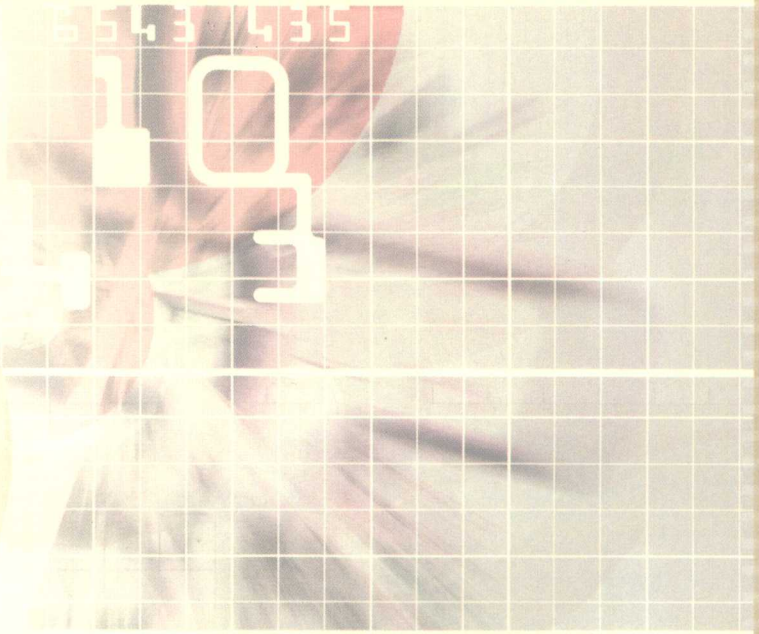
21世纪大学计算机基础规划教材

Visual C++程序设计基础

（第二版）

柴 欣 张红梅 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

21 世纪大学计算机基础规划教材

Visual C++程序设计基础（第二版）

柴　欣　张红梅　主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

Visual C++功能强大，使用灵活，但对于初学程序设计的人来说还是较难深入的。本书采用循序渐进、逐步深入的方法，对VC的基础、面向对象的程序设计方法及Windows编程进行讲解，力求使读者易学易懂。

本书共9章，其中1～6章是C++的语言基础；第7章是面向对象基础；第8、9章是Visual C++的Windows编程。这三部分内容相互衔接、前后呼应，讲解由浅入深、循序渐进。为了提高读者的编程技巧，大部分章节结合典型实例对基本概念和方法进行讲解，并在每章后设计了许多习题，便于读者练习。同时本书还配有《Visual C++程序设计实验教程（第二版）》一书，可对学生上机实验提供指导与帮助。

本书语言表达严谨、流畅，通俗易懂，内容重点突出，实例丰富，适合作为高等院校各专业程序设计语言课程的正式教材，又可作为研究生计算机基础教育的教材，也比较适合作为广大计算机爱好者的自学和参考用书。

图书在版编目（CIP）数据

Visual C++程序设计基础/柴欣，张红梅主编. —北京：中国铁道出版社，2007.8
（21世纪大学计算机基础规划教材）
ISBN 978-7-113-08274-1

Ⅰ.V... Ⅱ.①柴...②张... Ⅲ.C语言－程序设计－高等学校－教材 Ⅳ.TP312

中国版本图书馆CIP数据核字（2007）第131189号

书　　名：Visual C++程序设计基础（第二版）
作　　者：柴　欣　张红梅
出版发行：中国铁道出版社（100054，北京市宣武区右安门西街8号）
策划编辑：严晓舟　秦绪好
责任编辑：李　旸　黄园园
封面设计：付　巍
封面制作：白　雪
印　　刷：三河市华晨印务有限公司
开　　本：787×1092　1/16　**印张**：20.5　**字数**：481千
版　　本：2007年8月第1版　　2007年8月第1次印刷
印　　数：1～5 000册
书　　号：ISBN 978-7-113-08274-1/TP・2559
定　　价：27.00元

