王海军 著

程序设计

CHENGXUSHEJI

入门与提高

RUMENYUTIGAO

黄河出版社

程序设计

CHENGXUSHEJI

入门与提高

RUMENYUTIGAO

王海军 著

黄河出版社

责任编辑 张清训 葛春亮 **封面设计** 张宪峰

图书在版编目(CIP)数据

程序设计入门与提高/王海军著. —济南:黄河出版社,2008.8
ISBN 978-7-80152-986-2

Ⅰ.程… Ⅱ.王… Ⅲ.程序设计 Ⅳ.TP311.1

中国版本图书馆 CIP 数据核字(2008)年第 130686 号

书　名 程序设计入门与提高
作　者 王海军
出　版 黄河出版社
发　行 黄河出版社发行部
(济南市英雄山路 21 号　250002)
印　刷 济南霏帆印务有限公司
规　格 787 毫米×1092 毫米　16 开本
14.75 印张　350 千字
版　次 2008 年 8 月第 1 版
印　次 2008 年 8 月第 1 次印刷
印　数 1-1000 册
书　号 ISBN 978-7-80152-986-2/G·230
定　价 58.00 元

说　明

本书共12章，前2章介绍计算机程序与程序语言以及C++语言的概述、如何使用编译环境、C++中数据的描述与操作。第3、4章叙述三种基本程序结构、流程控制语句和函数，讨论了函数的定义和调用方法、函数参数的传递方式、系统函数、作用域与生存期、函数原型以及一些特殊的函数如内联函数、递归函数、重载函数，最后介绍了编译预处理中的一些问题。第5、6章讲了数组与自定义数据类型如结构体、共用体和枚举类型，接着讲解了指针、受限定的指针和指针数组、最后对引用做了介绍。第7章介绍了面向对象的程序设计发展历程以及相应的软件开发方法。接着第8、9章讲述类和对象、构造函数与析构函数、拷贝构造函数、this指针与静态成员以及名空间的使用、继承与派生的相关概念与定义以及在程序设计中如何使用继承与派生技术，书中给出了相应实例。第10、11章叙述了在联编中如何体现和应用多态性与虚函数、友元与运算符重载、程序设计中的模板，讲了函数模板和类模板以及相关应用。最后一章讲了I/O流的一些概念和应用以及在程序中如何对文件进行操作。本书可以作为大学本、专科计算机、电子等专业的教材，本书起点低，不要求学过其他程序设计语言，讲解通俗易懂、深入浅出，对例题重点阐述设计思想，培养读者的程序设计思维，因此本书既可作为面向对象程序设计的入门语言来学习，也可作为从事计算机应用工作的工程技术人员培训和自学的参考书和工具书。

王海军

目　　录

第一章　认识C++

1.1　计算机程序与程序设计语言

计算机程序或者软件程序(通常简称程序)是指一组指示计算机每一步动作的指令,通常用某种程序设计语言编写,运行于某种目标体系结构上。打个比方,一个程序就像一个用文字(程序设计语言)写下的工作步骤(程序),用于指导懂文字的人(体系结构)按照步骤(程序代码)来做这个工作。通常,计算机程序要经过编译和链接而成为一种人们不易理解而计算机易理解的格式,然后运行。未经编译就可运行的程序通常称之为脚本程序。

程序设计语言(Programming Language)是用于编写计算机程序的语言,就象人类语言用来表达思想一样。语言的基础是一组字符和一组规则。根据规则由字符构成的字符串的总体就是语言。在程序设计语言中,这些字符串就是程序。程序设计语言包含三个方面,即语法、语义和语用(这和学外语是很类似的)。语法表示程序的结构或形式,亦即表示构成程序的各个字符之间的组合规则,但不涉及这些字符的特定含义。语义表示程序的含义,表达的功能。语用表示程序与使用的关系。

程序设计语言的基本成分有:①数据成分,用于描述程序所涉及的数据;②运算成分,用以描述程序中所包含的运算;③控制成分,用以描述程序中所包含的控制;④传输成分,用以表达程序中数据的传输。

