



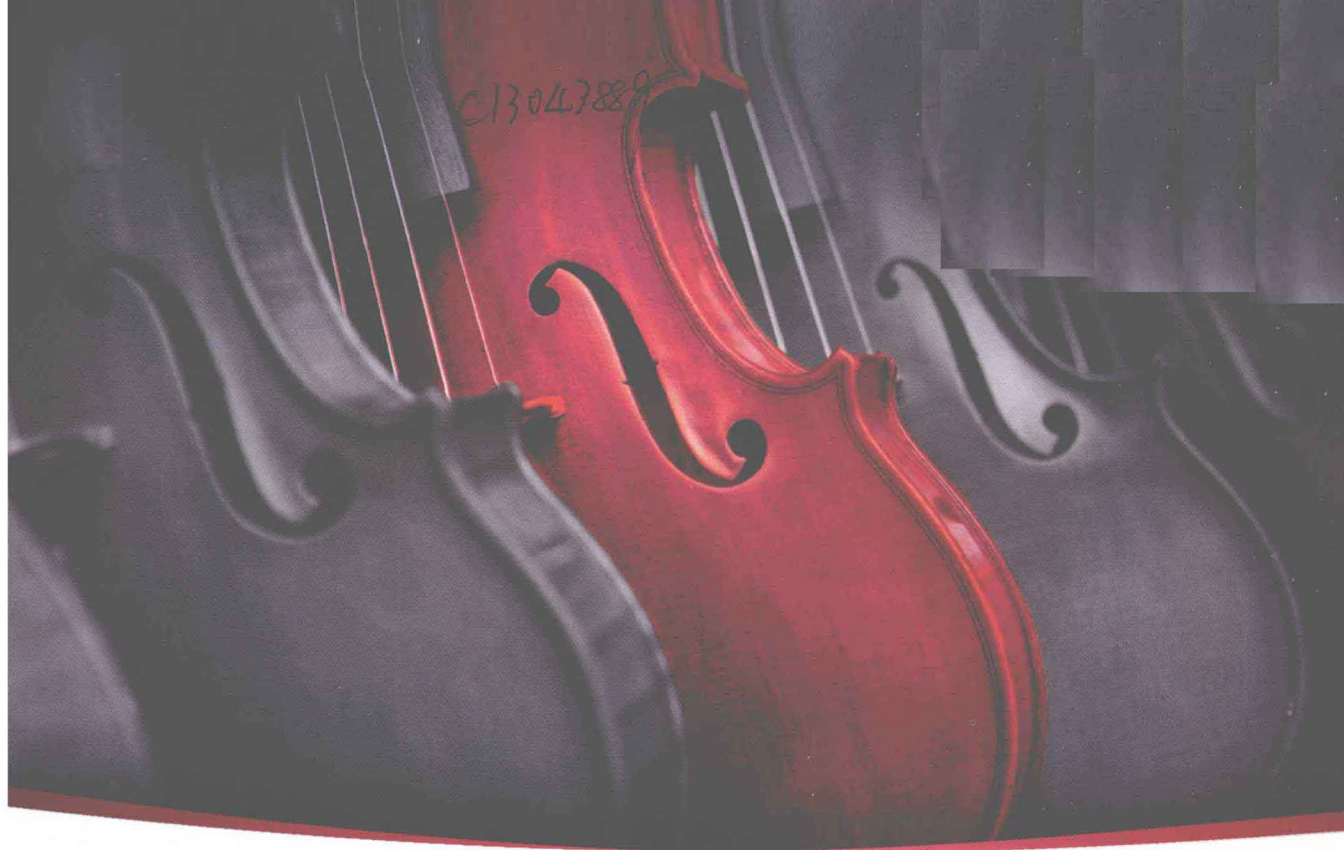
Ivor Horton's Beginning Visual C++ 2012

Visual C++ 2012 入门经典(第6版)

[美] Ivor Horton 著
苏正泉 李文娟 译



清华大学出版社



Visual C++ 2012 入门经典

(第 6 版)

[美] Ivor Horton 著
苏正泉 李文娟 译

清华大学出版社

Ivor Horton

Ivor Horton's Beginning Visual C++ 2012

EISBN: 978-1-1183-6808-4

Copyright © 2012 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: 01-2012-8287

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

Visual C++ 2012 入门经典(第 6 版)/(美) 霍尔顿(Horton, I.) 著; 苏正泉, 李文娟 译.
—北京: 清华大学出版社, 2013.5

书名原文: Ivor Horton's Beginning Visual C++ 2012

ISBN 978-7-302-31900-9

I. ①V… II. ①霍… ②苏… ③李… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 074802 号

责任编辑: 王 军 于 平

装帧设计: 牛静敏

责任校对: 成凤进

责任印制: 杨 艳

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 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: 清华大学印刷厂

