

C++面向对象程序设计

刘永华 主编
于春花 李晓波 副主编



高等学校计算机专业教材精选·算法与程序设计

C++面向对象 程序设计

刘永华 主 编
于春花 李晓波 副主编

清华大学出版社
北京

内 容 简 介

C++ 是一种混合型的程序设计语言,它既具有面向过程程序设计语言的特点,又具有面向对象程序设计语言的特点。

本书全面而又系统地介绍了 C++ 的基本概念、基本语法和基本的编程方法,较详尽地介绍了 C++ 面向对象的基本特性,主要包括类、对象、重载、继承和派生类、多态性和虚函数、模板、输入输出和异常处理等。

本书内容安排合理,讲解通俗易懂,各个知识点均辅以大量的典型例题,使广大读者能够通过学习本书掌握 C++ 的基本知识,并灵活运用所学知识编写应用程序。

本书每章都配有丰富的习题,题型丰富,使读者能随时自我检查学习效果并获得相应的反馈信息。本书最后还提供了上机实验的内容,使读者能深刻理解和领会面向对象程序设计的特点和风格,掌握其方法和要领。

本书可作为高等学校应用型专业本专科学学生学习 C++ 程序设计课程的教材,也可作为 C++ 自学者的教材或参考书。

本书封面贴有清华大学出版社防伪标签,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

C++ 面向对象程序设计 / 刘永华主编. —北京:清华大学出版社, 2011.6

(高等学校计算机专业教材精选·算法与程序设计)

ISBN 978-7-302-24780-7

I. ①C… II. ①刘… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 030830 号

责任编辑:白立军 张为民

责任校对:李建庄

责任印制:杨 艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京四季青印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:15.25 字 数:374 千字

版 次:2011 年 6 月第 1 版 印 次:2011 年 6 月第 1 次印刷

印 数:1~3000

定 价:25.00 元

产品编号:036734-01

出版说明

我国高等学校计算机教育近年来迅猛发展,应用所学计算机知识解决实际问题,已经成为当代大学生的必备能力。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。现在,很多高等学校都在积极探索符合自身特点的教学模式,涌现出一大批非常优秀的精品课程。

为了适应社会的需求,满足计算机教育的发展需要,清华大学出版社在进行了大量调查研究的基础上,组织编写了《高等学校计算机专业教材精选》。本套教材从全国各高校的优秀计算机教材中精挑细选了一批很有代表性且特色鲜明的计算机精品教材,把作者们对各自所授计算机课程的独特理解和先进经验推荐给全国师生。

本系列教材特点如下。

(1) 编写目的明确。本套教材主要面向广大高校的计算机专业学生,使学生通过本套教材,学习计算机科学与技术方面的基本理论和基本知识,接受应用计算机解决实际问题的基本训练。

(2) 注重编写理念。本套教材作者群为各高校相应课程的主讲,有一定经验积累,且编写思路清晰,有独特的教学思路和指导思想,其教学经验具有推广价值。本套教材中不乏各类精品课配套教材,并力图努力把不同学校的教学特点反映到每本教材中。

(3) 理论知识与实践相结合。本套教材贯彻从实践中来到实践中去的原则,书中的许多必须掌握的理论都将结合实例来讲,同时注重培养学生分析问题、解决问题的能力,满足社会用人要求。

(4) 易教易用,合理适当。本套教材编写时注意结合教学实际的课时数,把握教材的篇幅。同时,对一些知识点按教育部教学指导委员会的最新精神进行合理取舍与难易控制。

(5) 注重教材的立体化配套。大多数教材都将配套教师用课件、习题及其解答,学生上机实验指导、教学网站等辅助教学资源,方便教学。

随着本套教材陆续出版,相信能够得到广大读者的认可和支持,为我国计算机教材建设及计算机教学水平的提高,为计算机教育事业的发展做出应有的贡献。

清华大学出版社

前 言

C++ 语言(简称 C++)是近年来国内外广泛使用的面向对象程序设计语言之一,也是软件开发人员必须掌握的一种计算机程序设计语言。各类高等学校也普遍开设了 C++ 课程。本书是 C++ 入门教科书,适用于大学应用型本科和高职高专相关专业学生的培养。

本书内容共分 12 章。第 1 章为程序设计及 C++ 概述,主要介绍程序设计的基本概念、面向对象的软件开发过程、C++ 概述。第 2 章为 C++ 在非面向对象方向的实现,主要介绍 C++ 的标识符和数据类型、常量及变量、运算符和表达式、程序流程控制、数组、指针和引用、结构体、共用和枚举等。第 3 章为类和对象,主要介绍面向对象程序设计概述、类、对象和 this 指针等。第 4 章为构造函数和析构函数,主要介绍构造函数、析构函数、对象数组和对象指针、静态成员和友元等。第 5 章为重载,主要介绍函数重载和运算符重载及其应用。第 6 章为继承和派生类,主要介绍宏继承和派生类、派生类的构造函数和析构函数、多重继承。第 7 章为多态性和虚函数,主要介绍多态性、静态联编和动态联编、虚函数、抽象类和纯虚函数等。第 8 章为模板,主要介绍模板的基本概念、函数模板和类模板等。第 9 章为 C++ 的输入输出,主要介绍 C++ 流类库及其基本结构、格式化输入输出、文件操作等。第 10 章为异常处理,主要介绍异常处理机制的形式和执行过程。第 11 章为综合应用实例,主要通过一个综合实例“学生管理系统”介绍面向对象程序设计的一般方法。第 12 章为上机实验,供读者练习。