程序设计语言按照语言级别可以分为低级语言和高级语言。低级语言有机器语言和汇编语言。低级语言与特定的机器有关、功效高,用二进制(0和1)来表示,但使用复杂、繁琐、费时、易出差错。机器语言是表示成数码形式的机器基本指令集,或者是操作码经过符号化的基本指令集。如在计算机中要表示数5,6,7,则必须用四位二进制数表示:0101,0110,0111。其他的如文字、图片、操作等都最终要转换为二进制,计算机才能处理。

汇编语言是机器语言中地址部分符号化的结果,或进一步包括宏构造。汇编语言表示加法 AL=3+2 则可写成:MOV AL,3 ADD AL,2,表示先把3移动(MOV)到变量AL,然后加上(ADD)2,结果存放在变量AL里,这样的语言表达的内容看上去就更容易理解了。高级语言的表示方法要比低级语言更接近于待解问题的表示方法,更符合人类的思维习惯,其特点是在一定程度上与具体机器无关,易学、易用、易维护。例如上面的加法,用高级语言表示可写为:AL=3+2,即把3+2的结果放在变量AL里,这就和数学的表达式是一样的,所以就更容易理解、写程序也就更方便了。

程序设计语言按照用户的要求有过程式语言和非过程式语言之分。过程式语言也称结构化语言,主要依据1972年出现的结构定理:任何程序逻辑都可以用顺序、选择和循环三种基本结构来表示。它的主要特征是,用户可以指明一列可顺序执行的运算,以表示相

应的计算过程，如 C、FORTRAN、COBOL、PASCAL 等。非过程式语言指面向对象程序设计语言，本书从第 7 章将逐步讨论，C + + 就是这样的语言。

按照应用范围，有通用语言与专用语言之分。如 FORTRAN、COLBAL、PASCAL、C 等都是通用语言。目标单一的语言称为专用语言，如 APT 等。

按照使用方式，有交互式语言和非交互式语言之分。具有反映人机交互作用的语言成分的语言成为交互式语言，如 BASIC 等。不反映人机交互作用的语言称为非交互式语言，如 FORTRAN、COBOL、ALGOL69、PASCAL、C 等都是非交互式语言。

按照成分性质，有顺序语言、并发语言和分布语言之分。只含顺序成分的语言称为顺序语言，如 FORTRAN、C 等。含有并发成分的语言称为并发语言，如 PASCAL、Modula 和 Ada 等。

程序设计语言是软件的重要方面，其发展趋势是模块化、简明化、形式化、并行化和可视化。

1.2 面向对象的程序设计

面向对象就是思考问题的方法，以对象为主体，眼睛看到的都是对象。对象这个概念是直接从英文翻译过来的，对中国人也许说，面向东西，面向物体（抽象的物体），面向事物，之类的翻译或许更好理解。面向对象是和面向过程相对的。以前“面向对象”的程序概念还没有出来前，程序的设计都是在想怎样才能一步一步的解决问题，所以思维方式是过程和步骤。早期的 C 语言、BASIC 都是属于这一类。

而“面向对象”的基本思想是如何设计一个“能动”的物体（Object 对象）。他们有不同的属性和功能，之后的问题就是如何调用这些具有一定独立功能的东西，相互组合调用各个对象的功能（Method 方法），最后完成一个大的目的，即软件的目的。当然对于一个具体的功能实现，还是要使用步骤、过程、解决问题的先后顺序等这些面向过程的思想的。

面向对象的思考方法和概念提高了软件的开发效率。因为面向对象的设计中，都是在设计一个个的物体（Object），共同的地方，可以重复利用。在面向对象程序设计 C + + 中还有 C 语言等没有的很多新的概念，如继承，接口等，利用这些概念也可以提高开发的效率，减少重复开发，这在大的项目里是很有优势的。面向对象程序设计中，用户更多的去集中抽象现实中的问题，以人的思维方式为本位，减少了对机器物理构造或工作方式的迎合，可以用更多的经历去考虑怎么解决问题，怎么实现某些功能。

总体来说面向对象程序设计语言的特点是：

数据抽象化：通过从特定的实例中抽取共同的性质形成一般化的概念的过程。即对现实的问题抽象出一个模型，然后用数据来描述，以方便计算机处理。

数据封装：也叫数据隐藏，用户无需知道内部工作流程，只要知道接口和操作就可以了，C + + 中的一般用类来实现封装。这就好比一辆汽车，只需知道如何使用方向盘、刹车、离合器、油门等（即接口）去驾驶就可以了，汽车本身是如何实现行驶的被封装起来，不用驾驶者关心。