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 48 插 页: 1 字 数: 1321 千字

版 次: 2013 年 5 月第 1 版 印 次: 2013 年 5 月第 1 次印刷

印 数: 1~5000

定 价: 98.00 元

产品编号: 050272-01

作者简介

Ivor Horton 原来是一位数学家，却因向往信息技术工作轻松而收入丰厚，因而涉足信息技术领域。尽管现实情况常常是工作辛苦而收入却相对不高，但他仍坚持从事计算机工作至今。在不同的时期，他从事过的工作包括程序设计、系统设计、顾问工作以及管理和实现相当复杂的项目。

Horton 在计算机系统的设计和实现方面，拥有多年的工作经验，这些系统应用于多种行业的工程设计和制造运营。他不仅能运用多种编程语言开发特殊用途的应用程序，而且还为科研人员和工程人员提供教学，以帮助他们完成这类工作，在这些方面他都拥有相当丰富的经验。他多年来一直从事程序设计方面书籍的撰写工作，目前出版的著作有 C、C++ 和 Java 等教程。目前，他既没有忙于写书，也不提供咨询服务，而是在钓鱼、旅游和尽情地享受生活。

技术编辑简介

Marc Gregoire 是来自比利时的一位软件工程师。他毕业于比利时天主教鲁汶大学，获得了“Burgerlijk ingenieur in de computer wetenschappen”学位(等同于计算机科学工程的科学硕士学位)。此后，他以优异成绩获得了同一所大学的人工智能硕士学位，并开始供职于一家大型软件咨询公司 (Ordina 公司网址: <http://www.ordina.be>)。他在西门子和诺基亚西门子通信公司为大型电信运营商开发运行于 Solaris 上至关重要的 2G 和 3G 软件，这需要在国际团队中工作，包括南美、USA、EMEA 和亚洲。现在，Marc 在尼康公司开发三维扫描软件。

他主要擅长 C/C++，具体地说就是 Microsoft VC++ 和 MFC framework。除 C/C++ 之外，他也喜欢 C#，并使用 PHP 制作网页。除主要致力于 Windows 开发以外，他在开发全天候运行在 Linux 平台上的 C++ 程序方面，也具有一定的经验，例如 EIB 家用自动控制和监视软件。

由于在 Visual C++ 方面具有杰出的专业技能，Marc Gregoire 自从 2007 年 4 月开始，每年都荣获了 Microsoft MVP (Most Valuable Professional) 大奖。

Marc 不仅是 Belgian C++ 用户组 (www.becpp.org) 的创始人和 CodeGuru 论坛的活跃分子 (会员名是 Marc G)。他制作的免费程序和共享程序通过其站点 www.nuonsoft.com 来发布，并维护其博客 www.nuonsoft.com/blog/。

致 谢

在为本书的出版而付出劳动的所有人员组成的大型团队中，作者只是其中的一员。感谢 John Wiley & Sons 公司和 Wrox 出版社的编辑和生产团队，感谢他们自始至终提供的帮助和支持。

尤其要感谢技术编辑 Marc Gregoire，感谢他审阅本书，并认真核对了书中提供的所有代码段和示例。他对以更好的方式呈现书中内容提出了很多建设性意见和建议，这毫无疑问使本书成为一本更出色的教程。

感谢妻子 Eve 一如既往的爱和支持，这为我在从事其他工作的同时撰写完成本书奠定了基础。感谢妻子为我提供的一切，虽然我的工作负担对家庭生活造成了一定的影响，但妻子对此默默忍耐，并保持愉快。

前言

欢迎使用本书。通过学习本书,您可以使用 Microsoft 公司最新的应用程序开发系统 Visual Studio 2012,成为优秀的 C++程序员。本书旨在讲述 C++程序设计语言,然后讲述如何运用 C++语言开发自己的 Windows 应用程序。在此过程中,读者将了解这一最新 Visual C++版本所提供的很多激动人心的新功能。

Visual C++ 2012 是 Microsoft 开发环境 Visual Studio 2012 的所有版本的一部分,本书提到 Visual C++时,都是指 Visual Studio 2012 包含的 Visual C++ 2012 功能。

0.1 本书读者对象

本书针对任何想要学习如何使用 Visual C++编写在 Microsoft Windows 操作系统下运行的 C++应用程序的读者。阅读本书不需要预先具备任何特定编程语言的知识。如果属于下列 3 种情形之一,您就适合学习本教程:

- 属于编程新手,十分渴望投入编程世界,并最终掌握 C++。要取得成功,您至少需要对计算机的工作原理有大体的理解——包括内存的组织方式以及数据和指令的存储方式。
- 具备一些其他语言的编程经验,如 BASIC;渴望学习 C++,并想提升实际的 Microsoft Windows 编程技能。
- 有一些使用 C 语言或 C++语言的经验,但使用环境不是 Microsoft Windows;希望使用最新的工具和技术,扩展在 Windows 环境下编程的技能。

