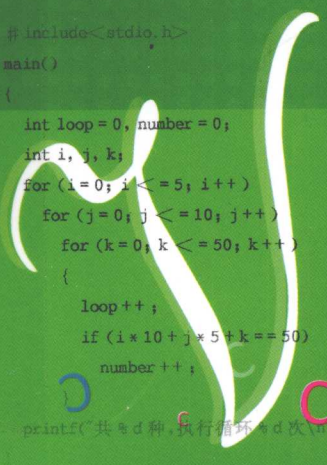
普通高等院校规划教材

面向对象程序设计与VC程序设计入门

主　编　余祖龙　孙开琼
副主编　江少锋　王　玉　李军华



北京航空航天大学出版社

普通高等院校规划教材

面向对象程序设计与VC程序设计入门

主　编　余祖龙　孙开琼
副主编　江少锋　王　玉　李军华

北京航空航天大学出版社

内 容 简 介

系统地介绍面向对象程序设计的基本理论和 Visual C++程序设计的基本方法。全书分C++和 VC 两大部分，共 15 章。内容包括：C++概述，C++语言基础，C++基本控制结构，函数，类与对象，数组与指针，继承与派生，多态性，Visual C++集成开发环境，基于文档视图的程序设计，菜单、工具栏、状态栏和快捷键，基于对话框的程序设计，定时器及其应用，Windows 标准控件，设备上下文与图形设备接口。

本书可作为高等学校相关课程的教材或参考书，也可作为 C++和 Visual C++的学习者自学或培训教材。

图书在版编目(CIP)数据

面向对象程序设计与 VC 程序设计入门/余祖龙，孙开琼主编. --北京：北京航空航天大学出版社，2010.2

ISBN 978-7-5124-0014-6

Ⅰ. ①面… Ⅱ. ①余… ②孙… Ⅲ. ①C 语言—程序设计 Ⅳ. TP312

中国版本图书馆 CIP 数据核字(2010)第 015893 号

面向对象程序设计与 VC 程序设计入门

主　编　余祖龙　孙开琼

副主编　江少锋　王　玉　李军华

责任编辑　王　实

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100191)　发行部电话：(010)82317024　传真：(010)82328026

http://www.buaapress.com.cn　E-mail:bhpress@263.net

北京时代华都印刷有限公司印装　各地书店经销

*

开本：787 mm×1 092 mm　1/16　印张：18　字数：461 千字

2010 年 2 月第 1 版　2010 年 2 月第 1 次印刷　印数：4 000 册

ISBN 978-7-5124-0014-6　　定价：27.00 元

前　言

从20世纪60年代提出面向对象概念至今，面向对象技术已发展成为一种比较成熟的编程思想，并且逐步成为目前软件开发领域的主流技术。这种技术从根本上改变了人们以往设计软件的思维方式，它集抽象、封装、继承和多态于一体，实现了代码重用和代码扩充，极大地减少了软件开发的复杂性，提高了软件开发的效率。

目前，很多理工科院校都开设了“面向对象程序设计”课程，主要讲解C++的基本理论知识，而对VC部分通常不作介绍。学生在学习了C++的理论知识后，由于没有合适的开发平台让他们把所学的理论知识应用到实际的软件设计中去，从而很难具备工程软件设计能力。但是，当前的用人单位对学生工程软件的设计能力有较高的要求，通常需要他们具备用VC++开发工程软件的能力。因此，教学环节与人才的培养及用人单位的实际需求严重脱钩。我们针对存在的问题，根据实际需求，编写了本书，把面向对象程序设计的基本理论知识和Visual C++程序设计的基本方法有机地结合起来，以符合人才培养的方向及社会发展的需求。

本书的编写宗旨与特色表现在以下几个方面：

1. 重点突出，理论联系实践。在对比十几本C++和Visual C++的教材内容的基础上，我们在编写本书时选取了C++精髓部分，做到重点突出，并且结合VC，使C++的理论知识能够迅速应用到VC程序的开发中，并将所学的理论知识用于VC程序设计中，做到理论紧密联系实践。

