

国外计算机科学教材系列

PEARSON

C++ 程序员教程

C++ for Programmers

包含240个实例程序

指针
模板
vector
异常处理
运算符重载
C++标准类库

集成的OOP案例研究
Boost类库和C++的未来
GNU和Visual C++调试器
OOD/UML 2 ATM案例研究
面向对象编程：类、对象、继承、多态
标准模板类库 (STL)：容器、迭代器和算法

Paul J. Deitel
[美]
Harvey M. Deitel

著

张良华 吴明飞 胡强 等译



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

国外计算机科学教材系列

C++ 程序员教程

C++ for Programmers

Paul J. Deitel

[美]

Deitel & Associates, Inc.

著

Harvey M. Deitel

Deitel & Associates, Inc.

张良华 吴明飞 胡 强 等译

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是一本 C++ 编程方面的优秀图书,全面介绍了 C++ 编程的原理和方法,涉及类与对象、控制语句、函数与递归、数组与类模板 vector、指针与基于指针的字符串、类、运算符重载、继承、多态、模板、流输入/输出、异常处理、文件处理、字符串流处理、STL、Boost 类库、TR1、C++ 0x 等,可以说本书是学习 C++ 编程语言最好的教程。附录中还分别给出了完整的 ATM 案例研究的代码以及 GNU、Visual Studio 调试器的用法。书中 240 个活代码例子,对学习 C++ 非常有帮助。

本书适合软件设计人员学习 C++ 编程,也可作为高等院校相关专业的编程语言教材。

Authorized translation from the English language edition, entitled C++ for Programmers, ISBN: 9780137001309 by Paul J. Deitel and Harvey M. Deitel, published by Pearson Education, Inc., publishing as Prentice Hall, Copyright © 2009 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY, Copyright © 2010.

本书简体中文版由 Pearson Education 培生教育出版集团授予电子工业出版社,未经许可,不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号:图字:01-2009-3827

图书在版编目(CIP)数据

C++ 程序员教程/(美)戴特尔(Deitel, P. J.), (美)戴特尔(Deitel, H. M.)著;张良华等译.

北京:电子工业出版社,2010.1

(国外计算机科学教材系列)

书名原文:C++ for Programmers

ISBN 978-7-121-09905-2

I. C… II. ①戴…②戴…③张… III. C 语言-程序设计-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 211146 号

策划编辑:许菊芳

责任编辑:许菊芳

印 刷:北京市天竺颖华印刷厂

装 订:三河市鑫金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:47.5 字数:1505 千字

印 次:2010 年 1 月第 1 次印刷

定 价:85.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社 发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zlt@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

C++ FOR PROGRAMMERS

DEITEL DEVELOPER SERIES

Paul J. Deitel
Deitel & Associates, Inc.

Harvey M. Deitel
Deitel & Associates, Inc.



Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

*In memory of Joseph Weizenbaum
MIT Professor Emeritus of Computer Science:
For making us think.*

Paul and Harvey Deitel

Deitel Resource Centers

Our Resource Centers focus on the vast amounts of free content available online. Find resources, downloads, tutorials, documentation, books, e-books, journals, articles, blogs, RSS feeds and more on many of today's hottest programming and technology topics. For the most up-to-date list of our Resource Centers, visit:

www.deitel.com/ResourceCenters.html

Let us know what other Resource Centers you'd like to see! Also, please register for the free *Deitel® Buzz Online* e-mail newsletter at:

www.deitel.com/newsletter/subscribe.html

Computer Science

Functional Programming
Regular Expressions

Programming

ASP.NET 3.5
Adobe Flex
Ajax
Apex
ASP.NET Ajax
ASP.NET
C
C++
C++ Boost Libraries
C++ Game Programming
C#
Code Search Engines and Code Sites
Computer Game Programming
CSS 2.1
Dojo
Facebook Developer Platform
Flash 9
Functional Programming
Java
Java Certification and Assessment Testing
Java Design Patterns
Java EE 5
Java SE 6
Java SE 7 (Dolphin) Resource Center
JavaFX
JavaScript
JSON
Microsoft LINQ
Microsoft Popfly
.NET
.NET 3.0
.NET 3.5
OpenGL
Perl
PHP
Programming Projects
Python
Regular Expressions
Ruby
Ruby on Rails

Silverlight
Visual Basic
Visual C++
Visual Studio Team System

Web 3D Technologies
Web Services
Windows Presentation Foundation
XHTML
XML

Games and Game

Programming
Computer Game Programming
Computer Games
Mobile Gaming
Sudoku

Internet Business

Affiliate Programs
Competitive Analysis
Facebook Social Ads
Google AdSense
Google Analytics
Google Services
Internet Advertising
Internet Business Initiative
Internet Public Relations
Link Building
Location-Based Services
Online Lead Generation
Podcasting
Search Engine Optimization
Selling Digital Content
Sitemaps
Web Analytics
Website Monetization
YouTube and AdSense

Java

Java
Java Certification and Assessment Testing
Java Design Patterns
Java EE 5
Java SE 6

Java SE 7 (Dolphin) Resource Center
JavaFX

Microsoft

ASP.NET
ASP.NET 3.5
ASP.NET Ajax
C#
DotNetNuke (DNN)
Internet Explorer 7 (IE7)
Microsoft LINQ
.NET
.NET 3.0
.NET 3.5
SharePoint
Silverlight
Visual Basic
Visual C++
Visual Studio Team System
Windows Presentation Foundation
Windows Vista
Microsoft Popfly

Open Source & LAMP Stack

Apache
DotNetNuke (DNN)
Eclipse
Firefox
Linux
MySQL
Open Source
Perl
PHP
Python
Ruby

Software

Apache
DotNetNuke (DNN)
Eclipse
Firefox
Internet Explorer 7 (IE7)
Linux
MySQL
Open Source
Search Engines

SharePoint
Skype
Web Servers
Wikis
Windows Vista

Web 2.0

Alert Services
Attention Economy
Blogging
Building Web Communities
Community Generated Content
Facebook Developer Platform
Facebook Social Ads
Google Base
Google Video
Google Web Toolkit (GWT)
Internet Video
Joost
Location-Based Services
Mashups
Microformats
Recommender Systems
RSS
Social Graph
Social Media
Social Networking
Software as a Service (SaaS)
Virtual Worlds
Web 2.0
Web 3.0
Widgets

Dive Into Web 2.0

eBook
Web 2 eBook

Other Topics

Computer Games
Computing Jobs
Gadgets and Gizmos
Ring Tones
Sudoku

前 言

“The chief merit of language is clearness . . .”

