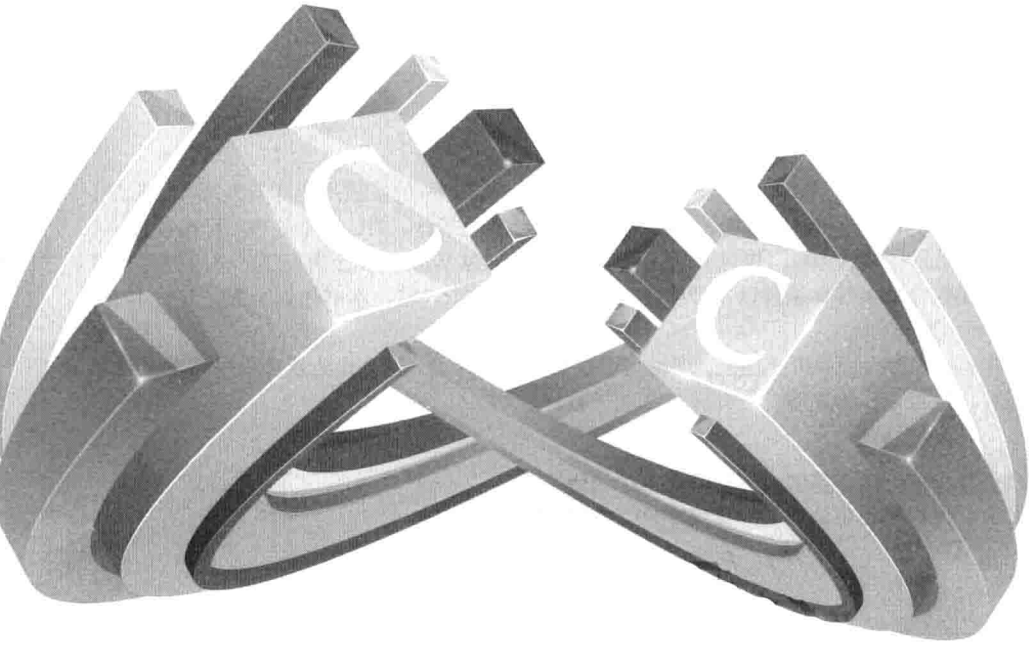
董志鹏
侯艳书 编著

Visual C++编程
从基础到应用

清华大学出版社



董志鹏
侯艳书 编著

Visual C++编程
从基础到应用

清华大学出版社
北京

内 容 简 介

本书全面介绍了C++的相关知识，全书共17章。本书不仅包含了C++的发展历史和开发环境、常量、变量、运算符、数据类型、常用流程控制语句、数组、字符串、函数、指针、类、异常处理、标准模板库、对话框，以及菜单栏、工具栏和状态栏，也包含了如何使用MFC创建Windows应用程序、常用控件、串行化和数据库编程，还包含了多媒体编程以及如何在窗口上绘图等多个内容。本书知识全面，案例丰富，可以帮助读者透彻学习C++语言。

本书具有案例丰富、内容全面、指导性强、理论与实践相结合等特点，适合作为软件程序开发人员和设计人员的参考资料，也可以作为高等院校计算机专业的教材。

图书在版编目（CIP）数据

Visual C++编程从基础到应用/董志鹏等编著. —北京：清华大学出版社，2014
从基础到应用
ISBN 978-7-302-32506-2

Ⅰ. ①V…　Ⅱ. ①董…　Ⅲ. ①C语言-程序设计-教材　Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2013）第108077号

责任编辑：夏兆彦
封面设计：胡文航
责任校对：胡伟民
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 装 者：北京密云胶印厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：32　　**字　　数**：799千字
附光盘1张
版　　次：2014年3月第1版　　**印　　次**：2014年3月第1次印刷
印　　数：1～4000
定　　价：59.00元

产品编号：045969-01

FOREWORD

前言

计算机诞生初期，人们要使用计算机时必须用机器语言或汇编语言编写程序。世界上第一种计算机高级语言诞生于 1954 年，它是 FORTRAN 语言。随后出现了多种计算机高级语言，其中使用最广泛、影响最大的便是 BASIC 语言和 C 语言。但是随着软件规模的增大，使用 C 语言编写的应用程序越来越不能够满足广大用户的需求了，C++语言由此诞生。

