
3.1.3 创建新项目的步骤序列	47
3.1.4 设计用户界面	47
3.1.5 处理步骤的代码	47
3.1.6 变量命名规则和约定	49
3.1.7 使用 TryParse()方法	49
3.1.8 浮点数据类型	52
3.1.9 浮点的精度	53
3.1.10 使用各种浮点数据类型 的场合	54
3.2 货币值: decimal 数据类型	55
3.3 Boolean 数据类型	57
3.4 小结	58
3.5 习题	59
第 4 章 理解 C#语句	61
4.1 编程语言的基本构件块	61

4.1.1 操作数和运算符	62	5.5 习题	111
4.1.2 表达式	62	第 6 章 在代码中制定决策	113
4.1.3 语句	63	6.1 关系运算符	113
4.1.4 运算符优先级	64	6.1.1 使用关系运算符——if 语句	114
4.1.5 重写默认优先级次序	65	6.1.2 if-else 语句	118
4.2 定义变量	65	6.1.3 简单 if-else 的简化版本： 三元运算符	119
4.3 在程序中使用变量	68	6.1.4 if 和 if-else 语句的风格 考虑事项	119
4.4 Bucket Analogy	69	6.1.5 嵌套的 if 语句	121
4.5 程序错误的类型	71	6.1.6 RDC	122
4.5.1 语法错误	71	6.1.7 层叠 if 语句	123
4.5.2 语义错误	71	6.2 逻辑运算符	124
4.5.3 逻辑错误	71	6.3 switch 语句	126
4.6 Visual Studio 调试器	72	6.4 小结	128
4.6.1 调试过程	72	6.5 习题	129
4.6.2 简化重复的调试工作	73	第 7 章 使用循环重复语句	131
4.6.3 使用调试器	73	7.1 程序循环	131
4.7 防御编码	76	7.1.1 良性循环和恶性循环	132
4.7.1 使用程序注释	77	7.1.2 for 循环	132
4.7.2 使用有意义的变量名	78	7.1.3 for 循环的嵌套	139
4.7.3 避免使用幻数	78	7.1.4 while 循环	141
4.8 小结	80	7.1.5 采用多种类型循环的原因	143
4.9 习题	80	7.2 do-while 程序循环	143
第 5 章 引用数据类型	81	7.3 continue 语句	147
5.1 字符串变量	81	7.4 小结	148
5.1.1 定义字符串引用变量	82	7.5 习题	148
5.1.2 引用类型不同于值类型 的原因	84	第 8 章 数组	151
5.1.3 引用变量规则	85	8.1 数组的定义	151
5.1.4 引用类型变量与值类型变量	86	8.1.1 部分数组细节	153
5.1.5 引用变量采用这种操作方式 的原因	86	8.1.2 Listview 对象	158
5.1.6 按值传递与按引用传递	87	8.2 数组是对象	161
5.1.7 提高效率	87	8.3 多维度组	163
5.1.8 使用字符串变量	88	8.4 初始化数组	166
5.1.9 字符串操作	89	8.4.1 初始化数组的变体	167
5.2 字符串与转义序列	100	8.4.2 初始化多维度组	168
5.3 DateTime 引用对象	101	8.4.3 对象的初始化列表	168
5.4 小结	111		

8.5 集合	170	10.7 习题	238
8.6 ArrayList 对象	172	第 11 章 异常处理与调试	239
8.7 小结	176	11.1 概述	239
8.8 习题	176	11.2 程序错误	241
第 III 部分 编写自己的类		11.2.1 数据有效性验证	241
第 9 章 设计类	179	11.2.2 限制用户输入	242
9.1 类设计	180	11.3 异常处理	247
9.1.1 向项目中添加类	180	11.4 程序调试	253
9.1.2 作用域	181	11.4.1 自然法则	253
9.1.3 形象化作用域	184	11.4.2 Visual Studio 调试器	255
9.1.4 C# 支持作用域的原因	185	11.4.3 单步调试程序	258
9.2 设计程序	186	11.4.4 脚手架代码	259
9.2.1 五步程序步骤(Five Program Steps)	187	11.4.5 防御性编码	260
9.2.2 不要只见树木不见森林	189	11.5 小结	261
9.3 UML Light	189	11.6 习题	261
9.3.1 访问说明符	190	第 12 章 泛型	263
9.3.2 UML 方法	192	12.1 泛型的概念	263
9.3.3 像用户一样思考	194	12.1.1 递归	268
9.4 clsDates 设计	195	12.1.2 数据问题	269
9.4.1 名称空间修饰符	197	12.2 泛型的定义	269
9.4.2 类的组织	197	12.2.1 泛型与 ArrayList	270
9.5 用户界面的不同含义	201	12.2.2 使用泛型类	277
9.6 小结	201	12.3 泛型 Quicksort	278
9.7 习题	201	12.3.1 使用带约束和接口 的泛型	282
第 10 章 设计与编写自定义类	203	12.3.2 接口	282
10.1 构造函数	203	12.3.3 使用接口的原因	283
10.1.1 默认构造函数	203	12.3.4 使用接口	284
10.1.2 非默认构造函数	204	12.4 小结	288
10.1.3 构造函数重载	204	12.5 习题	288
10.2 属性方法	207	第 IV 部分 存储数据	
10.3 属性方法中发生错误后 的对策	211	第 13 章 使用磁盘数据文件	291
10.4 扑克牌程序的类设计	214	13.1 目录	291
10.5 设计使用 clsCardDeck 的 纸牌游戏	224	13.1.1 DriveInfo 类	292
10.6 小结	237	13.1.2 Directory 类	292
		13.1.3 DirectoryInfo 类	293
		13.2 File 名称空间	298