2. 实例程序的趣味性。选取了大量耳熟能详的小游戏设计程序作为实例，如石头剪刀布、猜数字等，使读者在学习程序设计的过程中充分体会到编程带来的乐趣，寓学于乐，提高学习的效率和效果。

3. 内容选取上的创新性。根据程序设计的实际需要，在第2章加入了随机数知识的讲解，并在第13章介绍了定时器及其应用。

4. 代码的准确性。书中所有的例题源代码都在Visual C++ 6.0上调试通过，以确保程序代码准确无误。

本书分为两大部分：一是C++部分，内容包括C++概述，C++语言基础，C++基本控制结构，函数，类与对象，数组与指针，继承与派生，多态性；二是VC部分，内容包括Visual C++集成开发环境，基于文档/视图的程序设计，菜单、工具栏、状态栏和快捷键，基于对话框的程序设计，定时器，Windows标准控件，设备上下文与图形设备接口。

为了与书中程序对应及保证全书体例上的统一，本书中的符号全部采用正体

书写。

本书由余祖龙、江少锋、王玉、孙开琼和李军华共同编写。余祖龙负责编写第1,5,8～15章,江少锋博士负责编写第2,3章,王玉讲师负责编写第7章,孙开琼博士负责编写第6章,李军华博士负责编写第4章。全书由余祖龙统一修改、整理和定稿。

感谢我的学生马继鹏、周慧、苏成臣、宋红林、王超、陈波、劳万利和蒋空空,他们为本书的编写做了大量的文字输入与校对工作。

感谢为本书出版付出辛勤劳动的北京航空航天大学出版社的工作人员。

由于编者水平有限,时间仓促,书中的错误与疏漏之处,恳请读者批评指正。在使用本书时如遇到什么问题需要与作者交流,或想索取本书例题的源代码与电子讲稿,请与作者联系。联系方式:yuzulong1979@126.com。

编 者

2009年12月

注:本书配有教学课件,授课教师可通过 bhkejian@126.com,或者致电 010-82317027 索取,非常感谢您对北航出版社图书的关注与支持。

目　录

第1章　C++概述

1.1　C++的产生

众所周知，C语言是面向过程的结构化程序设计语言。在进行较小规模的程序设计时，设计者用C语言较为得心应手。但是，当问题比较复杂、程序的规模比较大时，特别是进行大型软件设计时，结构化程序设计方法就显现出它的不足，具体表现在数据的封装差、代码重用性差等。

为了解决软件设计存在的问题，美国AT&T(贝尔实验室)的Bjarne Stroustrup博士在20世纪80年代初期发明并实现了C++(最初这种语言被称为C with Classes)。一开始，C++是作为C语言的增强版出现的，从给C语言增加类开始，不断地增加新特性。虚函数(virtual function)、运算符重载(operator overloading)、多重继承(multiple inheritance)、模板(template)、异常(exception)、命名空间(namespace)逐渐被加入标准。1998年，国际标准组织(ISO)颁布了C++程序设计语言的国际标准ISO/IEC 14882—1998。C++是具有国际标准的编程语言，通常称为ANSI/ISO C++。1998年是C++标准委员会成立的第一年，以后每5年视实际需要更新一次标准。

C++是由C语言发展而来的，与C语言兼容。用C语言写的程序基本上可以不加修改地用于C++。从C++的名字可以看出它是C的超集。C++既可用于面向过程的结构化程序设计，也可用于面向对象的程序设计，是一种功能强大的混合型程序设计语言。

目前，C++越来越受到重视并已得到了广泛采用，许多软件公司为C++设计编译系统，提供不同应用级别的类库和越来越方便的开发环境，如Microsoft公司的Visual C++ 6.0及以上版本、Borland公司的Borland C++ 5.02，以及自由软件GCC等。

1.2　计算机程序语言的发展

1.2.1　程序和程序语言

程序是计算机处理对象和计算规则的描述。程序设计语言是用来描述计算机事物处理过程、便于计算机执行的规范化语言。语言的基础是一组记号和规则，根据规则由记号构成记号串的总体就是语言。