继承性：一种支持重用的意思，在现有的类型派生出新的子类，例如新型电视机在原

有型号的电视机上增加若干种功能而得到，新型电视机是原来电视机的派生，继承了原来电视机的属性，并增加了新的功能。

多态性：指在一般类中定义的属性或行为，被特殊继承之后，可以具有不同的数据类型或表现出不同的行为。例如“吃”这种行为，可以吃点心、吃饭、吃冰淇淋，同样是“吃”，却表现不同的吃行为。

动态联编：指一个计算机程序自身彼此关联的过程，按照联编所进行的阶段不同，可分为两种不同的联编方法：静态联编和动态联编。

1.3 C++语言的发展

C++是从C语言发展而来的，而C语言的历史可以追溯到1969年。在1969年，美国贝尔实验室的Ken Thompson为DEC PDP-7计算机设计了一个操作系统软件，这就是最早的UNIX。接着，他又根据剑桥大学的Martin Richards设计的BCPL语言为UNIX设计了一种便于编写系统软件的语言，命名为B。B语言是一种无类型的语言，直接对机器字操作，这一点和后来的C语言有很大不同。作为系统软件编程语言的第一个应用，Ken Thompson使用B语言重写了其自身的解释程序。

1972-1973年间，同在贝尔实验室的Denis Ritchie改造了B语言，为其添加了数据类型的概念，并将原来的解释程序改写为可以在直接生成机器代码的编译程序，然后将其命名为C。1973年，Ken Thompson小组在PDP-11机上用C重新改写了UNIX的内核。与此同时，C语言的编译程序也被移植到IBM 360/370、Honeywell 11以及VAX-11/780等多种计算机上，迅速成为应用最广泛的系统程序设计语言。然而，C语言也存在一些缺陷，例如类型检查机制相对较弱、缺少支持代码重用的语言结构等，造成用C语言开发大程序比较困难。

为了克服C语言存在的缺点，贝尔实验室的Bjarne Stroustrup博士及其同事开始对C语言进行改进和扩充，将“类”的概念引入了C语言，构成了最早的C++语言(1983)。后来，Stroustrup和他的同事们又为C++引进了运算符重载、引用、虚函数等许多特性，并使之更加精炼，于1989后推出了AT&T C++ 2.0版。随后美国国家标准化协会ANSI(American National Standard Instiute)和国际标准化组织ISO(International Standards Organization)一起进行了标准化工作，并于1998年正式发布了C++语言的国际标准ISO/IEC:98-14882。软件商推出的C++编译器都支持该标准，并有不同程序的拓展。

C++支持面向对象的程序设计方法，特别适合于中型和大型的软件开发项目，从开发时间、费用到软件的重用性、可扩充性、可维护性和可靠性等方面，C++均具有很大的优越性。目前，C++越来越受到重视并已得到了广泛的应用，许多软件公司为C++设计编译系统(编译系统可以理解为翻译家，把用户用高级语言编写的程序按照一定规则翻译成计算机能够理解的语言-机器语言，从而使计算机能够执行用户的程序)，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和Borland公司的Borland C++5.02等。因此利用C++设计并实现应用系统变得日益简单和快捷。

1.4 C++语言的特点

C++语言的主要特点在于支持面向对象程序设计，表现在：

1. 引入了类和数据封装特性

C++中的类是面向对象程序设计的基本支撑，类是数据抽象及信息隐藏的工具，对象是类的具体化。抽象对象的值和相关的操作被封装在一个类定义中。对象可被说明为给定类的变量，称之为实例，对象之间可以通过发送和接受消息相联系，接受消息的对象通过调用类的方法来实现相应的操作。

类可以包含一组构造函数和一个析构函数。构造函数是在每次创建一个特定刘象时由C++编译系统自动执行的类成员函数，析构函数是在特定的类的对象被撤消时自动执行的类成员函数。类的构造函数保证了在类的对象生成时可以自动进行初始化，类的析构函数保证对类的对象正常清除。