—Galen

欢迎来到 C++ 编程的世界! Deitel & Associates 公司为 Prentice Hall 出版社编写专业的编程语言图书和教材,面向全球提供企业培训课程,发展 Internet 业务。本书的读者对象是那些还不了解 C++ 的程序员,他们可能已经知道或者还不熟悉面向对象编程。

本书的特点

这个前言部分,将为读者提供一个书中关于 C++ 和面向对象编程所涵盖范围的概要。以下是本书的一些主要特性。

- **尽早接触类与对象方法。**从一开始到贯穿于全书的过程中,都体现了面向对象编程的思想。
- **集成的案例研究。**第 3~7 章开发了一个 GradeBook 类,第 9~10 章开发了一个 Time 类,第 12~13 章开发了一个 Employee 类,第 1~7 章、第 9 章、第 13 章和附录 E 中开发了一个选读的 OOD/UML ATM 案例研究。
- **统一建模语言 2(UML 2)。**统一建模语言(Unified Modeling Language, UML)已经成为了设计面向对象系统的首选图形化建模语言。书中用 UML 类框图来可视化地表示类以及它们之间的继承关系,并采用 UML 活动框图演示了每个 C++ 控制语句的控制流。在选读的 OOD/UML ATM 案例研究中,重点讲解了 UML。
- **选读的 OOD/UML ATM 案例研究。**首先讲解的是一个经过了简化的 UML 2 子集,然后让面向对象的设计员/程序员初学者进行初次的设计体验。这个案例研究已经由不同的 OOD/UML 学术和业界专家评审过了。这不是一个练习,而是一个完整开发的端到端的学习体验,最后形成了一个完整的过程,由 877 行 C++ 代码实现。在这个前言的后面,将对这个案例研究的 9 个小节进行详细说明。
- **函数调用栈解释。**第 6 章中给出了函数调用栈和活动记录的详细讨论(还给出了演示),以解释 C++ 是如何能够跟踪当前正在执行的函数的、函数的自动变量是如何在内存中自动维护的,以及当函数执行完成后该返回到什么地方。
- **string 类。**本书采用 string 类而不是 C 风格的基于指针的 char * 字符串,用于书中大多数字符串操作中。第 8、10、11、19 章中包含了对 char * 的讨论,向读者提供了指针操作的实践,以演示 new 和 delete 操作符的动态内存分配、建立自己的 string 类、在遗留的 C 和 C++ 代码中使用 char * 字符串等。
- **类模板 vector。**本书使用类模板 vector 而不是 C 风格的基于指针的数组操作。但是,第 7 章中依然探讨了 C 风格的基于指针的数组,以便能使用遗留的 C 和 C++ 代码,而且它还是第 11 章中建立定制的 Array 类的基础。
- **继承和多态的介绍。**第 12~13 章中包含一个 Employee 类层次,对于新接触 OOP 的程序员而言,能使他们对继承和多态的理解变得清晰而触手可及。
- **讨论并演示了多态是如何在“幕后”起作用的。**第 13 章中包含一个详细的框图,解释了 C++ 是如何在内部实现多态、虚函数和动态绑定的。这些内容将帮助读者对这些功能的工作原理有深刻的理解。更重要的是,它可以帮助读者理解多态的开销,也就是额外的内存消耗和处理器时间。它有助于决定什么时候该使用多态、什么时候不该使用。
- **标准模板类库(STL)。**STL 也许是本书中关于软件复用的最重要的主题之一。STL 中定义了功能强大的、基于模板的、可复用的组件,实现了许多常用的数据结构以及处理它们的算法。第 20 章中将

介绍 STL 并探讨它的三个主要概念:容器、迭代器和算法。利用 STL 组件提供的极其强大的表现能力,经常可将占据许多行的非 STL 代码缩减成一条语句。

- **符合 ISO/IEC C++ 标准。**对于本书中的程序,作者已经根据最新的 ISO/IEC C++ 标准文档进行了审查,以满足完备性和精确性。[注:C++ 标准的 PDF 版本(文档编号 INCITS/ISO/IEC 14882-2003)可以在 webstore.ansi.org/ansidocstore/default.asp 购买到。]
- **C++ 的未来。**第 21 章探讨了 C++ 的未来,在那里引入了 Boost C++ 类库、Technical Report 1 (TR1) 和 C++ 0x。Boost C++ 类库是免费的开源库,它由 C++ 社区的成员创建。Technical Report 1 描述了对 C++ 标准类库的建议修改之处,其中有许多是基于当前的 Boost 类库的。目前,C++ 标准委员会正在修订 C++ 标准,新标准的主要目标是使 C++ 更容易学习、提升类库的创建功能,以及提高与 C 编程语言的兼容性。最新的标准发表于 1998 年。目前,正在修订的新标准被称为 C++ 0x,它开始于 2003 年,新标准很可能会在 2009 年发布^①。它将包括对语言核心的修改,当然更多的还是根据 TR1 中的类库进行的修改。本书将分析 TR1 中的类库,并提供“正则表达式”类库和“智能指针”类库的代码例子。
- **附录中讲解了调试器。**附录 G 和附录 H 中分别讲解了 Visual Studio 调试器和 GNU C++ 调试器的使用方法。
- **多平台上的代码测试。**本书中的代码例子已经在各种流行的 C++ 平台上测试通过。对于其中的大多数例子,迁移到与标准兼容的编译器中是很容易的。
- **给出多种平台下的错误和警告消息。**对于那些故意包含错误以说明关键概念的程序,书中给出了多种流行平台下的错误消息。

所有这些结果都已经由行业开发人员和业界专家仔细评审过了。我们相信,本书将为读者提供一种信息丰富的、有趣的、充满挑战而又令人愉悦的 C++ 学习体验。

在阅读本书时,如果有任何问题,可发电子邮件至 deitel@deitel.com,我们会尽快回复。有关本书的更新信息、所有支持性的 C++ 软件的状态,以及关于 Deitel 出版物和服务的所有最新消息,请访问 www.deitel.com。在 www.deitel.com/newsletter/subscribe.html 上注册,即可获得免费的 Deitel Buzz Online 电子邮件新闻简报,并可在 www.deitel.com/ResourceCenters.html 上获得正在不断增加的 C++ 以及相关的资源中心的清单。每个星期,我们都会在新闻简报中发布最新的资源中心信息。