人类自然语言是人们进行交流和表达思想的工具。那么，人与计算机如何进行“交流”呢？为此，就产生了计算机语言，其功能是人用计算机语言编写一系列动作，计算机能够“理解”这些动作，并按照指定的动作去执行。正是这种相同点，所以计算机语言和自然语言都叫做“语言”。

自然由于其历史性和文化性，除了其语法外，还包含复杂的语义和语境，所以，人们也能理解很多不完全符合语法的语句。但计算机语言是由人发明的，它主要是用语法来表达人的思

想，因而在编写程序时要严格遵守语法规则。

如同人类有很多种语言一样，计算机语言也有很多种。按照计算机历史的发展有如下几类：

1. 机器语言

它是面向机器的，是特定计算机系统所固有的语言。用机器语言进行程序设计，需要对机器结构有较多的了解。用机器语言编写的程序可读性差，程序难以修改和维护。

2. 汇编语言

为了提高程序设计效率，人们考虑用有助记忆的符号来表示机器指令中操作码和运算数，例如用 ADD 表示加法，SUB 表示减法等。相对于机器语言而言，用汇编语言编写程序的难度有所降低，程序的可读性有所提高，但仍与人类的思维相差甚远。

3. 高级语言

汇编语言和计算机语言十分接近，它的书写格式在很大程度上取决于特定计算机的机器指令，这对于人们抽象思维和交流十分不便。高级语言指的是像 FORTRAN、C、Pascal 和 Basic 等与具体机器无关的语言。有了高级语言，程序设计者不需要了解机器的内部结构，只要按照计算机语言的语法编写程序即可。

1.2.2 结构化程序设计

出现高级语言之后，如何用它来编写较大的程序呢？人们把程序看成是处理数据的一系列过程。过程或函数定义为一个接一个顺序执行的一组指令。数据与程序分开存储，程序设计的主要技巧在于追踪哪些函数和调用哪些函数，以及哪些数据发生了变化。为此，结构化程序设计应运而生。

结构化程序设计的主要思想是功能分解并逐步求精。也就是说，当要设计某个目标系统时，先从代表目标系统整体功能的单个处理着手，自顶向下不断地把复杂的处理分解为子处理，这样一层一层地分解下去，直到仅剩下若干个容易处理的子项目为止。当所分解出的子项目已经十分简单，其功能显而易见时，就停止这种分解过程，对每个这样的子项目程序加以实现。

结构化程序设计仍然存在诸多问题：生产率低下，软件代码重用程度低，软件维护复杂，等等。针对结构化程序设计的缺点，提出了面向对象的程序设计方法。

1.2.3 面向对象的程序设计

面向对象程序设计的本质是把数据和处理数据的过程当成一个整体，即对象。

一般认为，面向对象语言至少包含下面一些概念。

1. 类

把众多的事物归纳、划分成一些类，把具有共性的事物划分为一类，得出一个抽象的概念，是人类认识世界经常采用的思维方法。类是面向对象语言必须提供的用户定义的数据类型，它将具有相同状态、操作和访问机制的多个对象抽象成为一个对象类。在定义了类以后，属于这种类的一个对象叫做类的实例或对象。一个类的定义应包括类名、类的说明和类的实现。

2. 对 象

对象是人们要进行研究的任何实际存在的事物，它具有状态（用数据来描述）和操作（用来改变对象的状态）。面向对象语言把状态和操作封装于对象体之中，并提供一种访问机制，使

对象的“私有数据”仅能由这个对象的操作来执行。用户只能通过向允许公开的操作提出要求，才能查询和修改对象的状态。这样，对象状态的具体表示和实现都是隐蔽的。

3. 继　承

继承是面向对象语言的另一个必备要素。类与类之间可以组成继承层次，一个类的定义（称为子类）可以定义在另一个定义类（称为父类）的基础上。子类可以继承父类中的属性和操作，也可以定义自己的属性和操作，从而使在内部表示上有差异的对象可以共享与它们结构有共同部分的有关操作，达到代码重用的目的。