编 委 会

前　言

随着计算机技术的飞速发展，社会对人才的计算机应用与开发水平的要求也日益提高，为适应这一形势，高校的计算机基础教学内容也在不断的改革之中。

目前，C 语言已经成为高校理工科学生的必修或选修课，但随着软件工程技术的不断发展，面向对象的编程技术已经成为当今软件开发的重要手段之一，尤其是 Visual C++的出现，大大推进了面向对象与可视化编程技术的应用与发展。因此，掌握面向对象与可视化程序设计的内容和方法已经成为大学生计算机应用与开发能力的要求之一。Visual C++功能强大，使用灵活，但是对于初学程序设计的人来说，接受起来还是有一定的难度。虽然目前市面上关于 Visual C++的书籍很多，但适合以上读者群体的高教教材并不多见，鉴于这种情况，我们对《Visual C++程序设计基础》一书进行了修订，出版第二版。本书保持了第一版的写作风格，保留了通俗易懂的特点，同时扩充了典型的实例，力求使初学程序设计的读者能够掌握 Visual C++的程序设计方法，并初步具备使用 Visual C++开发应用程序和解决实际问题的能力。

全书共分为 9 章，第 1～6 章为 C++的基础部分。在该部分中，较为系统地讲述了 C++语言的基础知识、基本规则及编程方法，其中第 1 章介绍了程序设计语言的发展，并通过一个简单实例，对 Visual C++ 6.0 集成开发环境进行了介绍；第 2～6 章讲述了 C++语言的基本内容，包括：常量、变量、运算符和表达式、各种语句、函数和预处理、指针和引用、结构和联合体等，这些内容也是构成 C++程序的基础。第 7 章重点介绍 C++的面向对象的基本思想及面向对象的设计方法。在这一章中，较系统地讲述了 C++语言中面向对象的主要特征，如封装、继承、多态等，这些都是 C++的核心内容，体现了 C++语言面向对象的特点。第 8、9 章重点介绍了可视化编程的基本方法。在这两章中主要介绍 Windows 编程基础知识、使用 MFC 进行可视化编程的基本方法。

本书的编者长期从事 C++语言程序设计课的教学工作，并曾利用 C++、Visual C++语言开发了多个软件项目，因此有着丰富的教学经验和较强的科研能力，对 Visual C++有着较深入的理解。在全书的编写过程中，编者遵循由浅入深、循序渐进的方针，本着加强基础、注重实践、勇于创新、突出应用的原则，力求使本教材具有可读性、适用性和先进性。为了便于读者自学，提高读者编程技巧，书中大部分章节都提供了典型例题，不仅适用于教学，同时也适用于用 Visual C++开发应用程序的用户参考。如教师选用本书作为大学生软件技术基础课程的教材时，可根据实际授课时数取舍本书章节。由于授课时数的限制，教师可在规定授课时数内重点讲授基础部分的内容，而在后续的选修课程或研究生课程中介绍后三章的内容，以使学生能够完整地学习 Visual C++的内容。

本书中所给出的程序示例均在 Visual C++ 6.0 下进行过调试和运行。为帮助读者更好地学习 Visual C++，作者还编写了配套的《Visual C++程序设计实验教程（第二版）》一书，该书充实了一些编程实例，可供学生上机实验时使用。

本书由柴欣、张红梅主编，并负责全书的总体策划与统稿、定稿工作，武优西、刘洪普任副主编，各章编写分工如下：第 1 章由柴欣编写，第 2 章由梁艳红编写，第 3 章由李艳编写，第 4 章由刘洪普编写，第 5 章由史巧硕编写，第 6 章由路静编写，第 7 章由武优西编写，第 8、9 章由张红梅编写。李惠然、付灵丽、朱怀忠、毕晓博、曹新国、刘靖宇、王建勋、杨素梅等老师参加了本书大纲的讨论及部分编写工作，并对书中的程序进行了调试。

在本书的编写过程中一直得到有关专家热心的指导与无私的帮助，编者在此一并表示衷心的感谢。此外在本书写作时，还参考了大量文献资料，在此也向这些文献资料的作者深表感谢。