13.3	FileInfo 类	299	14.3.5	创建新数据库	364
13.4	文件的类型	300	14.3.6	创建新表 (CREATE TABLE)	366
13.4.1	文本与二进制数据文件	300	14.3.7	向表中添加记录(INSERT INTO)	371
13.4.2	读取数据	309	14.3.8	编辑记录(UPDATE)	374
13.5	连续文件与随机访问文件	310	14.3.9	删除记录(DELETE)	378
13.5.1	连续文件	310	14.4	数据库查询	379
13.5.2	随机访问文件	311	14.4.1	将 DataGridView 对象 绑定到数据库	381
13.5.3	frmMain	315	14.4.2	使用不带数据绑定的 DataGridView 对象	383
13.5.4	导航记录	322	14.5	使用 LINQ	388
13.5.5	删除记录	323	14.5.1	LINQ 查询关键字	388
13.5.6	clsRandomAccess	324	14.5.2	LINQ 的名称空间与引用	391
13.6	串行化与反串行化	341	14.6	小结	397
13.7	MDI、菜单和文件对话框	348	14.7	习题	397
13.7.1	添加菜单	348	第 15 章	继承与多态	399
13.7.2	添加 File Open 对话框	349	15.1	继承的概念	399
13.7.3	调用另一个窗体	351	15.1.1	继承示例	400
13.8	小结	352	15.1.2	基类不是派生类	414
13.9	习题	353	15.1.3	抽象类	414
第 14 章	使用数据库	355	15.2	多态	415
14.1	数据库的概念	355	15.3	小结	417
14.1.1	数据库的结构	356	15.4	习题	418
14.1.2	设计数据库表的字段	357	附录	习题答案	419
14.2	使用自己的数据库	359			
14.3	使用 SQL	361			
14.3.1	SELECT 语句	361			
14.3.2	WHERE 谓词	362			
14.3.3	ORDER BY 子句	362			
14.3.4	聚合	363			



1.1 面向对象编程(OOP)简史

许多人认为 OOP 产生于 20 世纪 80 年代,其产生过程是 Bjarne Stroustrup 通过创建 C++ 语言将 C 语言带入面向对象领域。实际上,最早的两种面向对象编程语言是 SIMULA 1(1962 年)和 Simula 67(1967 年)。Simula 语言由 Ole-John Dahl 和 Kristen Nygaard 在挪威首都奥斯陆的“挪威计算中心”创建。虽然较早的 Simula 语言已经具有了 OOP 的大部分优点,但直到 C++ 在 20 世纪 90 年代的普及, OOP 才得到了蓬勃发展。

C 语言是 C++ 的父语言。人们经常说 C 语言的功能强大到可以独立完成开发工作,而 C++ 语言则在此基础上更进一步,具备更为强大的功能。大多数开发人员承认 C++ 语言是一种功能非常强大的语言,如今该语言仍然被广泛使用。然而,它的这种强大功能也带来了大量复杂性。语言开发人员希望使用一种更为简单而且简化的语言进行 OOP 开发。

OOP 发展的下一步始于 1991 年 1 月,当时 James Gosling、Bill Joy、Patrick Naughton、Mike Sheridan 与科罗拉多州阿斯彭市遇到的其他几位开发人员正在讨论 Stealth Project 的思想。这个小组希望开发一种能够集中进行控制并通过手持设备进行编程的智能电子设备。他们认为 OOP 是开发语言的正确发展方向,C++ 没有能力承担这一工作。他们的研究成果就是 Oak 编程语言,该语言最终演变成 Java 语言(必须将 Oak 改成其他名称,因为这个小队发现已经有一种语言采用了该名称)。