4. 多态性

多态性是指同样的消息被不同类型的对象接收时导致完全不同的行为。多态具有可替换性、可扩充性、灵活性以及简化性等特点，这也是面向对象程序设计和结构化程序设计的一个主要区别之一。

面向对象程序设计的主要优点如下：

（1）与人类习惯的思维方式一致

结构化程序设计是面向过程的，以算法为核心，把数据和过程作为相互独立的部分。面向对象程序设计以对象为中心。对象是一个统一体，是由描述内部状态表示静态属性的数据以及可以对这些数据施加的操作封装在一起所构成的。面向对象设计方法是对问题域的模拟，模拟客观世界。

（2）代码重用性好

面向对象的软件技术在利用可重用的软件成分构造新的软件系统时有很大的灵活性。有两种方法可以重复使用一个对象类：一种方法是创建该类的实例，从而直接使用它；另一种方法是从它派生出一个满足当前需要的新类。继承性机制使得子类不仅可以重用其父类的数据结构和程序代码，而且可以在父类代码的基础上方便地修改和扩充，这种修改并不影响对原有类的使用。人们可以像使用集成电路（IC）构造计算机硬件那样，比较方便地重用对象类来构造软件系统。

（3）可维护性强

类是理想的模块机制，它的独立性好，修改一个类通常很少会牵扯到其他类。如果仅修改一个类的内部实现部分（私有数据成员或成员函数的算法），而不修改该类的对外接口，则可以完全不影响软件的其他部分。面向对象软件技术特有的继承机制，使得对软件的修改和扩充比较容易实现，通常只要从已有类派生出一些新类，无须修改软件原有成分。面向对象软件技术的多态性机制，使得扩充软件时对原有代码所需作的修改进一步减少，需要增加的新代码也比较少。所以，面向对象方法设计的程序具有很好的可维护性。

正因为面向对象程序设计有众多的优点，所以，今天程序设计方法逐步由结构化程序设计发展为面向对象程序设计。

1.3　C++语言的特点

C++语言的主要特点表现在以下两个方面：

- 与 C 语言有较好的兼容性；
- 支持面向对象的方法。

首先，C++的确是一个更好的 C 语言，它保持了 C 的简洁、高效和接近汇编语言等特点，对 C 的类型系统进行了改革和扩充，因此 C++比 C 更安全，C++的编译系统能检查出更多的类型错误。

由于 C++与 C 保持兼容，从而使许多 C 代码不经修改即可为 C++所用，用 C 编写的众多的库函数和实用软件也可以用于 C++中。另外，由于 C 语言已被广泛使用，因而极大地促进了 C++的普及和面向对象技术的广泛应用。也正是由于对 C 的兼容而使 C++既支持面向过程的程序设计，又支持面向对象的程序设计。虽然与 C 的兼容使得 C++具有双重特点，但它在概念上仍是和 C 语言完全不同的语言。因此，在程序设计的时候，应该注意按照面向对象的思维方式去编写程序。

如果读者已经有其他面向过程高级语言的编程经验，那么学习 C++语言时应该着重学习它的面向对象的特征，对于与 C 语言兼容的部分只要了解一下就可以了。C 语言与其他面向过程的高级语言在程序设计方法上是类似的。

1.4 简单的C++程序

为了使读者对 C++程序有一个基本的认识，下面首先介绍一个简单的 C++程序。

例 1.1 在屏幕上输出字符“hello world!”。

程序如下：

```
#include <iostream>
using namespace std;  //使用标准命名空间
int main( )
{
cout << "hello world!" << endl;
return 0;
}
```

程序运行结果：

```
hello world!
```

程序说明：

① 在 C++程序中，一般在主函数 main 前面加一个类型声明符 int，表示 main 函数的返回值为整型(标准 C++规定 main 函数必须声明为 int 型，即此主函数带回一个整型的函数值)。“return 0;”的作用是向操作系统返回 0，如果程序不能正常执行，则会自动向操作系统返回一个非零值，一般为－1。