本书编排特色

书中包含大量的例子,浓缩了良好的软件工程的原则,并强调了程序的清晰性。本书是通过例子来讲解的。两位作者是在全球业界讲授编程语言的教师,他们为 Deitel & Associates 公司各种层次的政府、行业、军队以及学院客户讲授过培训课程。

活代码方法。本书是以“活代码”例子为载体的。通过这种方法,每个新概念都是在一个完全可工作的 C++ 程序中体现的,紧跟在程序后面的是一个或几个实际的执行结果,它们显示了程序的输入和输出。

语法阴影。本书中所有的 C++ 代码都采用带阴影的语法,它类似于大多数 C++ 集成开发环境 (IDE) 和代码编辑器中带颜色的语法代码。这极大地提高了代码的可读性——对于包含 15 500 多行代码的本书而言,这是一个特别重要的目标。书中的语法阴影遵循如下的约定。

注释以斜体显示。

关键字显示为粗斜体。

错误和 ASP.NET 脚本分隔符以粗体黑字出现。

常量和字面值用粗体灰字表示。

所有其他代码采用普通的黑体字。

突出显示重要代码段。在每个程序的关键代码段的周围,放上了白色矩形框。

^① 到本书出版时为止,新标准还没有推出——编者注。

用字体突出显示。对每个定义的关键术语以及索引中的页面引用,都以粗斜体表示,便于查找。屏幕上显示的内容以加粗的 Helvetica 字体表示(例如 File 菜单),而 C++ 程序文本采用的是 Lucida 字体(例如, `int x = 5`)。^①

本书的网址。书中所有的源代码例子,都可以从 www.deitel.com/books/cppfp/ 下载。

目标。每一章都以一个目标概述开头。这使读者可以知道本章将讲解什么,并有机会在学习完一章之后检验一下是否达到了目标。

引文。目标的前面有一段引文,有些是幽默,有些是哲理,而另外一些是有趣的真知灼见。但愿读者能够喜欢它们。

提纲。每一章的提纲可帮助读者按自顶向下的方法学习,从而可以预先知道后面将讲解的内容,并可有效地安排学习进度。

插图。书中包含大量的图标、表格、线状图、程序以及程序输出。控制语句中的控制流是用 UML 的活动框图建模的。UML 类框图建模了类的字段、构造函数和方法。在选读的 OOD/UML 2 ATM 案例研究中,大量使用了 6 种主要的 UML 框图类型。

编程提示。书中提供的编程提示,可帮助读者关注程序开发过程中的重要方面。这些提示和实践,体现了两位作者累计 70 年的编程经验之精华,它们提供了建立良好软件的基础。



好的编程经验

“好的编程经验”提示有助于得到更清晰的、更易理解的和更易维护的程序。



常见编程错误

在“常见编程错误”提示中指出这些错误,可使读者减少犯同样错误的可能性。



错误防止提示

这种提示包含暴露程序中的 bug 并删除它们的建议,其中许多这样的提示描述的是如何从一开始就防止将 bug 带入 C++ 程序中。



性能提示

这些提示强调的是使程序运行得更快或使内存占用最小化的机会。



可移植性提示

这个提示的作用是帮助读者编写能在各种平台下运行的代码,并解释了 C++ 是如何获得它的高度可移植性的。



软件工程观察

“软件工程观察”提示强调的是体系性以及设计性的问题,它们会影响软件系统的建立,尤其是大规模的系统。

“小结”节。每一章都以一个简短的“小结”节结束,回顾本章的内容并过渡到下一章。

数千个索引项。书中包含大量的索引,当将本书作为参考书时,这些索引尤其有用。

活代码例子的“双索引”。对于书中的每个源代码程序,对图题的索引同时采用字母表顺序和“例子”下子索引项的方法。这样做更易找出使用特定特性的例子。^②

内容介绍

下面简要介绍一下读者将从本书中学到的 C++ 知识。图 1 中描述的是本书中各章间的依赖关系。建议读者按箭头所示的顺序学习本书,当然其他的学习方法也是可行的。

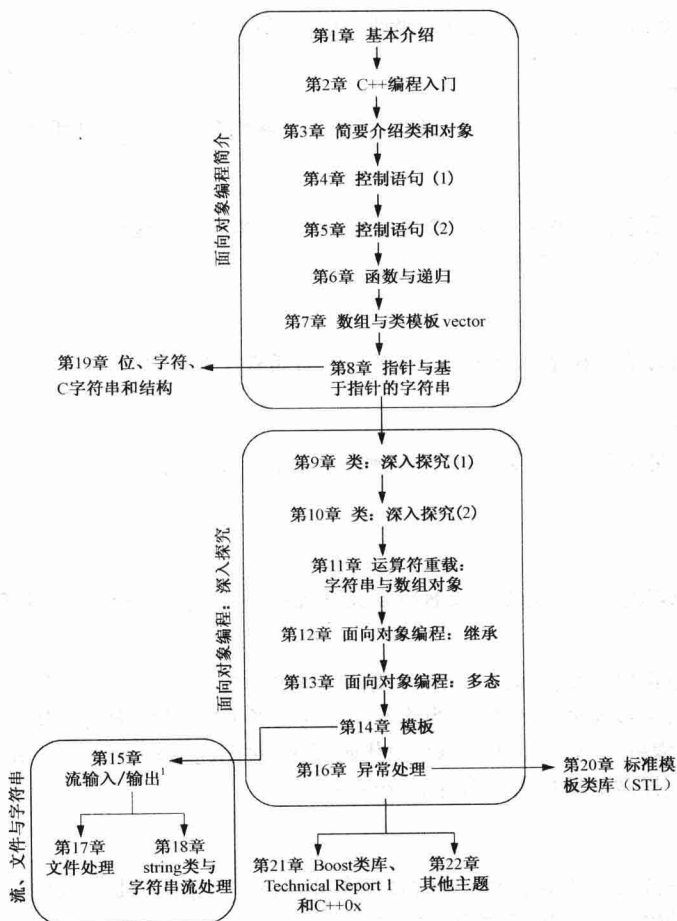
① 简体中文版中取消了这些字体区分,统一采用宋体——编者注。

② 本书取消了对正文中例子的索引——编者注。

第1章“基本介绍”。探讨了C++编程语言的起源,并介绍了典型的C++编程环境。本章给出了Windows和Linux平台下一个以“测试驱动的”典型C++程序。还引入了基本的对象技术概念和技术,介绍了统一建模语言。