由于时间仓促和水平所限，书中难免有不当和欠妥之处，敬请各位专家、读者不吝批评指正。

编　者

2007 年 7 月

目 录

第1章 绪 论

为了让计算机完成不同的工作，就要编制不同的程序（Program）。本章首先对程序设计进行了概述，介绍了程序设计语言、程序设计方法和C/C++语言的有关内容，然后介绍了简单的C++程序的结构以及如何在Microsoft Visual C++ 6.0中完成一个C++程序的编制。

1.1 概 述

计算机科学发展的每一步几乎都在软件设计和程序设计语言中得到了体现。随着软件开发规模的不断扩大和开发方式的变化，发展了许多程序设计方法，程序设计开始被人们作为一门学科来对待。

1.1.1 程序设计语言

随着计算机的迅速发展和广泛应用，计算机能完成越来越纷繁复杂的工作。而计算机要完成不同的工作，就要运行不同的程序。程序就是为完成某项任务而由若干条计算机指令组成的有序集合，编制程序称为程序设计。人们通过编写程序，发挥计算机的作用，解决工作中的各种问题。

人与人之间交流需用相互理解的语言沟通，人与计算机交流也要使用相互理解的语言。程序设计语言就是用来实现人与计算机之间交流的，通常分为机器语言、汇编语言和高级语言三类。

1．机器语言（Machine Language）

计算机产生的初期，都使用机器语言编写程序。每种型号的计算机的指令系统就是该种计算机的机器语言。因为机器语言用一串二进制代码表示一条指令，所以用机器语言编写的程序计算机可直接识别和执行，称为目标程序。机器语言是计算机唯一能够识别并直接执行的语言，所以与其他程序设计语言相比，执行速度最快、效率最高。

2．汇编语言（Assemble Language）

由于机器语言中每条语句都是一串二进制代码，可读性差、不易记忆，编写程序非常困难而且容易出错，程序的调试和修改难度也很大，使得机器语言不易掌握和使用。

由于机器语言的缺点，人们进行改进以方便程序的编写和维护。20世纪50年代初，出现了汇编语言。汇编语言用比较容易识别、记忆的助记符号代替相应的二进制代码，也叫符号语言。

用汇编语言编写的程序称为汇编语言源程序，计算机不能直接识别和执行它，必须先把汇编语言源程序翻译成目标程序，然后才能被计算机执行。

3．高级语言

虽然使用汇编语言比机器语言方便了许多，但只是表示方法上的改进，和机器语言的性质是一样的，仍然依赖于计算机的型号，通用性差，而且与人类自然语言及数学公式的表达形式相差甚远。这时的程序设计相当麻烦，编制和调试一个稍大一点的程序常常要花费很长的时间，培养一个熟练的程序员更需经过长期的训练和实习，这种局面严重影响了计算机的普及应用。

到了20世纪50年代中期，人们研制了高级语言。高级语言是用接近自然语言表达各种意义的“词”和常用的“数学公式”形式，按照一定的“语法规则”编写程序的语言。这里的“高级”，是指这种语言与自然语言和数学公式相当接近，而且不依赖于计算机的型号，通用性好。高级语言的使用，改善了程序的可读性、可维护性和可移植性，大大提高了编写程序的效率。

用高级语言编写的程序称为高级语言源程序，不能被计算机直接识别和执行，也要用翻译的方法把高级语言源程序翻译成目标程序才能执行。

高级语言的出现大大简化了程序设计，缩短了解题周期，因此显示出强大的生命力。此后，编制程序已不再是软件专业人员才能做的事，一般工程技术人员花上较短的学习时间，也可以使用计算机解题。这个时期，随着计算机应用日益广泛地渗透到各学科和技术领域，发展了一系列不同风格的、为不同对象服务的程序设计语言。其中较为著名的有FORTRAN、BASIC、COBOL、ALGOL、LISP、PL/1、Pascal、C等十几种语言。