② 在 C++程序中，可以使用 C 语言中的“/* …… */”形式的注释行，还可以使用以“//”开头的注释。从例 1.1 可以看到：以“//”开头的注释可以不单独占一行，它可以出现在一行中的语句之后。编译系统将“//”以后到本行末尾的所有字符都视为注释。应注意：它是单行注释，不能跨行。这种注释方法比较灵活方便，程序设计人员大多愿意用这种注释方法。

③ 在 C++程序中，一般用 cout 进行输出。cout 是 C++系统定义的对象名，称为输出流对象。对象和输出流对象的概念将在后面介绍。为了便于理解，把用 cout 和“<<”实现输

出的语句简称为 cout 语句。“<<”是“插入运算符”，与 cout 配合使用，在本例中，它的作用是将运算符“<<”右侧双撇号内的字符串“hello world!”插入到输出流，C++系统将输出流 cout 的内容输出到系统指定的设备(一般为显示器)中。除了可以用 cout 进行输出外，在 C++程序中还可以用 printf 函数进行输出。

④ 使用 cout 需要用到头文件 iostream。程序的第一行 #include <iostream>是一个预处理命令。文件 iostream 的内容是提供输入或输出时所需要的一些信息，从它的形式就可以知道它代表输入或输出流的意思。由于这类文件都放在程序单元的开头，所以称为头文件(head file)。

注 意 在C语言中所有的头文件都带后缀.h(如 stdio.h)，而按C++标准要求，由系统提供的头文件不带后缀.h，用户自己编制的头文件可以有后缀.h。在C++程序中也可以使用C语言编译系统提供的带后缀.h的头文件，如“#include <math.h>”。

⑤ 程序中“using namespace std;”的意思是使用命名空间 std。C++标准库中的类和函数是在命名空间 std 中声明的，因此程序中如果需要使用 C++标准库中的有关内容(此时需要用 #include 命令行)，就需要用“using namespace std;”语句作声明表示要用到命名空间 std 中的内容。命名空间的概念暂可不必深究，只需知道：如果程序有输入或输出时，必须使用 #include <iostream>以提供必要的信息，同时要用“using namespace std;”语句使程序能够使用这些信息，否则程序编译时将出错。本书中几乎所有的 C++程序的开头都包含此两行。

1.5 C++程序开发

1.5.1 C++程序开发过程

程序的开发通常要经过编辑、编译、链接、运行调试这几个步骤。编辑是将源程序输入计算机中，生成后缀为.cpp 的磁盘文件。编译是将程序的源代码转换为机器语言代码。但是，编译后的程序还不能由计算机执行，还需要链接。链接是将多个目标文件以及库中的某些文件连在一起，生成一个后缀为.exe 的可执行文件。最后，还要对程序进行运行、调试。

在编译和链接时，都会对程序中的错误进行检查，并将查出的错误显示在屏幕上。编译阶段查出的错误是语法错误，链接时查出的错误称为链接错误。只有将错误修改正确才能最终运行正确的可执行程序。

1.5.2 C++程序开发环境

C++程序的开发编译环境有很多，如 Borland 公司开发的 Turbo C++ 3.0 等，不过最方便、应用最广泛的还是微软公司的 Visual C++集成开发环境，本书所有的程序都是在 Visual C++ 6.0(简称 VC 6.0)中设计并调试完成的。

以例 1.1 为例，下面介绍如何在 VC 6.0 环境下开发和运行 C++程序。

当启动 VC 6.0 以后，可以看到如图 1-1 所示 VC 6.0 集成开发环境的窗口。

选择菜单命令 File | New，出现如图 1-2 所示的 New 对话框，单击 Projects 标签，在该选项卡中选择 Win32 Console Application(Win32 控制台应用程序)，在 Location 列表框中选择存放的路径，可以单击其右侧的浏览按钮“…”来对默认的存放路径进行修改，在 Project name(项目名称)文本框中为项目输入一个名字 Mycpp，然后单击 OK 按钮。