第2章“C++编程入门”。本章提供了C++编程的初步介绍。本章中的程序演示了如何在屏幕上显示数据、从键盘获得数据、作出判断以及执行各种算术运算。

第3章“简要介绍类和对象”。本章初步介绍了类以及对象,利用各种简单的、真实的例子,引入了类、对象、成员函数、构造函数和数据成员。本章搭建起了一个设计良好的框架,用于在C++中组织面向对象程序。通过一个简单的例子引入了类的概念。然后,按照精心设计的步骤,通过7个完整的程序来展示如何创建并使用自己的类。通过这些例子就开始了我们的集成案例研究,将开发一个GradeBook类,教师可用这个类来维护学生的测验成绩。这个案例研究将在后面的几章中逐步强化,最后的版本在第7章给出。GradeBook案例研究描述了如何定义类并用它创建对象。这个案例研究探讨了如何声明和定义成员函数,以实现类的行为;如何声明数据成员,以实现类的属性;如何调用对象的成员函数,使它执行任务。本章还介绍了C++的标准类库string类,并创建了string对象,用于保存GradeBook对象所表示的课程名。解释了类的数据成员与函数的局部变量的差异,并讲解了如何利用构造函数来保证创建对象时会初始化它的数据。也展示了如何将类的定义与使用这个类的客户代码分离,以提高软件复用性。介绍了良好软件工程的另一个基本原则——将接口与实现分离。



1 第15章的大部分内容以第7章为基础,少部分要求先阅读第12章和第14章。

图1 各章的相互关系图

第4章“控制语句(1)”。本章主要涉及创建有用的类时的程序开发过程。引入了一些用于作出判断(if和if...else)和循环(while)的控制语句。分析了使用第3章中GradeBook类的计数器控制循环和标记控制循环,并介绍了C++中的增量、减量和赋值运算符。本章包含GradeBook类的两个增强版本,它们都以第3章的版本为基础。本章使用了几个简单的UML活动框图来展示通过每条控制语句的控制流。

第5章“控制语句(2)”。继续C++控制语句的探讨,例子中使用了for循环语句、do...while循环语句、switch选择语句、break语句以及continue语句。创建了GradeBook类的一个增强版本,利用switch语句来计算由用户输入的A、B、C、D和F成绩个数。本章还探讨了逻辑运算符。

第6章“函数与递归”。深入探讨对象以及它们的成员函数。研究了C++标准类库中的函数,并更进一步地分析了如何创建自己的函数。本章中的第一个例子延续GradeBook类案例研究,给出了一个带多个参数的函数例子。读者可能会喜欢本章中对随机数和模拟的讲解,也会对称为craps的掷骰子游戏的讨论充满兴趣,它更好地使用了控制语句。本章探讨了所谓的“C++中对C增强的功能”,包括内联函数、引用指针、默认实参、一元作用域分解操作符、函数重载以及函数模板等。还给出了C++的按值调用和按引用调用的功能。本章中的头文件表引入了书中将用到的许多头文件。探讨了函数调用栈和活动记录,以解释C++是如何能够跟踪当前正在执行的函数的、函数的自动变量是如何在内存中自动维护的,以及当函数执行完成后该返回到什么地方。最后,深入研究了递归。

第7章“数组与类模板vector”。本章讲解如何处理值的列表(list)以及表(table)。探讨了数组中具有同种类型的数据项的构造方式,演示了数组是如何有利于由对象执行的任务的。本章的前面部分使用了C风格的、基于指针的数组。正如第8章中将看到的那样,可以将它们看成指向内存中数组内容的指针。然后,将数组表示成功能完整的对象,引入了C++标准类库vector类模板,它是一种健壮的数组数据结构。本章中给出了一维数组和二维数组的大量例子,它们示范了许多常见的数组操作,包括输出条形图、排序数据以及将数组传递给函数等。本章还包含了最后两个GradeBook案例研究小节,在程序执行期间用数组来保存学生成绩。这个类的旧版本处理用户输入的一组成绩,但不在类的数据成员中保存各个成绩值。在本章中,将使用数组来使GradeBook类的对象能操作内存中的一组成绩,从而消除了要不断重复输入同一组成绩的需求。这个类的第一个版本将成绩保存在一个一维数组中,第二个版本用一个二维数组来保存大量学生在一个学期中的多次考试成绩。本章中另一项主要内容是讨论基本的排序和搜索技术。

第8章“指针与基于指针的字符串”。本章讲解了C++中功能最为强大的特性之一:指针,提供了关于指针运算符、按引用调用、指针表达式、指针运算、指针与数组的关系、指针数组以及函数指针等的详细解释。讨论了如何对指针使用const,以实现最低权限原则。探讨了使用sizeof运算符来判断程序编译期间数据类型或数据项的字节大小。C++中,指针、数组和C风格的字符串之间具有紧密的关系,因此本章中介绍了基本的C风格的字符串处理函数,比如getline(输入文本行)、strcpy和strncpy(复制字符串)、strcat和strncat(拼接两个字符串)、strcmp和strncmp(比较两个字符串)、strtok(将字符串“标记化”成片段)以及strlen(返回字符串的长度)。本章中,经常会用(第3章中介绍的)string对象来替代C风格的、基于char*指针的字符串。但是,本章中将使用char*字符串,以帮助读者熟悉指针并为进入一个专业领域做好准备,在那里会接触到过去三十年来所开发的大量C遗留代码。位于C和“原始C++”中的数组和字符串,在内存中是指向数组和字符串内容的指针(甚至函数名也是指针)。

第9章“类:深入探究(1)”。本章继续面向对象编程的探讨,使用了一个内容丰富的Time类案例研究来演示访问类成员、将接口与实现分离、使用访问函数和工具函数、用构造函数初始化对象、用析构函数删除对象、通过默认的逐成员(memberwise)复制进行赋值以及软件复用等。探讨了在对象的生命期内构造函数和析构函数的调用顺序。Time类案例研究的一个修改版本演示了当成员函数返回私有数据成员的引用时可能导致的问题,这样做会破坏类的封装性。

第10章“类:深入探究(2)”。继续类的探讨并给出了其他的面向对象编程概念。本章讲解了常量对象、常量成员函数、合成(建立类的过程,其成员为其他类的对象)、友元函数、友元类(对类的私有成员和保护成员具有特殊的访问权限)、this指针(使对象知道自己的地址)、动态内存分配、静态类成员(包含和操作类际数据)、流行的抽象数据类型(数组、字符串和队列)的例子、容器类以及迭代器。在常量对象的讨论