1.1.2 程序设计方法

程序设计是一门技术，需要相应的理论、技术、方法和工具来支持。就程序设计方法和技术的发展而言，主要经历了结构化程序设计和面向对象的程序设计阶段。

1．结构化程序设计方法

随着高级语言的出现和快速发展，人们编写的程序越来越复杂、功能越来越强、规模越来越大。20世纪60年代末到70年代初，出现了大型的软件系统，如操作系统、数据库管理系统等，这给程序设计带来了新的问题。例如一个大型操作系统有时需要几千人/年的工作量，而所获得的系统又常常会隐藏着几百甚至几千个错误。大型软件系统的研制花费了大量的人力和物力，但编写出来的软件可靠性差、错误多、难以维护，已经到了程序员无法控制的地步，这就是“软件危机”。

“软件危机”震动了软件界，程序设计的传统习惯和工作方式导致了不清晰的程序结构，使得程序的可靠性难以保障；另一方面，程序设计工具的严重缺乏也使得大型系统的开发陷入困境。此时人们开始重新审视程序设计中的一些最基本的问题，例如，程序的基本组成部分是什么？应该用什么样的方法来设计程序？如何保证程序设计的正确性？程序设计的主要方法和技术应如何规范等。

1969年，E.W.Dijkstra首先提出了结构化程序设计的概念，强调了从程序结构和风格上研究程序设计，为克服“软件危机”起了很大的缓解作用。此后，结构化设计方法得到了很大的发展，Niklans Wirth又提出了“算法＋数据结构＝程序设计”的程序设计方法，将软件

划分成若干个可单独命名和编址的部分，它们被称为模块，模块化使软件能够被有效地管理和维护，从而能够有效地分解和处理复杂问题。

结构化程序设计主要采用自顶而下、逐步细化的模块化设计方法，即先全局后局部、先整体后细节、先抽象后具体的设计方法。由于实际问题往往比较复杂，为了提高效率，在进行结构化程序设计时，常常伴随着使用关键部分优先考虑的设计方法。

将一个完整的问题分解成若干相对独立的问题，只要这些问题能分别得到正确的解决，整个问题也就解决了。子问题又可进一步分解为若干子问题，这样可以一直重复下去，直到每个问题都已简单到我们满意的程度。对每步分解，都要做出分解方法的决策，不同的决策会导致不同的解法。把这种程序设计方法称之为逐步细化法，也就是在编写程序时一步一步地不断细化的过程。

1966年，Boehm和Jacopini证明了任何程序使用顺序、选择和循环结构就足以表达出来，这就是构成结构化程序的三种基本结构。使用这三种基本结构来构造程序，可使程序的结构清晰、易读易懂且质量好、效益高。下面简单介绍一下组成结构化程序的顺序、选择和循环这三种基本结构。

（1）顺序结构

顺序结构是一种最简单、最基本的结构，由一组顺序执行的程序块组成。在顺序结构内，各块按照它们出现的先后顺序依次执行。图1-1表示了一个顺序结构形式，从图中可以看出它有一个入口a，一个出口b，在结构内A框和B框都是顺序执行的处理框。

（2）选择结构

选择结构又叫分支结构，也是一种基本和常用的结构，它向我们提供了根据条件的值来选择不同处理块的方法。选择结构中包含一个判断框，根据给定的条件p是否成立而选择执行A框或B框，当条件成立时，执行A，否则执行B。A框或B框可以是空框，即不执行任何操作，执行完A或B后从出口b退出，然后接着执行其后的过程。图1-2所示的虚线部分就是选择结构，在选择结构中程序产生了分支，但对于整个的虚线框而言，它仍然只具有一个入口a和一个出口b。