0.2 本书主要内容

本书的第一部分通过一个详细的循序渐进式教程,讲授了使用 Visual Studio 2012 支持的 C++语言技术编写 C++程序的基础知识。您将了解 ISO/IEC C++语言的语法和用法,并通过一系列范围广泛的可行示例,获得实际运用它的经验和信心。本书也提供了一些练习,可以检验所学的知识,并且可以下载练习题答案。

当然,本语言教程也介绍和说明了 C++标准库功能的用法,因为开发程序时极有可能使用它们。随着深入地学习 C++语言,您的标准库知识会不断增加。还将学习标准模板库(Standard Template Library, STL)提供的强大工具。

对 C++的运用有信心之后,就可以继续学习 Windows 编程了。通过创建超过 2000 行代码的大型可运行的应用程序,学习如何使用 MFC 来开发本地 Windows 应用程序。开发此应用程序贯穿多章内容,用到了 MFC 提供的一系列用户界面功能。还要学习如何使用 Windows 8 UI 编写应用程序的基础知识,并开发了一个示例。

0.3 本书结构

本书内容的结构安排如下：

- 第 1 章介绍使用 C++ 编写程序所需要理解的基本概念，以及在 Visual C++ 开发环境中体现的主要思想，还叙述了如何使用 Visual C++ 的功能来创建本书其余部分要学习的各种 C++ 应用程序。
- 第 2~9 章讲授 C++ 语言。首先是简单的过程式程序示例，然后学习类和面向对象的编程。第 10 章介绍如何使用标准模板库(Standard Template Library, STL)。STL 是一组功能强大且全面的工具，用来组织和操作 C++ 程序中的数据。由于 STL 是独立于应用程序的，因此可以在上下文中大量应用它。
- 第 11 章讨论 Microsoft Windows 桌面应用程序的组织方式，并描述和展示了在所有为 Windows 操作系统编写的桌面应用程序中都存在的基本元素。本章解释了以 C++ 语言编写的、使用 Windows API 和 MFC 的 Windows 应用程序示例，还给出了一个 Windows 应用程序的基础示例。
- 第 12~17 章讲述 Windows 桌面应用程序的编程。详细描述了如何使用 MFC 提供的构建 GUI 的功能编写 C++ Windows 应用程序。我们将学习如何创建并使用通用控件来构建应用程序的图形用户界面，还将学习如何处理因用户与程序的交互作用而产生的事件。除了学习构建 GUI 的技术以外，还将从开发该应用程序的过程中学到如何打印文档，以及如何在磁盘上保存应用程序数据。
- 第 18 章讲述为 Windows 8 编写应用程序的基本概念。

本书各章内容都包括许多工作示例，通过这些示例阐明所讨论的编程技术。每章结束时都总结了该章所讲述的要点，大多数章节都在最后给出了一组练习，您可以应用所学的技术来试着解答这些练习。练习的答案连同书中的所有代码都可以从 Wrox 出版社的网站上下载。

0.4 使用本书的前提

Visual Studio 2012 有几个版本，它们都有不同的功能。只有安装了 Visual Studio 2012 和 Windows 8，才能使用 Windows 8 UI 开发应用程序。下面是使用本书各部分的要求：

如果安装了 Visual Studio 11 的免费版本和 Windows 8，才能使用本书的所有示例和练习。Visual Studio 2012 最便宜的付费版本是专业版。这个版本或更高版本及 Windows 8 比较适合学习本书。

如果安装了 Visual Studio 2012 专业版或更高版本和 Windows 7，就可以使用第 1~17 章的使用示例和练习，但第 18 章不行。第 18 章介绍了使用 Windows 8 UI 的应用程序，这需要安装 Visual Studio 2012 和 Windows 8。

Windows 8 有一个免费的 Visual Studio 2012 Express 版本，但它不足以运行本书的许多示例。第 12~17 章的例子都不能用 Visual Studio 2012 Express 版本编译。

0.5 源代码

读者在阅读本书提供的代码时，既可以亲自输入所有代码，也可以使用随书提供的代码文件。本书所有代码均可以从 <http://www.wrox.com/> 或 <http://www.tupwk.com.cn/downpage> 网站下载。进入该网站后，读者可以根据本书的书名查找本书(既可以使用搜索框，也可以使用书名列表进行查找)，然后单击本书详细内容页面上提供的 Download Code 链接，就可以下载本书提供的所有代码。

注意：

由于许多书籍名称与本书类似，读者也可以通过 ISBN 进行查找，本书的 ISBN 为：978-0-111-36808-4。

另外，读者可以从前面提到的 CodePlex 网站下载本书或其他 Wrox 书籍的代码，也可以从 Wrox 的代码下载页面 <http://www.wrox.com/dynamic/books/download.aspx> 和 <http://www.tupwk.com.cn/downpage> 下载本书或其他 Wrox 书籍的代码。