中,提及了 `mutable` 关键字,它采用一种微妙的方法来修改“不可见的”常量对象。还探讨了使用 `new` 和 `delete` 操作符的动态内存分配问题。当使用 `new` 操作符失败时,默认情况下程序会终止,因为在标准 C++ 中,它会抛出异常。本章通过一个基于视频游戏的场景,探讨了静态类成员。强调了对类的客户隐藏实现细节的重要性。接着讨论了代理类,它提供了对类的客户隐藏实现(包括类的头文件中的私有数据)的一种方法。

第 11 章“运算符重载:字符串与数组对象”。本章讲解的是 C++ 课程中最为流行的一个主题,专业人士都会喜欢这些材料。他们发现,这是对第 9 章和第 10 章中那些详细讨论的类的补充。运算符重载使程序员能通知编译器,该如何将已有的运算符用于新类型的对象。C++ 已经知道了如何将运算符用于内置类型,比如整数、浮点数和字符。但是,如果创建了一个新的 `String` 类,则将加号放在这种 `String` 对象之间时,表示的是什么意思呢?许多程序员将加号用于字符串时,表示的是字符串拼接。在第 11 章,将讲解如何“重载”加号,使得当它用在表达式中的两个 `String` 对象之间时,编译器将产生一个调用“运算符函数”的函数,用于拼接两个 `String` 对象。本章讲解了运算符重载的基本知识、运算符重载的限制、重载类成员函数与非成员函数的比较、重载一元运算符和二元运算符、类型间的转换等。这一章的特点是集中分析了几个案例研究,包括 `Array` 类、`String` 类和 `Date` 类。大量使用运算符重载,可使程序中的类“熠熠生辉”。

第 12 章“面向对象编程:继承”。本章介绍了面向对象编程语言的最基本功能之一:继承,它是软件复用的一种形式。通过吸收已有的类的功能并添加适当的新功能,即可快速且简单地开发出新的类。在“员工继承层次”案例研究中,本章依次给出了 5 个例子,演示了私有数据、保护数据以及使用继承的良好软件工程。本章还探讨了基类与派生类、保护成员、公有继承、保护继承、私有继承、直接基类、间接基类、基类和派生类中的构造函数和析构函数以及使用继承的软件工程,比较了继承(“是”关系)和合成(“有”关系)。

第 13 章“面向对象编程:多态”。本章介绍面向对象编程语言的另一个基本功能:多态。它以第 12 章的继承概念为基础,关注的是类层次中类间的关系以及用这些关系建立起来的强大的处理能力。当许多类通过继承与一个共同基类相关联时,每个派生类对象都可以当成一个基类对象。这使得能够独立于派生类对象的特定类型,以一种简单而通用的方式编写程序。新类型的对象可以由同一个程序处理,从而使系统更具扩展性。本章探讨了通过虚函数实现多态行为的机制。它对抽象类(对象不能实例化)和具体类(对象能够实例化)的处理是不同的。对于通过继承层次向类提供可继承的接口而言,抽象类是有用的。本章包含了 `vtable`(虚函数表)的演示和精确的解释,C++ 编译器会自动建立 `vtable`,以支持多态。本章以运行时类型信息(RTTI)和动态强制转换的介绍结束,它们使程序能在运行时判断对象的类型,然后对这个对象采取相应的行动。

第 14 章“模板”。本章探讨了 C++ 中一种功能更强大的软件复用特性,即模板。函数模板使程序员能够用单段代码指定相关(重载)函数的全部范围,称为函数模板特殊化;类模板使程序员能够用单段代码指定相关类的全部范围,称为类模板特殊化。这种技术被称为泛型编程。可以为堆栈类编写一个类模板,然后让 C++ 产生各自的类模板特殊化,比如 `int` 堆栈类、`float` 堆栈类、`string` 堆栈类,等等。本章探讨了类模板的类型参数、非类型参数以及默认类型。还探讨了模板与其他 C++ 特性之间的关系,例如重载、继承、友元和静态成员。对于第 20 章的标准模板类库容器、迭代器和算法的讨论,本章极大地强化了模板的特性。

第 15 章“流输入/输出”。包含对 C++ 标准输入/输出功能的全面讲解。本章将探讨能够满足大多数常见 I/O 操作的那些功能,并将简要概述其他的功能。许多 I/O 特性都是面向对象的。C++ 具有各种 I/O 能力,包括使用流插入运算符的输出、使用流提取运算符的输入、类型安全的 I/O、格式化的 I/O、非格式化的 I/O(用于性能),等等。通过重载流插入运算符(`<<`)和流提取运算符(`>>`),就可以指定如何对用户定义类型的对象执行 I/O 操作。C++ 提供了执行格式化任务的各种流操作子。本章讨论的流操作子,提供的功能包括以各种基数显示整数值、控制浮点数精度、设置字段宽度、显示小数点和数字后面的 0、将输出对齐、设置和取消格式状态、设置字段的填充字符,等等。本章还给出了一个例子,它创建了用户定义的输出流操作子。

第 16 章“异常处理”。探讨异常处理如何使程序员能编写出健壮的、容错的、适合于关键业务和关键

任务环境的程序。本章讲解了何时适合进行异常处理,介绍了 try 代码块的异常处理能力、throw 语句以及 catch 处理器,演示了何时该重抛异常、如何重抛异常,解释了如何编写异常规范以及处理意外的异常,还探讨了异常与构造函数、析构函数和继承之间的紧密联系。研究了重抛异常,并演示了当内存耗尽时,new 操作是如何失败的。当 new 失败时,许多老式的 C++ 编译器会默认返回 0。本章展示了当 new 失败时的一种新处理办法:抛出 bad_alloc 异常(内存分配失败)。演示了对于内存耗尽的情形,如何利用 set_new_handler 函数来指定要调用的自定义函数。讲解了如何使用 auto_ptr 类模板来动态删除隐式分配的内存,从而避免了内存泄漏。在本章的结尾,给出了标准类库的异常层次。