C++是 Bjarne Stroustrup 在 20 世纪 80 年代早期开发的，它是一种基于 C 语言的面向对象语言，保留了 C 语言原有的优点，并且增加了面向对象机制。顾名思义，C++表示 C 的累加。由于 C++基于 C，所以这两种语言有许多共同的语法和功能，C 中所有低级编程的功能都在 C++中保留了下来。但是，C++比其前身丰富得多，用途也广泛得多。C++对内存管理功能进行了非常大的改进，还具有面向对象的功能，所以 C 在功能上只是 C++的一个子集。C++在适用范围、性能和功能上也是无可匹敌的。因此，目前大多数高性能的应用程序和系统仍使用 C++编写。

本书以渐进的顺序来介绍 C++的相关知识，首先从发展历史介绍，然后再介绍如何创建 Windows 应用程序。

第 1 章　Visual C++预备知识。本章从最基本的编程语言开始介绍，接着对 C++的诞生、语言特点以及与 Java 和 C#之间的关系进行介绍，然后介绍 C++语言的编译器和集成开发环境，最后介绍如何安装开发工具 Visual Studio 2010 和编码的规范内容。

第 2 章　C++基础语法。本章主要介绍 C++的基本语法，包括常量、变量、数据类型、运算符、表达式以及程序文件等相关内容。

第 3 章　流程控制语句结构。流程控制语句是每种语言重要的知识点之一。本章着重介绍 C++中最常用的流程控制语句，如条件语句、循环语句和跳转语句。

第 4 章　数组和字符串。C++中对数组和字符串的处理是最常使用的内容。在本章中，首先介绍一维数组和二维数组的定义、引用和初始化等，然后介绍与字符串相关的处理函数，以及如何使用字符串等知识。

第 5 章　函数。函数往往把一个完整的程序分解成若干个程序模块，每一个模块实现一个特定的功能。本章将详细介绍 C++中的函数，主要包括函数的返回值、如何调用、与数组的关系、函数重载以及模板等内容。

第 6 章　指针与引用。指针是存放数据存储的地址，它本身并不包含数据。与指针具有类似功能的是引用。本章将详细介绍指针和引用的相关知识。

第 7 章　自定义数据类型。本章的内容涉及结构体、共用体、枚举、类的概念，以及对象、类的构造函数和类的析构函数等内容。

第 8 章　类的更多功能及继承与多态。继承和多态是面向对象编程中两个非常重要的概念。继承可以利用另一个类的操作和数据成员来创建新类；多态是在继承的基础上实现的，它可以实现一个方法有不同的操作。

第 9 章　标准模板库和异常处理。它们是 C++语言中非常重要的特性，主要内容包括标准模板库的概念（如 STL 容器、STL 算法和头文件等）、顺序容器、容器适配器、关联容器、迭代容器、算法以及异常处理等知识。

第 10 章　使用 MFC 创建 Windows 应用程序。本章首先从简单的 Windows 应用程序开始介绍，接着介绍 Windows 程序的结构，然后介绍如何使用、创建和管理 MFC，最后对 MFC 中的程序代码进行详细分析。

第 11 章　对话框。对话框在 Windows 应用程序中使用非常广泛，大多数的 Windows 程序都使用它来管理用户输入的数据。本章的内容包括对话框的设计、模态对话框、非模态对话框、属性页式对话框以及通用对话框等。

第 12 章　常用控件。控件在 Windows 程序中最常用到，本章主要包括两部分：传统控件和新型的 Win32 控件。传统控件包含静态控件、按钮控件、滚动条控件、列表框控件以及组合框控件；新型的 Win32 控件包含微调控件、进度条控件、树形视图控件以及列表视图控件等。

第 13 章　菜单栏、工具栏和状态栏。本章将详细对 Windows 应用程序中常见的用户界面元素——菜单栏、工具栏和状态栏进行详细介绍。

第 14 章　文件和串行化。本章涉及文件和串行化两个知识点，包括与文件相关的内容和种类、常用的三种操作方式以及 CFile 类，以及与串行化相关的基本概念、可串行化类和 Serializ()函数等。

第 15 章　数据库编程。本章将详细介绍如何在 C++中对数据库中的数据进行操作，首先从常用的数据库开发技术介绍，然后介绍 ADO 编程技术中所使用到的主要对象，最后介绍如何操作数据库中的数据（如连接数据库和数据查询等）。

第 16 章　在窗口上进行绘图。本章主要介绍如何使用 GDI 技术在 Windows 程序中绘制基本图形，如直线、矩形、饼图、圆和椭圆等。另外还介绍画刷、画笔、坐标系统和映射模式等内容。