源代码下载成功后，读者用任一解压工具将其解压即可。

0.6 勘误表

为了避免本书文字和代码中存在错误，我们已经竭尽全力。然而，世界上并不存在完美无缺的事物，所以本书可能仍然存在错误。如果读者在我们编写的某本书籍中发现了诸如拼写错误或代码缺陷等问题，那么请告诉我们，我们对此表示感谢。利用勘误表反馈错误信息，可以为其他读者节省大量时间，同时，我们也能够受益于读者的帮助，这样有助于我们编写出质量更高的专业著作。

如果读者需要参考本书的勘误表，请在网站 <http://www.wrox.com> 中用搜索框或书名列表查找本书书名。然后，在本书的详细内容页面上，单击 Book Errata 链接。在随后显示的页面中，读者可以看到与本书相关的所有勘误信息，这些信息是由读者提交、并由 Wrox 的编辑们加上的。通过访问 <http://www.wrox.com/misc-pages/booklist.shtml>，读者还可以看到 Wrox 出版的所有书籍的勘误表。

如果读者没有在 Book Errata 页面上找到自己发现的错误，那么请转到页面 <http://www.wrox.com/contact/techsupport.shtml>，针对您所发现的每一项错误填写表格，并将表格发给我们，我们将对表格内容进行认真审查，如果确实是我们书中的错误，我们将在该书的 Book Errata 页面上标明该错误信息，并在该书的后续版本中改正。

0.7 关于 p2p.wrox.com 论坛

如果读者希望能够与作者进行讨论，或希望能够参与到读者的共同讨论中，那么请加入 p2p.wrox.com 论坛。该论坛是一个基于 Web 的系统，读者可以在论坛发表与 Wrox 出版的书籍及相关技术的信息，并与其他读者和技术用户进行讨论。论坛提供了订阅功能，可以将与读者所选定主题相关的新帖子定期发送到读者的电子邮箱。Wrox 的作者、编辑、业界专家，以及其他读者都会参与论坛中的讨论。

读者可以在 <http://p2p.wrox.com> 参与多个论坛的讨论, 这些论坛不仅能够帮助读者更好地理解本书, 还有助于读者更好地开发应用程序。如果读者希望加入论坛, 那么请按照以下步骤执行:

- (1) 进入 <http://p2p.wrox.com> 页面, 单击 **Register** 链接。
- (2) 阅读使用条款, 然后单击 **Agree** 按钮。
- (3) 填写必要的信息及可选信息, 然后单击 **Submit** 按钮。
- (4) 随后读者会收到一封电子邮件, 邮件中说明了如何验证账户并完成整个加入过程。

读者无须加入 P2P 论坛即可阅读论坛消息, 但如果需要发表主题或发表回复, 那么必须加入论坛。

成功加入论坛后, 读者就可以发表新主题了。此时, 读者还可以回复其他用户发表的主题。读者在任何时间都可以阅读论坛信息, 如果需要论坛将新的信息发送到自己的电子邮箱, 那么可以单击论坛列表中论坛名称旁的 **Subscribe to this Forum** 图标完成这项功能设置。

如果读者需要获得更多与 Wrox P2P 相关的信息, 请阅读 P2P FAQs, 这样可以获得大量与 P2P 和 Wrox 出版的书籍相关的具体信息。阅读 FAQs 时, 请单击 P2P 页面上的 FAQs 链接。

目 录

第 1 章 使用 Visual C++编程	1	2.3.1 整型变量	35
1.1 使用 Visual C++学习	1	2.3.2 字符数据类型	36
1.2 编写 C++应用程序	2	2.3.3 整型修饰符	37
1.3 学习桌面应用程序的编程	2	2.3.4 布尔类型	38
1.3.1 学习 C++	3	2.3.5 浮点类型	38
1.3.2 控制台应用程序	3	2.3.6 C++中的基本类型	39
1.3.3 Windows 编程概念	3	2.3.7 字面值	39
1.4 集成开发环境简介	5	2.3.8 定义数据类型的同义词	40
1.4.1 编辑器	5	2.4 基本的输入/输出操作	41
1.4.2 编译器	5	2.4.1 从键盘输入	41
1.4.3 链接器	5	2.4.2 到命令行的输出	41
1.4.4 库	6	2.4.3 格式化输出	42
1.5 使用 IDE	6	2.4.4 转义序列	43
1.5.1 工具栏选项	7	2.5 C++中的计算	45
1.5.2 可停靠的工具栏	8	2.5.1 赋值语句	45
1.5.3 文档	8	2.5.2 算术运算	45
1.5.4 项目和解决方案	8	2.5.3 计算余数	50
1.5.5 设置 Visual C++的选项	16	2.5.4 修改变量	50
1.5.6 创建和执行 Windows 应用程序	17	2.5.5 增量和减量运算符	51
1.6 小结	19	2.5.6 计算的顺序	53
1.7 本章主要内容	19	2.6 类型转换和类型强制转换	54
第 2 章 数据、变量和计算	21	2.6.1 赋值语句中的类型转换	55
2.1 C++程序结构	21	2.6.2 显式类型转换	56
2.1.1 main()函数	28	2.6.3 老式的类型强制转换	57
2.1.2 程序语句	28	2.7 AUTO 关键字	57
2.1.3 空白	30	2.8 类型的确定	58
2.1.4 语句块	30	2.9 按位运算符	58
2.1.5 自动生成的控制台程序	31	2.9.1 按位 AND 运算符	58
2.2 定义变量	32	2.9.2 按位 OR 运算符	60
2.2.1 命名变量	32	2.9.3 按位 EOR 运算符	61
2.2.2 声明变量	33	2.9.4 按位 NOT 运算符	61
2.2.3 变量的初始值	34	2.9.5 移位运算符	61
2.3 基本数据类型	34	2.10 lvalue 和 rvalue	63
		2.11 了解存储时间和作用域	64