第 17 章“文件处理”。探讨了创建并处理顺序文件和随机访问文件的技术。本章从数据层次的介绍开始,数据层次从位开始,依次为字节、字段、记录和文件。接下来,讲解了 C++ 中文件和流的视图。探讨了顺序文件并建立了几个程序,演示如何打开和关闭文件、如何有序地将数据保存到文件中、如何有序地从文件读取数据。然后,探讨了随机访问文件并建立了几个程序,演示如何创建随机访问文件、如何用随机访问读写文件中的数据、如何从随机访问文件中有序地读取数据。这个案例研究将访问顺序文件和随机访问文件的技术组合进了一个完整的事务处理程序中。

第 18 章“string 类与字符串流处理”。探讨了 C++ 从内存的字符串中输入数据、将数据输出到内存中的字符串的能力,这些能力常被称为“核内格式化”或“字符串流处理”。string 类是标准类库中必要的一部分。本章保留了第 8 章中 C 风格的、基于指针的字符串,后面给出了这样做的几个理由。首先,它强化了读者对于指针的理解。其次,在下一个 10 年左右的时间里,C++ 程序员依然需要能够阅读并修改过去 25 年中累积起来的大量 C 遗留代码,这些代码将字符串当作指针处理。而在过去的许多年中,业界的许多 C++ 代码也是这样编写的。本章探讨了字符串的赋值、拼接和比较。讲解了如何判断字符串的各种特性,比如字符串的大小、容量以及是否为空。也讲解了如何调整字符串的大小。分析了能在字符串中找出字符串的各种“find”函数(前向或后向搜索字符串),展示了如何从由字符组成的字符串中找出某个字符第一次或最后一次出现的位置,也展示了如何从由字符组成的字符串中找出没有包含某个字符第一次或最后一次出现的情况。还讲解了如何替换、删除和插入字符串中的字符,如何将 string 对象转换成 C 风格的 char * 字符串。

第 19 章“位、字符、C 字符串和结构”。本章先将 C++ 的结构与类进行了比较,然后定义并使用了 C 风格的结构。讨论了如何声明、初始化结构以及将结构传递给函数。C++ 中功能强大的位操作能力,使程序员能编写出涉及低级硬件功能的程序。这有助于让程序将字符串按位来处理、设置每个位,从而能更紧凑地保存信息。这些功能一般只在低级汇编语言中才有,它们使程序员能编写出系统软件,比如操作系统和联网软件。第 8 章中讨论过 C 风格的 char * 字符串操作,在那里给出了大多数流行的字符串操作函数。在本章中,继续讲解了字符以及 C 风格的 char * 字符串。给出了 <cctype> 类库中各种字符操作的能力,比如测试字符的能力,以判断它是数字、字母表中的字符、字母数字字符、十六进制数字、小写字母还是大写字母。本章还给出了与字符串相关的各种类库中的其他字符串操作函数。

第 20 章“标准模板类库(STL)”。STL 中定义了功能强大的、基于模板的、可复用的组件,实现了许多常用的数据结构以及处理它们的算法。STL 通过模板实现了泛型编程的概念,第 14 章介绍过模板。本章探讨了 STL 的三个主要概念:容器(模板化数据结构)、迭代器和算法。容器是能够保存任何数据类型对象的数据结构。存在三种风格的容器类:第一类容器、适配器和近容器。迭代器的属性与指针类似,程序可以利用迭代器操作容器元素。事实上,STL 容器也可以在操作标准数组时将标准指针用作迭代器。用迭代器操作容器是非常方便的,当与 STL 算法结合使用时,它们具有极其强大的威力。STL 算法是用于执行常见数据操作的函数,这些操作包括搜索、排序和比较元素(或整个容器)。其中的大多数算法都使用迭代器来访问容器元素。

第 21 章“Boost 类库、Technical Report 1 和 C++ 0x”。主要关注 C++ 的未来。介绍了 Boost 类库,它是免费的、开源的 C++ 类库集合。Boost 类库是精心设计的,以便能与 C++ 标准类库很好地协作。然后,探讨了 Technical Report 1 (TR1),它是针对 C++ 标准类库的变化和添加的描述。TR1 中的许多类库,都是最近从 Boost 类库中派生出来的。本章简要描述了 TR1 类库。给出了两个最有用的类库的代码例子,它们是

Boost.Regex 和 Boost.Smart_ptr。Boost.Regex 类库提供了对正则表达式的支持。本章演示了如何使用类库来搜索匹配正则表达式的字符串、验证数据、替换字符串的一部分、将字符串分隔成标记(token)。Boost.Smart_ptr 类库提供了智能指针,可用来帮助管理动态分配的内存。探讨了包含在 TR1 中的两个智能指针:shared_ptr 和 weak_ptr。提供的例子演示了它们如何能用来避免常见的内存管理错误。本章还探讨了即将发布的 C++ 新标准。

第 22 章“其他主题”。这一章是其他各种 C++ 主题的汇总。探讨了另外一个强制转换运算符——const_cast。const_cast、static_cast(第 5 章)、dynamic_cast(第 13 章)和 reinterpret_cast(第 17 章)提供的转换机制,比从 C 继承的原始 C++ 强制转换运算符(现在不提倡使用它们)更为健壮。本章探讨的名字空间,对于建立牢固系统的软件开发人员而言,这个特性尤其关键。名字空间防止了命名冲突,否则会妨碍大型软件的效果。还探讨了关键字 mutable,它使得常量对象的成员可以变化。以前,这种功能是由“强制转换掉常量性”实现的,这是一种危险的操作。还探讨了指针成员运算符.* 和 -> *、多重继承(包括“菱形继承”存在的问题)以及虚拟基类。

附录 A“运算符的优先级和结合性”。给出了 C++ 运算符符号的完整清单,其中的每个运算符都是按行排列的,依次给出了它的运算符符号、名称和它的结合性。

附录 B“ASCII 字符集”。本书中的所有程序都使用 ASCII 字符集,它在这个附录中列出。

附录 C“基础类型”。列出了 C++ 中的基础类型。

附录 D“预处理器”。讲解了预处理器的指令。这个附录包含 #include 指令的更完整的信息。在编译程序以及创建符号常量和宏的 #define 指令之前,这个指令会使指定文件的一个副本被包含到指令所在的位置。还解释了条件编译,能够控制预处理器指令的执行以及程序代码的编译。# 运算符会将它的操作数转换成一个字符串,而 ## 运算符会将两个标记拼接成字符串。本章给出了各种预定义的预处理器符号常量(__LINE__、__FILE__、__DATE__、__STDC__、__TIME__ 和 __TIMESTAMP__)。最后讨论的是 <cassert> 头文件的 assert 宏,它用于程序测试、调试、验证和确认。