第 17 章　多媒体技术。本章主要介绍 Visual Studio 2010 多媒体技术的应用，主要包括音频处理技术、视频处理技术和图像处理技术等。

本书特色

本书是一本完整介绍 C++语言的应用教程，在编写过程中精心设计了丰富的体例，以帮助读者顺利学习本书内容。

- 理论和实践紧密结合

本书通过基本知识—实例应用—习题—实践疑难解答的模式循序渐进，每一个知识点

后面都会对应一个小示例，充分体现了理论和实践紧密结合的思想。

❑ 内容丰富

本书涵盖了实际开发过程中 C++经常遇到的流程控制语句、对话框、窗口绘图以及多媒体编程等方面的热点问题。

❑ 形式新颖

用准确的语言总结概念，用直观的图示演示过程，用详细的注释解释代码，用形象的比喻帮助记忆。

❑ 携带光盘

本书为实践案例配置了视频教学文件，读者可以通过视频文件更加直观地学习 Visual C++的使用知识。

❑ 网站支持

读者在学习或者工作的过程中如果遇到实际问题，可以直接登录 www.itzcn.con 与我们取得联系，作者会在第一时间给予帮助。

❑ 贴心提示

为了方便读者阅读，本书还穿插着一些技巧、提示和注意等小贴士。

读者对象

本书在多家院校成熟教案以及自编教材的基础上整合编写，全面介绍了与 C++相关的知识，具有知识全面、实践案例精彩以及指导性强等特点。本书可以帮助中级读者提高技能，对高级读者也有一定的启发意义。

本书适合以下人员阅读学习：

❑ C++初学者和在校学生
❑ 网站开发人员
❑ 网站维护人员
❑ 各大院校的相关授课老师
❑ 其他 C++的从业人员

除了封面署名人员之外，参与本书编写的人员还有马海军、李海庆、陶丽、王咏梅、康显丽、郝军启、朱俊成、宋强、孙洪叶、袁江涛、张东平、吴鹏、王新伟、刘青凤、汤莉、冀明、王超英、王丹花、闫琰、张丽莉、李卫平、王慧、牛红惠、丁国庆、黄锦刚、李旎、王中行、李志国等。在编写过程中难免会有漏洞，欢迎读者通过我们的网站 www.itzcn.com 与我们联系，帮助我们改正提高。

CONTENTS

目录

第1章 Visual C++预备知识

在众多的计算机语言中，C++语言是举世公认的最优秀编程语言。C++的语法已经成为专业编程语言的标准，其设计原理也贯穿在整个软件设计领域。而且C++还是很多现代语言的始祖，像Java和C#等。因此，如果想成为一名专业的程序开发人员，那么学习C++是最佳选择，它是通往现代编程领域的入口。

本章首先简单介绍编程语言发展的三个阶段、编程语言的发展历史以及执行方式的分析；然后介绍产生C++语言的背景、C++语言的特点以及相关标准等等；接下来讨论开发C++程序常用的编译器和开发环境，并使用Visual C++ 6.0和 Visual Studio 2010各创建一个C++控制台应用程序；最后介绍C++程序的一些编码规范及常用的C++关键字。

通过本章的学习，读者将对C++语言有一个全面、深入地了解，并掌握如何开发一个简单的程序。

本章学习要点：

- 了解计算机语言发展过程中的三个阶段
- 了解编程语言的发展历史及执行方式
- 了解产生C++语言的背景
- 了解C++语言的特点
- 了解C++语言的标准及与Java和C#的关系
- 熟悉常用的C++编译器和集成开发环境
- 掌握Visual C++ 6.0开发C++程序的方法
- 掌握Visual Studio 2010开发C++程序的方法
- 熟悉C++的编码规范和常见关键字

1.1 编程语言概述

目前有很多种编程语言，每一种语言都有其优缺点。除了C++之外，读者一定还听说过Java、BASIC、C#、C和PASCAL等编程语言。所有这些语言都称为高级语言，因为它们可以比较容易地表达出要计算机完成的工作，而且不针对某台计算机。

高级语言中的每个源语句一般映射为几个内部机器指令，低级语言比较接近内部机器指令，通常称为汇编语言。一种汇编语言专门用于一种硬件设计，一般一个汇编指令映射为一个内部机器指令。

1.1.1 计算机语言简介

所谓语言就是一种按照默认的约定，双方进行交流的方式。例如，汉语和英语是人与人之间交流最常用的语言。另外，不同种类的动物之间也有它们的语言。

同样，人和计算机交流信息，也要解决语言问题。需要创建一种计算机和人都能识别的语言，这就是计算机语言。按照计算机语言的发展过程可以分为机器语言、汇编语言和高级语言三类。