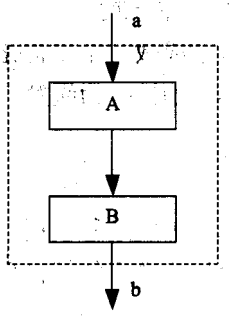


图1-1 顺序结构流程图

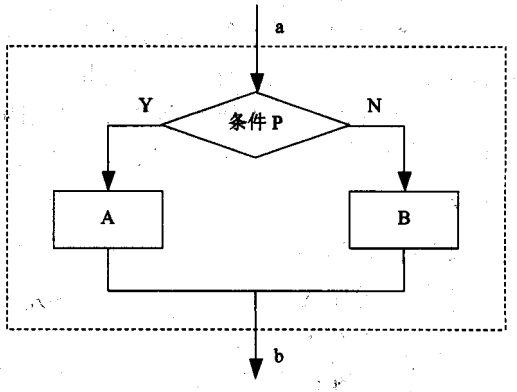


图1-2 选择结构流程图

（3）循环结构

循环结构又称重复结构，是指在一定条件下反复执行一个程序块的结构。循环结构也是

只有一个入口，一个出口。根据循环控制方式的不同，循环结构分为当型循环结构和直到型循环结构两种。

① 当型循环的结构如图 1-3 所示，其功能是：当给定的条件 p 成立时，执行 A 框操作，执行完 A 操作后，再判断 p 条件是否成立，如果成立，再次执行 A 操作，如此重复执行 A 操作，直到判断 p 条件不成立时才停止循环。此时不执行 A 操作，而从出口 b 脱离循环结构。

② 直到型循环的结构如图 1-4 所示，其功能是：先执行 A 框操作，然后判断给定条件 p 是否成立，如果不成立，再次执行 A 操作；然后再对 p 进行判断，如此反复，直到给定的 p 条件成立为止。此时不再执行 A 框，从出口 b 脱离循环。

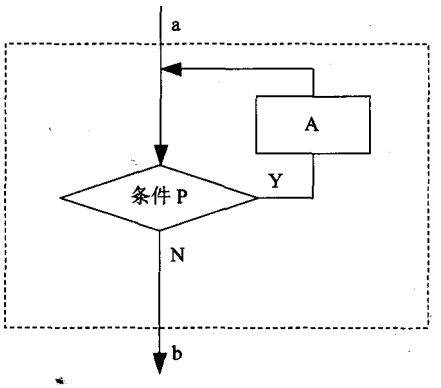


图 1-3　当型循环结构流程图

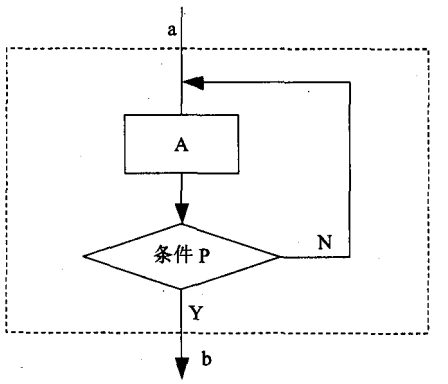


图 1-4　直到型循环结构流程图

由以上三种基本结构构成的程序，称为结构化程序。一个结构化程序，以及三种基本结构中的每一种结构都应具有以下特点：

- 只有一个入口。
- 只有一个出口。
- 没有死语句，即每一个语句都应该有一条从入口到出口的路径通过它。
- 没有死循环（无限制的循环）。

实践证明，任何满足以上四个条件的程序，都可以表示为由以上三种基本结构所构成的结构化程序；反之，任何一个结构化程序都可以分解为一个个基本结构。

经过多年的探索和实践，结构化程序设计的应用确实取得了成效，用结构化程序设计方法编写出来的程序不仅结构良好，易写易读，而且易于证明其正确性，为缓解当时的软件危机起到了很大作用。

2．面向对象程序设计方法

由于软件开发是对问题的求解过程，从认识论角度看，软件开发过程包括人们对要解决问题及相关事物的认识和基于认识所进行的描述。而结构化设计方法不能直接反映出人类认识问题的过程，它把数据和操作作为相互独立的部分，忽略了它们之间的内在联系。结构化设计方法以操作过程为核心，是面向过程的，所以又称为面向过程的程序设计方法。随着计算机软件的发展，软件系统越来越复杂庞大，结构化程序设计方法已显得力不从心。