附录 E“ATM 案例研究代码”。包含使用 UML 的面向对象设计案例研究的实现。这个附录将在(随后给出的)“选读的软件工程案例研究介绍”小节中介绍。

附录 F“UML 2 的其他框图类型”。探讨了 OOD/UML 案例研究中没有涉及的其他 UML 2 框图类型。

附录 G“使用 Visual Studio 调试器”。演示了 Visual Studio 调试器的主要特性,它使程序员能监视程序的执行,以找出并删除逻辑错误。这个附录给出了按步执行的指令,以便读者能学习到如何实际使用调试器。

附录 H“使用 GNU C++ 调试器”。演示了 GNU C++ 调试器的主要特性。这个附录给出了按步执行的指令,以便读者能学习到如何实际使用调试器。

参考文献。列出了许多图书和文章,以便能进一步学习 C++ 和面向对象编程。

索引。丰富的索引可方便读者根据任何术语或概念找到正文中的相关内容。

使用 UML 的 ATM 面向对象设计:选读的软件工程案例研究介绍

这一部分介绍的是书中选读的用 UML 进行面向对象设计的案例研究。这里给出的是书中 9 个“软件工程案例研究”节的内容(第 1~7 章、第 9 章和第 13 章)。完成这个案例研究之后,对于复杂的 C++ 程序进行面向对象的设计和实现就会非常熟悉了。

ATM 案例研究中的设计是由 Deitel & Associates 公司开发的,并经过了行业专家和学术界不同开发评审团队的审查。银行及其全球客户所使用的真正的 ATM 系统,采用的设计要复杂得多,它所考虑的问题要比本书中的多得多。本书的目标是创建一个简单的设计,它对 OOD 和 UML 新手是清晰的,但依然表达了主要的 OOD 概念以及相关的 UML 建模技术。

1.10 节“软件工程案例研究:对象技术和 UML 介绍”。讲解 UML 与面向对象设计案例研究。这一节涉及对象技术的基本概念和术语,包括类、对象、封装、继承和多态。还介绍了 UML 的历史。这是案例研究中唯一必读的一节。

2.7 节“(选读)软件工程案例研究:分析 ATM 需求规范”。探讨系统所要求的需求规范,系统就是设计并实现一个简单的自动柜员机(ATM)的软件。将分析面向对象系统的一般结构和行为。探讨了 UML 如何满足后续“软件工程案例研究”各节中的设计过程,为此提供了建模系统的其他几种框图类型。讨论了由需求规范指定的 ATM 系统与用户间的交互。特别地,分析了可能在用户与系统间出现的场景,这些场景被称为“用例”。这一节中用 UML 的用例框图建模了这些交互。

3.11 节“(选读)软件工程案例研究:确定 ATM 需求规范中的类”。开始 ATM 系统的设计。从需求规范中提取名词和名词短语,确定了 ATM 系统的类或“建筑块”。我们将这些类放入了 UML 类框图中,这个框图描述了类的结构。类框图还描述了类间的关系,称为关联(association)。

4.11 节“(选读)软件工程案例研究:确定 ATM 系统的类属性”。关注 3.11 节中描述的类的属性。类具有属性(数据)和操作(行为)。在后面的几节中将看到,改变对象的属性,通常会影响到它的行为。为了确定案例研究中类的属性,需从需求规范中提取出描述(定义类的)名词和名词短语的形容词,然后将属性放入 3.11 节创建的类框图中。

5.10 节“(选读)软件工程案例研究:确定 ATM 系统中对象的状态和活动”。探讨在任何给定的时刻对象如何具有指定的条件,条件称为“状态”。当对象接收到改变状态的消息时,会发生状态转移。UML 提供了状态机框图,它确定了对象能够具有的一组可能状态,并建模了对象的状态转移。对象还具有活动——生命周期内对象所做的工作。UML 提供活动框图——建模对象活动的流程图。这一节利用这两种框图类型来建模 ATM 系统的行为,比如系统如何完成取款交易、验证完用户后如何响应等。

6.22 节“(选读)软件工程案例研究:确定 ATM 系统的类操作”。明确类的操作或服务。这一节从需求规范中提取了动词和动词短语,它们指定每个类的操作。然后,修改 3.11 节中的类框图,以包含与类相关联的每个操作。至此,就已经从需求规范中收集到了所有可能的信息。但是在后面的几章中,当介绍到像继承之类的主题时,将修改这些类和框图。

7.12 节“(选读)软件工程案例研究:ATM 系统中对象间的协作”。提供 ATM 系统模型的“粗略骨架”。在这一节中,将看到系统是如何工作的。通过探讨协作,将分析这个仿真系统的行为。协作即对象间为了沟通而彼此发送的消息。6.22 节中确定的类操作,实际上就是系统中对象间的协作。先确定这些协作,然后将它们放入通信框图中,即建模协作的 UML 框图。这个框图表明了哪些对象需要协作以及何时进行协作。书中列出的这个通信框图,是执行 ATM 余额查询所需的对象间的协作。接着,列出的是 UML 的顺序框图,它建模系统中的交互。这个框图强调的是消息的时间排序,顺序框图建模系统中的对象如何交互,以便执行取款和存款交易。

9.11 节“(选读)软件工程案例研究:开始编程 ATM 系统的类”。暂时脱离系统行为的设计,开始突出第 9 章中所讨论材料的实现过程。利用 3.11 节中的 UML 类框图和 4.11 节、6.22 节中讨论的类的属性和操作,这一节展现的是在 C++ 中如何将设计实现成类。书中并没有实现所有的类,因为设计过程还没有最后完成。利用 UML 框图,这一节中创建了 Withdrawal 类的代码。

13.10 节“(选读)软件工程案例研究:在 ATM 系统中加入继承”。继续介绍面向对象编程。将考虑继承——具有共同特性的类,可以从一个“基本”类中继承属性和操作。在这一节中,分析了 ATM 系统可以从继承中获得的好处。我们将这些好处放入了建模继承关系的类框图中,UML 称这些关系为“泛化”。对 3.11 节中的类框图进行了修改,利用继承来分组具有相似特征的类。这一节以一个 ATM 系统模型部分的设计结束。ATM 系统的完整 C++ 实现见附录 E,它一共有 877 行代码。