2.11.1 自动变量	64	4.1.5 字符数组和字符串处理	119
2.11.2 决定变量声明的位置	66	4.1.6 多维数组	122
2.11.3 全局变量	67	4.2 间接数据访问	125
2.11.4 静态变量	70	4.2.1 指针的概念	125
2.12 具有特定值集的变量	70	4.2.2 声明指针	125
2.12.1 旧枚举	70	4.2.3 使用指针	126
2.12.2 类型安全的枚举	72	4.2.4 初始化指针	127
2.13 名称空间	74	4.2.5 sizeof 操作符	132
2.13.1 声明名称空间	75	4.2.6 常量指针和指向常量的指针	134
2.13.2 多个名称空间	76	4.2.7 指针和数组	136
2.14 小结	77	4.3 动态内存分配	142
2.15 练习	77	4.3.1 堆的别名——空闲存储器	142
2.16 本章主要内容	78	4.3.2 new 和 delete 操作符	142
第3章 判断和循环	79	4.3.3 为数组动态分配内存	143
3.1 比较数据值	79	4.3.4 多维数组的动态分配	146
3.1.1 if 语句	80	4.4 使用引用	146
3.1.2 嵌套的 if 语句	81	4.4.1 引用的概念	147
3.1.3 嵌套的 if-else 语句	85	4.4.2 声明并初始化 lvalue 引用	147
3.1.4 逻辑运算符和表达式	87	4.4.3 在基于范围的 for 循环中使用引用	148
3.1.5 条件运算符	89	4.4.4 rvalue 引用	148
3.1.6 switch 语句	91	4.5 字符串的库函数	149
3.1.7 无条件转移	94	4.5.1 确定以空字符结尾的字符串的长度	149
3.2 重复执行语句块	95	4.5.2 连接以空字符结尾的字符串	150
3.2.1 循环的概念	95	4.5.3 复制以空字符结尾的字符串	151
3.2.2 for 循环的变体	97	4.5.4 比较以空字符结尾的字符串	152
3.2.3 while 循环	104	4.5.5 搜索以空字符结尾的字符串	152
3.2.4 do-while 循环	106	4.6 小结	154
3.2.5 基于范围的循环	107	4.7 练习	155
3.2.6 嵌套的循环	107	4.8 本章主要内容	155
3.3 小结	110	第5章 程序结构(1)	157
3.4 练习	110	5.1 理解函数	157
3.5 本章主要内容	111	5.1.1 需要函数的原因	158
第4章 数组、字符串和指针	113	5.1.2 函数的结构	158
4.1 处理多个相同类型的数据值	113	5.1.3 替代的函数语法	161
4.1.1 数组	113	5.1.4 使用函数	161
4.1.2 声明数组	114	5.2 给函数传递实参	164
4.1.3 初始化数组	117		
4.1.4 使用基于范围的 for 循环	118		