2. 访问限制和信息隐蔽性

类成员有公有、私有和保护成员，它们有不同的访问限制。声明为私有成员只能由类自己的函数访问；保护成员可以由该类及其派生类的成员函数访问；公有成员构成类的接口，允许所有函数访问。通过将成员设置为具有不同的访问权限，实现了信息隐蔽。友元是C++面向对象的另一个重要特征。通常类的私有成员禁止该类外面的函数和类对象直接访问，而友元机制允许有选择地突破这种限制。只要在类定义中声明非成员函数以及其他类为该类的友员函数或友元类，则这些友元就可以直接访问类的私有部分和保护部分，这样方便了数据访问，也在一定程度上破坏了数据封装。

3. 对象和消息机制

对象是面向对象程序设计的基本单位，通过向对象发送消息来实现对象的操作，根据消息的内容调用相应的方法。C++中的消息传递采用类似函数调用的机制。

4. 运算符重载和函数重载

C++允许为已有的函数和运算符更新赋予新的含义，使它们可以用于实际的需要。运算符重载和函数重载使得我们可以以更自然的表现方式实现对象的操作，提高程序的可读性。

5. 继承和派生

在程序中定义类时，会出现许多两个或多个类享有相似特征的情况。这时可以定义一个包含它们公共成员的基类，然后通过继承，从基类派生出其他类。在新的派生类中，为其增加新的操作和成员，改写基类的部分内容，可以得到更实用的类。派生类的引入有力地支持了面向对象的设计思想。

6. 虚函数和多态性

C++中的虚函数可以支持动态联编，从而也支持多态性。多态性是指在类中定义的属性或行为被派生类继承之后，可具有不同数据类型或表现出不同的行为，并允许在设计中使用更高级抽象。

1.5 建立一个C++程序

你的第一个程序。程序的开始一般都有包含一些头文件，这些文件里有已经定义好的各种操作，包含它之后就可以直接使用这些操作了。

```
#include <iostream.h> //包含头文件，后面要用到文件里的内容
```

然后写一个叫 main 的主函数，一个程序里有且仅有一个主函数，main 是函数名，它前面的 void 叫返回类型，void 表示程序什么也不返回，后面的括号是必须的，因为有时候函数需要参数，那么参数就写在这括号里，即使没有参数，这对括号也不能省略，如下所示：

```
void main( ) //主函数，计算机将首先从该函数开始执行
//每个函数体都用一对大括号把函数的内容括起来          } 一对大括号范围组成函数体
cout << "Hello, world! This is my first program." << endl;
```

函数体只有一个语句组成，该语句将在计算机屏幕上显示双引号内的内容，其中 cout 就是包含在头文件里的流对象，用来向标准设备（屏幕）输出的，“<<”称为插入运算符，用于输出，endl 表示向屏幕输出后换行，语句最后以分号结束。

下面再给出一段复杂些的代码，接着再逐行进行解释。这个程序将在当前目录（硬盘上）建立一个文件，并写入一些字符：

```
#include <fstream.h> //包含文件流头文件，其中有文件的操作
void main( ) // 程序从这里开始运行
{
ofstream SaveFile("cpp-home.txt"); //定义一个输出流对象 SaveFile
SaveFile << "Hello World!";
SaveFile.close( );
}
```

程序将在当前运行目录下建立一个名为 cpp-home.txt 的文件，并向它写入“Hello World!”。

下面给出各行的含义：

#include <fstream.h> —— 你需要包含此文件以使用C++的文件输入/输出函数。注意：一旦包含了这个文件，你不再需要（为了使用 cout/cin）包含 iostream.h，因为 fstream.h 已经自动包含了它。在这个头文件中声明了若干个类，包括 ifstream，ofstream 及 fstream，它们都继承自 istream 和 ostream 类。

```
ofstream SaveFile("cpp-home.txt");
```

* ofstream 即“output file stream（输出文件流）”。它将建立一个句柄（handle），以便我们以后能以一个文件流的形式写入文件。

* SaveFile：这是文件对象（句柄）的名字，当然这只是根据需要和含义起的名字，你还可以换用任何一个你想要的名称。

* “cpp-home.txt”：打开名为 cpp-home.txt 的文件。如果程序运行的当前目录已经存在这样一个文件，则它将被替换掉；万一不存在，程序也会为你创建一个新文件，你不