由于 Internet 发展的驱动,Java 迅速变成一种普及语言。Web 和 Java 的迅速发展原因是:在 Web 上运行 Java 程序所需的“组件”很快成为各种 Web 浏览器的必备组件。由于 Java 对 Web 功能的改进,Web 得到了飞速发展。

对于许多程序员来说,C# 是微软应对 Java 的产品。有人甚至认为 C# 是微软拒绝推动非自主开发语言的结果。这种说法过于严厉,微软有许多很好的理由开发 C#,其中一个(不是最重要的)理由是,它需要开发在托管环境中运行的类型安全程序。您可能无法立刻理解这种理由的含义,但是在学习 C# 的过程中可以逐渐了解这一点。

C# 提供了一种健壮的面向对象编程语言,以及一组令人印象深刻的、能够处理几乎所有编程任务的工具。无论是开发桌面、分布式、Web 还是移动应用程序,C# 都能够胜任。

随着对 C# 越来越熟悉,您会喜欢该语言相对较少的关键字、清晰的语法及易用的开发环境。您会发现使用 C# 编写的程序片断可以在其他程序中重用。最后,您可能会欣喜地发现,掌握 C# 语言的程序员有更多的就业机会。

1.2 安装 C#

如果已经购买并安装了 Visual Studio 2008 和 C#,就可以跳过本节内容。如果还没有安装 C#,本节将介绍如何下载和安装 Visual Studio 的 C# Express 版本。C# Express 是微软提供的免费 C# 修改版。虽然 C# Express 版本比 Visual Studio 的商业版本缺少一些功能,但是使用 C# Express 能够编译和运行本书提供的所有示例程序。当您自信能够使用 C# 进行所有开发工作之后,就可以购买 Visual Studio 的完整版。

1.2.1 下载 C# Express

在编写本书时,可以从网址 <http://msdn2.microsoft.com/en-us/express/future/bb421473.aspx> 处下载 C# Express, 相应的 Web 页面如图 1-1 所示, 本书采用 Visual C# Express Edition Beta 2 编写(也许当您阅读本书时标题中的 Beta 2 部分已经改变)。现在单击 IMG 文件链接就可以下载文件。

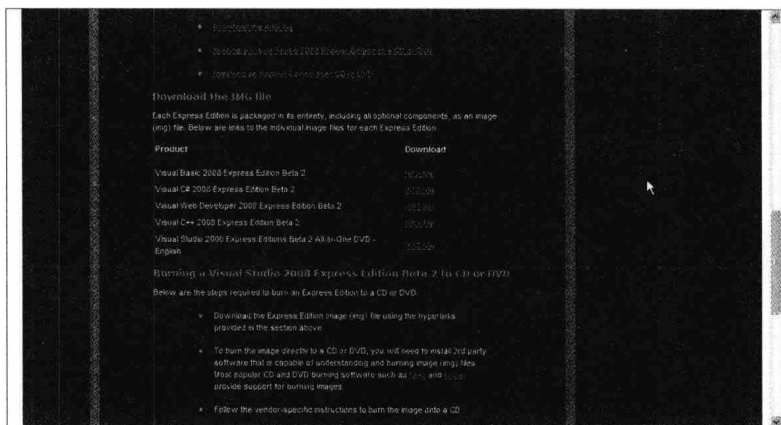


图 1-1

根据 Internet 连接速度, 需要几分钟将文件保存到您的系统上。

1.2.2 安装 C# Express

下载完成后, 单击提供的可执行文件(该文件名为 `vcsetup.exe`, 但是这个名称可能会有所不同), 可以看到如图 1-2 所示的界面, 即 C# Express Edition 安装界面。

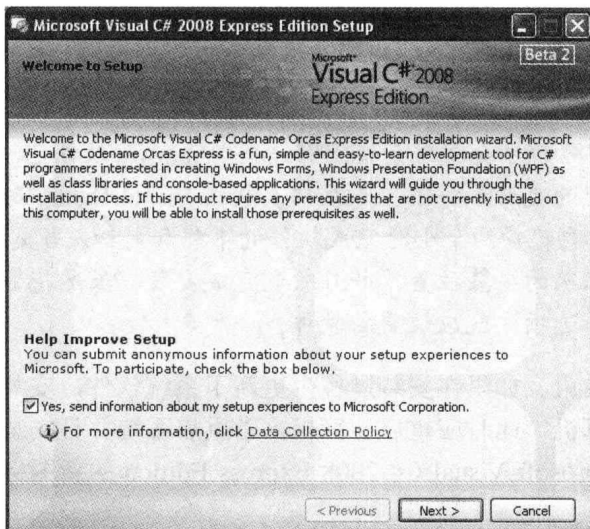


图 1-2