5.2.1 按值传递机制	165	6.8.4 获得项值	218
5.2.2 给函数传递指针实参	166	6.8.5 分析数	219
5.2.3 给函数传递数组	167	6.8.6 整合程序	221
5.2.4 给函数传递引用实参	171	6.8.7 扩展程序	223
5.2.5 使用 const 修饰符	173	6.8.8 提取子字符串	224
5.2.6 rvalue 引用形参	174	6.8.9 运行修改过的程序	226
5.2.7 main()函数的实参	176	6.9 小结	227
5.2.8 接受数量不定的函数实参	177	6.10 练习	227
5.3 从函数返回值	179	6.11 本章主要内容	228
5.3.1 返回指针	179		
5.3.2 返回引用	182	第 7 章 自定义数据类型	229
5.3.3 函数中的静态变量	184	7.1 C++中的结构	229
5.4 递归函数调用	186	7.1.1 结构的概念	230
5.5 小结	189	7.1.2 定义结构	230
5.6 练习	189	7.1.3 初始化结构	230
5.7 本章主要内容	189	7.1.4 访问结构的成员	231
		7.1.5 伴随结构的智能感知帮助	234
第 6 章 程序结构(2)	191	7.1.6 RECT 结构	235
6.1 函数指针	191	7.1.7 使用指针处理结构	236
6.1.1 声明函数指针	191	7.2 数据类型、对象、类和实例	237
6.1.2 函数指针作为实参	194	7.2.1 类的起源	239
6.1.3 函数指针的数组	196	7.2.2 类的操作	239
6.2 初始化函数形参	196	7.2.3 术语	240
6.3 异常	198	7.3 理解类	240
6.3.1 抛出异常	199	7.3.1 定义类	240
6.3.2 捕获异常	200	7.3.2 声明类的对象	241
6.3.3 重新抛出异常	201	7.3.3 访问类的数据成员	241
6.3.4 MFC 中的异常处理	202	7.3.4 类的成员函数	243
6.4 处理内存分配错误	203	7.3.5 成员函数定义的位置	245
6.5 函数重载	204	7.3.6 内联函数	245
6.5.1 函数重载的概念	204	7.4 类构造函数	246
6.5.2 引用类型和重载选择	207	7.4.1 构造函数的概念	247
6.5.3 何时重载函数	207	7.4.2 默认的构造函数	248
6.6 函数模板	208	7.4.3 默认的形参值	250
6.7 使用 decltype 操作符	210	7.4.4 在构造函数中使用初始化列表	252
6.8 使用函数的示例	212	7.4.5 声明显式的构造函数	253
6.8.1 实现计算器	212	7.5 类的私有成员	254
6.8.2 从字符串中删除空格	215	7.5.1 访问私有类成员	256
6.8.3 计算表达式的值	216	7.5.2 类的友元函数	257

7.5.3 默认复制构造函数	259	8.7.1 定义类模板	316
7.6 this 指针	260	8.7.2 根据类模板创建对象	318
7.7 类的 const 对象	263	8.7.3 使用有多个形参的类模板	321
7.7.1 类的 const 成员函数	263	8.7.4 函数对象模板	323
7.7.2 类外部的成员函数定义	264	8.8 完美转发	324
7.8 类对象的数组	265	8.9 使用类	327
7.9 类的静态成员	267	8.9.1 类接口的概念	327
7.9.1 类的静态数据成员	267	8.9.2 定义问题	327
7.9.2 类的静态函数成员	270	8.9.3 实现 CBox 类	328
7.10 类对象的指针和引用	270	8.10 组织程序代码	343
7.10.1 类对象的指针	270	8.11 字符串的库类	345
7.10.2 类对象的引用	273	8.11.1 创建字符串对象	345
7.11 小结	274	8.11.2 连接字符串	346
7.12 练习	274	8.11.3 访问与修改字符串	350
7.13 本章主要内容	275	8.11.4 比较字符串	353
8.11.5 搜索字符串	356	8.12 小结	364
第8章 深入理解类	277	8.13 练习	364
8.1 类析构函数	277	8.14 本章主要内容	365
8.1.1 析构函数的概念	277	第9章 类继承和虚函数	367
8.1.2 默认的析构函数	278	9.1 面向对象编程的基本思想	367
8.1.3 析构函数与动态内存分配	280	9.2 类的继承	368
8.2 实现复制构造函数	283	9.2.1 基类的概念	369
8.3 在变量之间共享内存	284	9.2.2 基类的派生类	369
8.3.1 定义联合	285	9.3 继承机制下的访问控制	372
8.3.2 匿名联合	286	9.3.1 派生类中构造函数的操作	375
8.3.3 类和结构中的联合	286	9.3.2 声明类的保护成员	378
8.4 运算符重载	287	9.3.3 继承类成员的访问级别	380
8.4.1 实现重载的运算符	287	9.4 派生类中的复制构造函数	382
8.4.2 实现对比较运算符的完全支持	290	9.5 禁止派生类	384
8.4.3 重载赋值运算符	294	9.6 友元类成员	385
8.4.4 重载加法运算符	299	9.6.1 友元类	387
8.4.5 重载递增和递减运算符	303	9.6.2 对类友元关系的限制	387
8.4.6 重载函数调用操作符	304	9.7 虚函数	387
8.5 对象复制问题	305	9.7.1 虚函数的概念	389
8.5.1 避免不必要的复制操作	305	9.7.2 确保虚函数的正确执行	391
8.5.2 应用 rvalue 引用形参	308	9.7.3 禁止重写函数	391
8.5.3 命名的对象是 lvalue	310	9.7.4 使用指向类对象的指针	392
8.6 默认类成员	314		
8.7 类模板	315		