必为此而担心。

现在,让我们稍微深入一点。首先要指出的是:ofstream 是一个类。因此 ofstream SaveFile("cpp - home. txt"),这一语句将创建一个该类的对象;而我们在括号中所传递的参数实际上将传给构造函数:在这里将我们要建立的文件的名称作为实际参数传递给了该类的构造函数,以后的章节会有详细介绍。

SaveFile < <"Hello World!";,这是一个预定义好的运算符。这行语句所做的是将上面的文本写入文件。正如前面所提到的,SaveFile 是一个文件句柄,它关联一个打开的流式文件。所以,只须输入句柄名,再跟着输入" < <",然后接着写下一串用引号括起来的文本,就可以实现对文件的写入。如果我们想写入的是某个变量的值而不是带引号的文本,也只须将变量传递给句柄对象,像这样:

SaveFile < < variable; 就可以了!

SaveFile. close( );,既然我们打开了一个流文件,那么当用完它之后,就必须关闭它。SaveFile 是 ofstream 类的一个对象,而该类(ofstream)有一个用于关闭文件的成员函数,即 close( ) 函数。因此,我们只要依次输入文件句柄名、点号和 close( ),就可以关闭该文件!

注意:一旦你关闭文件,在你重新打开它以前,就再不能对它进行访问。

以上就是一个可以写文件的最简单的 C + +程序。的确很容易! 不过,有许多你可能还看不懂,那么本书将循序渐进地为你讲解更多的编程知识。

通过这个程序我们看到,一个 C + +程序的构成一般要有:

* 注释,这是良好编程风格的必不可少的条件;

* 编译预处理#include,通过这个命令,可将已有的编译好的文件内容插入到程序中,提高程序的可重用性;

* main 函数,一个最简单的 C + +程序就只有一个函数,那就是 main 函数,它是程序真正运行的地方,当然一个程序可以有很多函数构成,但有且只有一个 main 函数,函数之间可相互调用(除了 main 函数不能被用户调用外)。

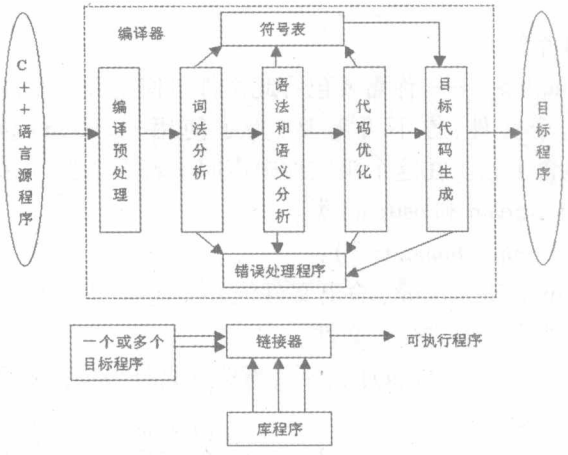

图 1 - 1　可执行程序生成流程

用程序设计语言写出的程序称为源程序(或称源代码),经编译系统处理后变为可执行文件,从而在计算机上运行使用,那么编译系统的处理过程是怎样的呢,图 1 -1 对此进行了简单的描述,当你的程序设计知识逐步积累时,也许是在学完了本书的内容时,将会对此有着更深刻的理解。

1.6 如何学习编程

有了中学的知识(特别是数学和英语)和对编程的兴趣,你就足以学好这门课程,相信不久你就会让计算机按照你的意愿来工作了,那将是一件美妙的事情。学好编程,需要你的勤学多练,注重培养你分析问题,解决问题的能力。对一些问题的解决还要建立在计算机能"理解"的基础上,因为你要用程序表示出来,计算机才能去计算,所以你考虑问题的角度要象程序员(未成为程序员之前)那样思考! 例如排队问题,现实当中我们去食堂排队买饭,是先来先买,那么计算机来解决这个问题,对先来先服务(买)的问题是用队列来实现的,队列就是从生活中抽象出来的数据模型(结构),所以学编程也需要你的抽象思维。

保持良好的编程风格是学好编程的另一个必要条件。程序不但能够高效率地解决问题,还要相互交流,供他人阅读,你也要阅读别人的程序,以便学习参考,这样一个良好的编程风格将让人赏心悦目,交流方便。也利于程序的后期维护。