1. 机器语言

机器语言是低级语言，也称为二进制代码语言。计算机使用的是由“0”和“1”组成的二进制数代表一串指令来表达计算机的语言。机器语言的特点是，计算机可以直接识别，不需要进行任何的翻译。

2. 汇编语言

汇编语言是面向机器的程序设计语言。为了减轻使用机器语言编程的痛苦，用英文字母或符号串来替代机器语言的二进制码，这样就把不易理解和使用的机器语言变成了汇编语言。这样一来，使用汇编语言就比机器语言便于阅读和理解程序。

3. 高级语言

由于汇编语言依赖于硬件体系，并且该语言中的助记符号数量比较多，所以应用起来仍然不够方便。为了使程序语言能更贴近人类的自然语言，同时又不依赖于计算机硬件，于是产生了高级语言。这种语言，其语法形式类似于英文，并且因为远离对硬件的直接操作，而易于被普通人所理解与使用。

高级语言的发展又经历了以下三个不同的阶段：

1）非结构化语言

非结构化语言是高级语言发展的第一阶段，编程风格比较随意，只要符合语法规则即可。而且也没有严格的规范要求，程序中的流程可以随意跳转。此时，很多开发人员为了追求程序执行的效率而采用了很多“小技巧”，使程序变得难以阅读和维护。早期的 BASIC 和 FORTRAN 等都是属于这个阶段的语言。

2）结构化语言

为了解决非结构化语言带来的问题，提出了结构化程序设计的方法。它规定程序必须由具有良好特性的基本结构（顺序、分支和循环）构成，且程序中的流程不允许随意跳转，总是由上而下顺序执行各个基本结构。

这个阶段编写的程序结构清晰，易于阅读和维护。属于结构化语言的有 BASIC 和 C 等。

3）面向对象语言

以上两种语言都是基于过程的语言，在编写程序时需要具体指定每一个过程的细节。在编写规模较小的程序时，还能应付。但在处理规模较大、业务逻辑复杂的程序时，就显得“力不从心”了。

经过实践的发展，人们又提出了面向对象的程序设计方法。这种方法将实现细节转换为一个个对象，对象则是由数据以及对数据进行的操作组成的。属于面向对象语言的主要有 C++、Java 和 C#。

1.1.2 语言发展简史

下面简单回顾一下编程语言的发展过程。

FORTRAN 是英文 Formula Translator 的缩写，中文为“公式翻译器”。FORTRAN 是世界上最早出现的计算机高级程序设计语言，广泛应用于科学和工程计算领域。1957 年，第一个 FORTRAN 编译器在 IBM704 计算机上实现，并首次成功运行了 FORTRAN 程序。时至今天，FORTRAN 以其特有的功能在数值、科学和工程计算领域发挥着重要作用。

COBOL 是英文 Common Business Oriented Language 的缩写，中文为“面向商业的通用语言”。COBOL 是最早的高级编程语言之一，是世界上第一个商用语言。COBOL 是数据处理领域应用最为广泛的程序设计语言，是第一个广泛使用的高级编程语言。在企业管理中数值计算并不复杂，但数据处理信息量却很大。为专门解决企业管理问题，1959 年由美国的一些计算机用户组织设计了专用于商务处理的计算机语言 COBOL，并于 1961 年由美国数据系统语言协会公布。经不断修改、丰富完善和标准化，COBOL 已发展为多种版本的庞大语言，在财会工作、统计报表、计划编制、情报检索、人事管理等数据管理及商业数据处理领域，都有着广泛的应用。

BASIC 是英文 Beginner's All-purpose Symbolic Instruction Code 的缩写，意思就是“初学者的全方位符式指令代码”，是一种给初学者使用的程序设计语言。BASIC 是一种直译式的编程语言，在完成编写后不须经由编译及链接等手续即可执行，但如果需要单独执行时仍然需要将其建立成可执行文件。BASIC 语言具有简单、易学的基本特性，很快地就普遍流行起来，几乎所有小型、微型家用计算机，甚至部分大型计算机，都有提供使用者以此种语言编写程序。在微型计算机方面，则因为 BASIC 语言可配合微型计算机操作功能的充分发挥，使得 BASIC 早已成为计算机的主要语言之一。