9.7.5 使用引用处理虚函数	393	10.8.3 使用输出流迭代器	485
9.7.6 纯虚函数	395	10.9 关于函数对象的更多内容	487
9.7.7 抽象类	395	10.10 关于算法的更多内容	488
9.7.8 间接基类	398	10.10.1 fill()	489
9.7.9 虚析构函数	400	10.10.2 replace()	489
9.8 类类型之间的强制转换	403	10.10.3 find()	489
9.9 嵌套类	403	10.10.4 transform()	490
9.10 小结	407	10.11 类型特质和静态断言	491
9.11 练习	407	10.12 λ 表达式	492
9.12 本章主要内容	409	10.12.1 capture 子句	493
第 10 章 标准模板库	411	10.12.2 捕获特定的变量	494
10.1 标准模板库的定义	411	10.12.3 模板和 λ 表达式	494
10.1.1 容器	412	10.12.4 包装 λ 表达式	498
10.1.2 容器适配器	414	10.13 小结	500
10.1.3 迭代器	414	10.14 练习	500
10.2 智能指针	415	10.15 本章主要内容	501
10.3 算法	418	第 11 章 Windows 编程的概念	503
10.4 STL 中的函数对象	418	11.1 Windows 编程基础	503
10.5 STL 容器范围	419	11.1.1 窗口的元素	504
10.6 序列容器	419	11.1.2 Windows 程序与操作系统	505
10.6.1 创建矢量容器	420	11.1.3 事件驱动型程序	505
10.6.2 矢量容器的容量和大小	423	11.1.4 Windows 消息	506
10.6.3 访问矢量中的元素	428	11.1.5 Windows API	506
10.6.4 在矢量中插入和删除元素	428	11.1.6 Windows 数据类型	506
10.6.5 在矢量中存储类对象	431	11.1.7 Windows 程序中的符号	507
10.6.6 排序矢量元素	436	11.2 Windows 程序的结构	508
10.6.7 排序矢量中的指针	437	11.2.1 WinMain() 函数	509
10.6.8 双端队列容器	442	11.2.2 消息处理函数	519
10.6.9 使用列表容器	445	11.3 MFC	524
10.6.10 使用 forward_list 容器	454	11.3.1 MFC 表示法	524
10.6.11 使用其他序列容器	456	11.3.2 MFC 程序的组织方式	525
10.6.12 tuple<> 类模板	466	11.4 小结	528
10.7 关联容器	469	11.5 本章主要内容	528
10.7.1 使用映射容器	469	第 12 章 使用 MFC 编写 Windows 程序	531
10.7.2 使用多重映射容器	480	12.1 MFC 的文档/视图概念	531
10.8 关于迭代器的更多内容	481	12.1.1 文档的概念	531
10.8.1 使用输入流迭代器	481	12.1.2 文档界面	532
10.8.2 使用插入迭代器	484	12.1.3 视图的概念	532

12.1.4 链接文档和视图	533	14.4.1 鼠标发出的消息	587
12.1.5 应用程序和 MFC	534	14.4.2 鼠标消息处理程序	588
12.2 创建 MFC 应用程序	535	14.4.3 使用鼠标绘图	590
12.2.1 创建 SDI 应用程序	536	14.5 绘制草图	611
12.2.2 MFC Application Wizard 的输出	539	14.5.1 运行示例	612
12.2.3 创建 MDI 应用程序	548	14.5.2 捕获鼠标消息	612
12.3 小结	549	14.6 小结	613
12.4 练习	550	14.7 练习题	613
12.5 本章主要内容 0	550	14.8 本章主要内容	614
第 13 章 处理菜单和工具栏	551	第 15 章 改进视图	615
13.1 与 Windows 进行通信	551	15.1 Sketcher 应用程序的缺陷	615
13.1.1 了解消息映射	552	15.2 改进视图	616
13.1.2 消息类别	554	15.2.1 更新多个视图	616
13.1.3 处理程序中的消息	554	15.2.2 滚动视图	617
13.2 扩展 Sketcher 程序	555	15.2.3 使用 MM_LOENGLISH 映射模式	622
13.3 菜单的元素	556	15.3 删除和移动元素	622
13.4 为菜单消息添加处理程序	559	15.4 实现上下文菜单	623
13.4.1 选择处理菜单消息的类	560	15.4.1 关联菜单和类	624
13.4.2 创建菜单消息函数	560	15.4.2 选中上下文菜单项	625
13.4.3 编写菜单消息函数的代码	562	15.5 标识位于光标下的元素	626
13.4.4 添加更新菜单消息的处理 程序	565	15.5.1 练习弹出菜单	627
13.5 添加工具栏按钮	568	15.5.2 突出显示元素	627
13.5.1 编辑工具栏按钮的属性	569	15.5.3 实现移动和删除功能	631
13.5.2 练习使用工具栏按钮	570	15.6 处理屏蔽的元素	637
13.5.3 添加工具提示	571	15.7 小结	639
13.6 小结	571	15.8 练习	639
13.7 练习	571	15.9 本章主要内容	639
13.8 本章主要内容	571	第 16 章 使用对话框和控件	641
第 14 章 在窗口中绘图	573	16.1 理解对话框	641
14.1 窗口绘图的基础知识	573	16.2 理解控件	642
14.1.1 窗口工作区	573	16.3 创建对话框资源	642
14.1.2 Windows 图形设备界面	574	16.3.1 给对话框添加控件	643
14.2 MFC 的绘图机制	576	16.3.2 测试对话框	644
14.2.1 应用程序中的视图类	576	16.4 对话框的编程	644
14.2.2 CDC 类	577	16.4.1 添加对话框类	644
14.3 实际绘制图形	585	16.4.2 模态和非模态对话框	645
14.4 对鼠标进行编程	587	16.4.3 显示对话框	646