附录 E“ATM 案例研究代码”。ATM 系统案例研究的主要工作涉及模型(即数据和逻辑)的设计。在这个附录中,用 C++ 实现了这个模型。它使用了前面所创建的所有 UML 框图,给出了实现这个模型所要求的 C++ 类。利用了本书各章中讲到的 UML 面向对象设计和 C++ 的面向对象编程。看完这个附录之后,就完成了——一个真实系统的设计和实现,并可充满信心地应对更大型的系统了。

附录 F“UML 2 的其他框图类型”。探讨 OOD/UML 案例研究中没有涉及的其他 UML 2 框图类型。

编译器和其他资源

有许多工具可以用来开发 C++ 程序。本书主要采用 Microsoft 免费的 Visual C++ Express Edition(www.microsoft.com/express/vc/)和免费的 GNU C++ (gcc.gnu.org),后者已经安装在大多数 Linux 系统之中,它同样

也可安装在 Mac OS X 上。苹果公司在它的 Xcode 开发工具中包含了 GNU C++，Mac OS X 用户可以从 developer.apple.com/tools/xcode 下载它。

其他的资源和可以下载的软件，位于作者的 C++ 资源中心：

www.deitel.com/cplusplus/

以及本书的 Web 站点：

www.deitel.com/books/cppfp/

如果想了解其他可以下载的免费 C++ 编译器的情况，可访问：

www.thefreecountry.com/developercity/ccompilers.shtml
www.compilers.net

关于老式 C++ 编译器的警告和错误消息

本书中设计的程序可以用于支持标准 C++ 的编译器。但是，可能有某些编译器会偶尔出现警告或错误。此外，尽管标准指出了必须产生错误的各种情况，但它并没有指明编译器应当发出的错误消息。不同的编译器，会产生不同的警告和错误消息。

一些老式的 C++ 编译器所产生的错误或警告消息，在较新的编译器中并不会出现。尽管本书中的大多数例子都可以采用这些老式的编译器，但有一些例子需要稍作修改才能在这些编译器中编译。

关于 using 声明和 C 标准类库函数的说明

C++ 标准类库包含了来自于 C 标准类库的函数。根据 C++ 标准文档的说明，来自于 C 标准类库头文件的内容，是“std”名字空间的一部分。当遇到的 using 声明用于 C 函数时，有些编译器（旧的和新的）会产生错误消息。

Deitel 的在线资源中心

作者的 Web 站点提供各种主题的资源中心 (www.deitel.com/ResourceCenters.html)，包括编程语言、软件、Web 2.0、Internet 业务和开源工程等。资源中心是为了支持我们的著作和业务需求而建立的。还可以找到许多其他的在线资源（多数是免费的），包括教程、文档、软件下载、文章、博客、视频、代码样本、图书、电子书，等等。资源中心中提供了许多有价值的资源的链接，以帮助读者避开 Internet 上数量庞大的内容。每个星期，作者都会在新闻简报中发布最新的资源中心信息。新闻简报即作者的 Deitel Buzz Online，订阅站点是 www.deitel.com/newsletter/subscribe.html。对于本书的读者而言，可能感兴趣的资源中心如下所示：

- C++
- Visual C++ 2008
- C++ Boost Libraries
- C++ Game Programming
- Code Search Engines and Code Sites
- Computer Game Programming
- Computing Jobs
- Open Source
- Programming Projects
- Eclipse
- Linux
- .NET
- Windows Vista

Deitel Buzz Online 免费的电子邮件新闻简报

每个星期，Deitel Buzz Online 新闻简报都会发布最新的资源中心简报，并带有对行业趋势和发展的评论、已出版和即将出版图书的免费文章和资源的链接、产品发布日程表、勘误、质疑、趣闻、Deitel & Associates 公司课堂培训课程的信息，等等。它也是跟踪与本书相关主题的一种好办法。要订阅电子邮件新闻简报，可以访问：

www.deitel.com/newsletter/subscribe.html

自己掌握进度的 Deitel LiveLessons 视频培训

Deitel LiveLessons 产品是一种由学员自己掌握学习进度的视频培训课程。每个产品都提供了多于 14 小时的课程,对学员进行编程培训。

这些课程的教师 Paul Deitel,在 IBM、Sun Microsystems、NASA 等机构从事过编程培训。通过 LiveLessons 产品中功能强大的视频,学员可按自己的进度学习编程基础、面向对象编程以及其他的课题。

Deitel LiveLessons 产品以它对应的畅销书以及 Paul 在数百场企业培训讲座中的经验为基础。如果想观看视频样本,可以访问:

www.deitel.com/books/livelessons/

现在可观看的视频是 Java Fundamentals I 和 Java Fundamentals II。关于即将发布的 Deitel LiveLessons 产品的信息,包括 C++ Fundamentals、C# 2008 Fundamentals 和 JavaScript Fundamentals,可订阅 Deitel Buzz Online 电子邮件新闻简报,见 www.deitel.com/newsletter/subscribe.html。

Deitel Dive-Into 系列课堂讲授的培训

利用作者的公司、网站和课堂讲授的 Dive-Into 系列编程培训课程(见图 2),通过 Deitel & Associates 公司里国际知名专家的讲授,专业人士可学习 C++、Java、C、Visual Basic、Visual C#、Visual C++、Python 以及 Internet 和 Web 编程。我们的作者、授课教师以及聘请的教师,已经在 100 多个国家为 100 万名学员讲授过几乎每一种主要的编程语言,他们的教学方法包括:

- “程序员教程”系列专业图书
- “大学教程”系列教材
- 在大学讲课
- 专业技术讲座
- 交互式多媒体 CD-ROM Cyber Classrooms、全面的培训课程以及 LiveLessons 视频培训
- 卫星广播

我们的唯一使命是使非程序员变成程序员、使专业程序员精通新的编程语言。关于作者的网上培训信息、课堂讲授的 Dive-Into 系列编程培训等信息,请访问:

www.deitel.com/training/

Deitel Dive Into 系列编程培训课程

Java

Java 简介,非程序员适用: 1

Java 简介,非程序员适用: 2

适用于 Visual Basic、C 或 COBOL 程序员的 Java

适用于 C++ 或 C# 程序员的 Java

高级 Java

C++

C++ 简介,非程序员适用: 1

C++ 简介,非程序员适用: 2

C++ 与面向对象编程

C

C 简介,非程序员适用: 1

C 简介,非程序员适用: 2

适合于程序员的 C

图 2 Deitel Dive Into 系列编程培训课程