1.7 Visual C + + 6.0 初步使用

本书的程序都是在 Visual C + +6.0 中测试通过的,因此有必要先介绍一下该编译环境的情况和使用。软件可购买到,安装方法很简单。现在的软件安装盘都可自动运行的,你只需根据提示点击下一步,安装序列号在光盘里有,采用默认设置安装即可。如果你对软件有了一些了解,也可以自定义安装,这样可只安装你需要的部分,以减少软件占用的磁盘空间。

1.7.1 Visual C + + 6.0 概述

Visual C + +6.0 是以 C + +作为语言、以 MFC 类库为基础的功能强大的可视化软件开发工具。1986 年 Borland 公司开发了 Turbo C + + ,随后又推出了功能上更完备的 Borland C + +语言。Microsoft 于 20 世纪 80 年代中期在 Microsoft C 6.0 的基础上开发了基于 DOS 环境的 Microsoft C/C + + 7.0,同时引入了基础类库 MFC1.0 版本。Visual C + + 6.0 是基于 Windows 操作系统的,编程中会经常调用操作系统的一些函数。

1.7.2 使用 Visual C + + 6.0

下面以 1.6 节给出的第一个程序为例介绍一下该开发环境的使用步骤。

软件安装在计算机上,默认路径一般在 C 盘里面,那么在开始菜单里就可找到启动该软件的快捷键选项,在"开始"菜单 - >"程序" - >"Microsoft Visual C + + 6.0" - >"Microsoft Visual C + + 6.0",点击最后一个选项,例如图 1 -2 是安装在 Window XP 环境下,打开的路径。则运行界面如图 1 -3 所示。

图 1－2　运行 Microsoft Visual C＋＋ 6.0

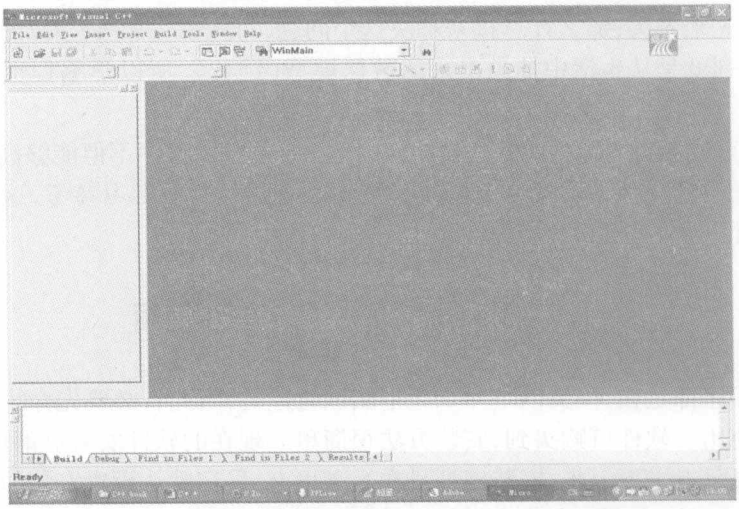

图 1－3　Microsoft Visual C＋＋6.0 运行后的界面

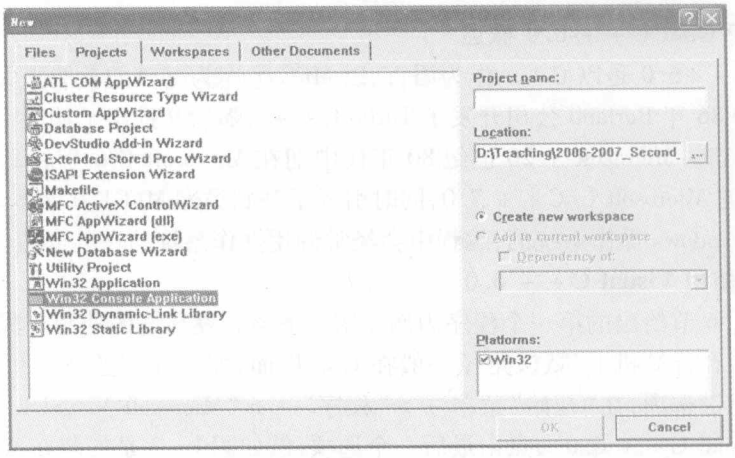

图 1－4　选择工程类型、填写工程名和保存路径