16.5 支持对话框控件	648	17.2.5 如何实现类的序列化	685
16.5.1 初始化对话框控件	648	17.3 应用序列化	685
16.5.2 处理单选按钮消息	649	17.3.1 记录文档修改	686
16.6 完成对话框的操作	650	17.3.2 序列化文档	687
16.6.1 给文档添加线宽	651	17.3.3 序列化元素类	689
16.6.2 给元素添加线宽	651	17.4 练习序列化	693
16.6.3 在视图中创建元素	653	17.5 打印文档	694
16.6.4 练习使用对话框	654	17.6 实现多页打印	697
16.7 使用微调按钮控件	655	17.6.1 获取文档的总尺寸	698
16.7.1 添加 Scale 菜单项和工具栏 按钮	655	17.6.2 存储打印数据	698
16.7.2 创建微调按钮	655	17.6.3 准备打印	699
16.7.3 生成比例对话框类	656	17.6.4 打印后的清除	700
16.7.4 显示微调按钮	659	17.6.5 准备设备上下文	701
16.8 使用缩放比例	660	17.6.6 打印文档	701
16.8.1 可缩放的映射模式	660	17.6.7 获得文档的打印输出	705
16.8.2 设置文档的大小	661	17.7 小结	705
16.8.3 设置映射模式	662	17.8 练习	705
16.8.4 同时实现滚动与缩放	663	17.9 本章主要内容	706
16.9 使用状态栏	665	第 18 章 编写 Windows 8 应用程序	707
16.9.1 给框架窗口添加状态栏	665	18.1 理解 Windows 8 应用程序	707
16.9.2 CString 类	669	18.2 开发 WINDOWS 8 应用程序	708
16.10 使用编辑框控件	669	18.3 Windows Runtime 的概念	709
16.10.1 创建编辑框资源	670	18.3.1 WinRT 名称空间	709
16.10.2 创建对话框类	671	18.3.2 WinRT 对象	709
16.10.3 添加 Text 菜单项	672	18.4 C++ COMPONENT EXTENSIONS (C++/CX)	710
16.10.4 定义文本元素	672	18.4.1 C++/CX 名称空间	710
16.10.5 实现 CText 类	673	18.4.2 定义 WinRT 类类型	711
16.11 小结	677	18.4.3 Ref 类类型的变量	713
16.12 练习	678	18.4.4 访问 ref 类对象的成员	713
16.13 本章主要内容	678	18.4.5 事件处理函数	714
第 17 章 存储和打印文档	679	18.4.6 转换 ref 类引用的类型	714
17.1 了解序列化	679	18.5 XAML	714
17.2 序列化文档	680	18.5.1 XAML 元素	715
17.2.1 文档类定义中的序列化	680	18.5.2 XAML 中的 UI 元素	716
17.2.2 文档类实现中的序列化	681	18.5.3 附加属性	719
17.2.3 基于 CObject 的类的功能	683	18.5.4 父元素和子元素	719
17.2.4 序列化的工作方式	684	18.5.5 控件元素	719

18.5.6	布局元素	720	18.6.11	洗牌	740
18.5.7	处理 UI 元素的事件	720	18.6.12	突出显示 UI 纸牌	741
18.6	创建 Windows 8 应用程序	721	18.6.13	处理翻牌事件	741
18.6.1	应用程序文件	721	18.6.14	处理图形事件	743
18.6.2	定义用户界面	722	18.6.15	确认赢家	745
18.6.3	创建标题	724	18.6.16	处理游戏控件的按钮事件	746
18.6.4	添加游戏控件	726	18.7	缩放 UI 元素	747
18.6.5	创建包含纸牌的网格	727	18.8	平移	749
18.6.6	实现游戏的操作	732	18.8.1	应用程序的启动动画	749
18.6.7	初始化 MainPage 对象	735	18.8.2	故事板动画	750
18.6.8	初始化一副纸牌	736	18.9	小结	752
18.6.9	建立 cardGrid 的子元素	736	18.10	本章主要内容	752
18.6.10	初始化游戏	738			