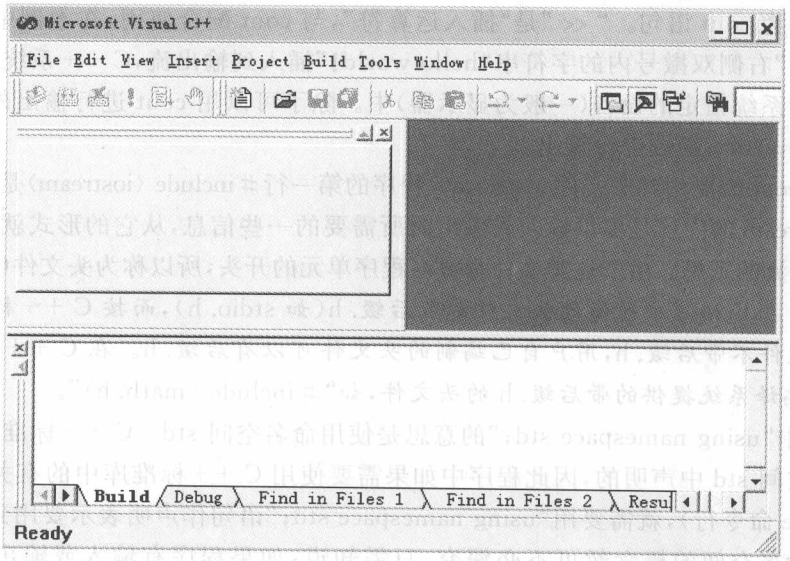


图 1-1 VC 6.0 开发环境的窗口

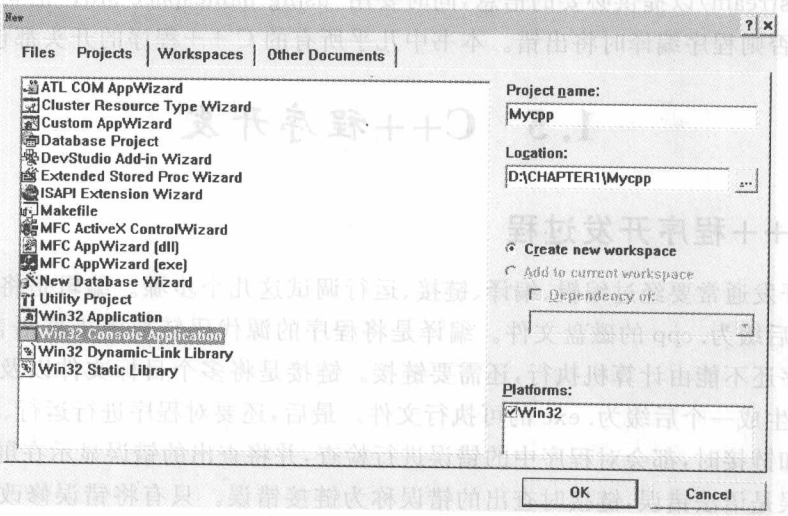


图 1-2 New 对话框

在弹出的 Win32 Console Application - Step 1 of 1 对话框中选择 An empty project 选项，如图 1-3 所示。然后单击 Finish 按钮，会弹出如图 1-4 所示的 New Project Information 对话框，单击 OK 按钮，完成项目的建立。

选择菜单命令 Project|Add to Project|New，在弹出的 New 对话框中的 Files 选项卡中选择 C++ Source File，并在 File 文本框中输入文件名称 cpp1，如图 1-5 所示。

单击图 1-5 中的 OK 按钮，会出现程序编辑区，如图 1-6 所示。光标在程序编辑区闪烁，提示用户在此处输入程序代码。

代码编辑完毕，按 F7 快捷键，或者选择菜单命令 Build|Build Mycpp.exe，或单击工具栏上的工具按钮，进行编译、链接，将会在输出区显示程序的编译和链接报告，如图 1-7 所示。

如果程序有错误，则屏幕下方的输出区就会显示错误信息，根据这些错误信息对源程序进行修改后，重新进行编译、链接，生成可执行程序。