Java 最初开发为 Oak 语言，是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言（以下简称 Java 语言）和 Java 平台的总称。用 Java 实现的 HotJava 浏览器（支持 Java applet）显示了 Java 的魅力：跨平台、动态的 Web、Internet 计算。从此，Java 被广泛接受并推动了 Web 的迅速发展，常用的浏览器现在均支持 Java applet。另一方面，Java 技术也不断更新。2010 年，Oracle 公司收购了 Sun。Java 语言具有很多优秀的特性，使它看起来像 C++，但有很大区别。Java 在移植性方面比 C++好，但执行性能比不上 C++。

C 语言是世界上流行的、使用最广泛的高级程序设计语言之一。它是在 1972 年由美国的 Dennis Ritchie 设计发明的，并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。1978 年后，C 语言已先后被移植到大、中、小及微型计算机上，它可以作为操作系统设计语言，编写系统应用程序，也可以作为应用程序设计语言，编写不依赖计算机硬件的应用程序。它的应用范围广泛，具备很强的数据处理能力，不仅仅是在软件开发上，而且各类科研都需要用到 C 语言。它适于编写系统软件，三维、二维图形和动画，具体应用如单片机以及嵌入式系统开发。目前，C 语言编译器普遍存在于各种不同的操作系统中，如 UNIX、

MS-DOS、Microsoft Windows 及 Linux 等。C 语言的设计影响了许多后来的编程语言，如 C++、Objective-C、Java、C#等。

C++是 Bjarne Stroustrup 在 20 世纪 80 年代早期开发的，是一种基于 C 语言的面向对象语言。顾名思义，C++表示 C 的累加。由于 C++基于 C，所以这两种语言有许多共同的语法和功能，C 中所有低级编程的功能都在 C++保留了下来。但是，C++比其前身丰富得多，用途也广泛得多。C++对内存管理功能进行了非常大的改进，还具有面向对象的功能，所以 C 在功能上只是 C++的一个子集。C++在适用范围、性能和功能上也是其他编程语言无可匹敌的。因此，目前大多数高性能的应用程序和系统仍使用 C++编写。

C#是微软公司在 2000 年 6 月发布的一种新的编程语言。C#是一种安全的、稳定的、简单的、优雅的，由 C 和 C++衍生出来的面向对象的编程语言。它在继承 C 和 C++强大功能的同时去掉了一些它们的复杂特性（如没有宏，以及不允许多重继承）。C#综合了 VB 简单的可视化操作和 C++的高运行效率，以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷地面向组件编程的支持，成为.NET 开发的首选语言。

1.1.3 语言执行方式分析

无论使用哪种编程语言编写出来的程序都是由各个指令或源语句构成的，它们描述了希望计算机执行的动作。这些指令或者源语句都统称为源代码，存储在磁盘的源文件中。任何规模的 C++程序都是由若干个源文件组成的。

编程语言的目的是，与计算机可以执行的程序相比，能够更简单地描述希望计算机执行的动作。计算机只能执行包含机器指令（机器代码）的程序，不能直接执行使用编程语言编写的源代码。要执行这种源代码有两种方式。一种是解释性地执行，即通过一个语言解释器来检查源代码，确定程序应该要做什么，再让计算机完成这些动作。1.1.2 小节介绍的 BASIC 就属于这种方式。

另一种是像 C++使用的编译方式。在执行 C++程序之前必须用另一个程序（编译器）把它转换为机器语言。编译器会检查并分析 C++程序，并生成机器指令，以执行源代码指定的动作。当然解释和编译都不像这里描述的那样简单，但其工作原理就是这样。这两种执行方式如图 1-1 所示。

使用解释性语言的执行过程是间接的。也就是说每次执行程序时，都需要确定源代码的意图。因此，这种语言经编译语言的对应程序的执行速度要慢很多。解释性语言的优点是在运行之前不必等待语言的编译。使用解释性语言，一旦输入代码就可以立即执行程序。任何一种语言要么是解释性的，要么是编译性的，这通常由该语言的设计和用途来决定。例如，1.1.2 小节介绍的 C、Java 和 C#都属于编译性语言。

于是就有了“哪种语言比较好”这个问题。简单地说，没有所谓“最好”的语言，因为这取决于环境。例如，用 BASIC 编写程序通常比使用其他语言快得多，所以如果开发速度比较重要，但执行性能不是很重要，那么 BASIC 是一种非常好的选择。另一方面，如果程序要求具有 C++提供的执行性，或者应用程序需要 C++中的功能而不是 BASIC，那显然应该使用 C++。如果应用程序必须在许多不同的计算机系统上执行，而且执行性能不是很重要，可能 Java 就是最好的选择。