在软件开发中各种概念和方法积累的基础上，就如何超越程序的复杂性障碍，如何在计

算机系统中自然的表示客观世界等问题，人们提出了面向对象的设计方法。它不是以过程为中心，而是以对象代表的问题为中心环节，提出了“对象＋对象＋……＝程序设计”理论，使系统的程序设计实现过程与人们对复杂系统的认识过程尽可能地一致。

面向对象的程序设计方法在20世纪60年代后期被首次提出，以60年代末挪威奥斯陆大学和挪威计算中心共同研制开发的SIMULA语言为标志。SIMULA体现了面向对象的基本要点，是面向对象的鼻祖。20世纪70年代出现的Ada语言也是一种基于对象的语言，是支持数据抽象类型的最重要的语言之一，但它不能全面地支持继承面向对象。后来出现的Smalltalk是最有影响的面向对象的语言之一，丰富了面向对象的概念。

到20世纪80年代中期以后，面向对象的程序设计语言日趋成熟，并被广泛地应用于程序设计。总体来说，面向对象的程序设计语言大致分两类：一类是纯面向对象的语言，如Smalltalk和Eiffel；另一类是混合型的面向对象语言，如C++和Objective C。面向对象的程序设计方法的出现使“软件危机”得到了很好的解决，也是目前广泛使用的程序设计方法。

1.1.3 C及C++语言的发展和特点

1960年出现的ALGOL 60是一种面向问题的高级语言，它离硬件比较远，不宜编写系统程序。1963年英国剑桥大学推出了CPL(Combined Programming Language)语言，它在ALGOL 60的基础上接近硬件一些，但规模比较大，使用困难。1967年英国剑桥大学的Martin Richards对CPL进行了简化，推出了BCPL（Basic Combined Programming Language）语言。

1970年，美国贝尔实验室的Ken Thompson以BCPL语言为基础，设计出了很简单又很接近硬件的B语言（BCPL的第一个字母），并用B语言对用汇编语言编写的UNIX操作系统进行了部分改写，此时的B语言过于简单，功能有限。1972～1973年间，贝尔实验室的D.M.Ritchie在B语言的基础上设计出了C语言(BCPL的第二个字母)。C语言既保持了BCPL和B语言精炼和接近硬件的优点，又克服了它们过于简单和无数据类型等缺点。1973年，贝尔实验室的Ken Thompson和D.M.Ritchie将UNIX操作系统的90%用C语言改写。

后来人们又对C语言进行了多次改进，但主要还是在贝尔实验室内部使用。直到1975年UNIX 6.0的发布，C语言的优点才引起人们的广泛注意。1978年以后，C语言先后移植到大、中、小、微型计算机上，很快风靡全世界，成为应用最广泛的计算机语言之一。

C语言是结构化、模块化的程序设计语言，是面向过程的，在处理小规模程序时较为得心应手。当问题非常复杂、程序规模很大时，面向过程的程序设计方法就显示出它的不足。而C语言的种种优点又令人们难以割舍，因此，为适应面向对象的程序设计方法，人们在C语言基础上加以发展。20世纪80年代AT&T公司的贝尔实验室的Bjarne Stroustrup博士在C语言基础上开发出支持面向对象的C语言，被他称为“带类的C”。后来为了强调它是C语言的增强版，1983年Rick Masenirti提出改称为C++。

C++是一门高效实用的混合型程序设计语言，它最初的设计目标是：支持面向对象编程技术、支持抽象形态的类、更好的C语言。C++语言包括两部分：一是C++基础部分，它是以C语言为核心的；另一部分是C++面向对象部分，是C++对C语言的扩充部分。这样它既支持面向对象程序设计方法，又支持结构化程序设计方法，同时由于它广泛的应用基础和丰富的开发环境的支持，也使面向对象设计得到很快普及。