本书的编写力求在体系结构上安排合理、重点突出、难点分散、便于掌握;在语言描述上注重概念清晰、逻辑性强、通俗易懂、便于自学,每章均由学习目标、正文、小结和习题四部分组成。在学习每章之前,通过阅读学习目标能够了解本章的主要内容和学习后应达到的目标,使学习目标明确,增强学习的积极性和主动性;每章最后的小结是对本章内容的总结和归纳,使所学的内容条理化和系统化,进一步加深对本章内容的理解和认识。

本书由刘永华任主编,于春花、李晓波任副主编,陈茜、张淑玉、刘芳、孙俊香、解圣庆、刘贞德、李晓利、赵艳杰参与了部分章节的编写。

由于作者水平有限,加之编写时间仓促,书中难免存在缺点及不足之处,恳请广大读者和同行批评指正。

作 者

2011 年 4 月

目 录

第 1 章 程序设计及 C++ 概述	1
1.1 程序设计的基本概念	1
1.1.1 程序和程序文件	1
1.1.2 程序设计和程序设计语言	1
1.2 面向对象的软件开发过程	3
1.2.1 分析	3
1.2.2 设计	3
1.2.3 编程	3
1.2.4 测试	4
1.2.5 维护	4
1.3 C++ 概述	4
1.3.1 C++ 的发展与特点	4
1.3.2 简单的 C++ 程序	5
1.3.3 C++ 程序的开发过程	6
小结	6
习题	7
 第 2 章 C++ 在非面向对象方向的实现	8
2.1 标识符和数据类型	8
2.1.1 标识符	8
2.1.2 数据类型	9
2.2 常量及变量	10
2.2.1 常量	10
2.2.2 变量	12
2.3 运算符和表达式	14
2.3.1 各种运算符和表达式	14
2.3.2 表达式中的类型转换	19
2.4 程序流程控制	20
2.4.1 基本结构	20
2.4.2 选择结构	21
2.4.3 循环结构	24
2.5 数组	27
2.5.1 一维数组	28
2.5.2 二维数组	29

2.6	指针和引用	32
2.7	结构体、共用体和枚举	33
2.7.1	结构体	34
2.7.2	共用体	38
2.7.3	枚举	41
2.8	简单的输入输出	44
小结		46
习题		46
第3章	类和对象	49
3.1	面向对象程序设计概述	49
3.2	类	50
3.2.1	类的定义	50
3.2.2	成员函数的定义	53
3.3	对象	55
3.3.1	类与对象的关系	55
3.3.2	对象的定义	55
3.3.3	对象成员的引用	57
3.3.4	类的作用域	58
3.4	this 指针	60
小结		61
习题		61
第4章	构造函数和析构函数	66
4.1	构造函数	66
4.1.1	构造函数的定义与使用	66
4.1.2	带参数的构造函数	69
4.1.3	用成员初始化列表对数据成员初始化	71
4.1.4	带缺省参数的构造函数	73
4.1.5	拷贝构造函数	75
4.2	析构函数	78
4.3	对象数组和对象指针	80
4.3.1	对象数组	80
4.3.2	对象指针	82
4.4	静态成员	84
4.4.1	静态数据成员	84
4.4.2	静态成员函数	86
4.5	友元	88
4.5.1	友元函数	88

4.5.2 友元类	90
小结	92
习题	93
第5章 重载	98
5.1 函数重载	98
5.2 运算符重载	100
5.2.1 运算符重载概述	100
5.2.2 运算符重载的两种形式	101
小结	105
习题	106
第6章 继承和派生类	107
6.1 继承和派生类	107
6.1.1 继承与派生的概念	107
6.1.2 继承的作用	108
6.1.3 派生类的定义	108
6.1.4 派生类的继承方式	110
6.2 派生类的构造函数和析构函数	116
6.2.1 构造函数	116
6.2.2 析构函数	119
6.3 多重继承	121
6.3.1 多重继承的实现	121
6.3.2 多重继承的构造函数和析构函数	123
6.3.3 虚基类	126
小结	128
习题	129
第7章 多态性和虚函数	133
7.1 多态性概述	133
7.2 静态联编和动态联编	134
7.3 虚函数	134
7.3.1 虚函数的作用和定义	134
7.3.2 虚析构函数	137
7.4 抽象类和纯虚函数	138
7.4.1 纯虚函数	138
7.4.2 抽象类	139
小结	140
习题	141

第 8 章 模板	144
8.1 模板的概念	144
8.2 函数模板	145
8.2.1 函数模板的声明.....	145
8.2.2 函数模板的实例化.....	146
8.3 类模板	148
8.3.1 类模板的声明.....	148
8.3.2 类模板的实例化.....	149
小结.....	150
习题.....	151
 第 9 章 C++ 的输入输出	153
9.1 C++ 流类库及其基本结构	153
9.1.1 流的概念.....	153
9.1.2 基本 I/O 流类库	154
9.1.3 预定义的标准输入输出流.....	155
9.2 格式化输入输出	158
9.2.1 ios 类的格式状态标志和格式控制函数	159
9.2.2 流操纵符.....	163
9.3 文件操作	165
9.3.1 文件的概念.....	165
9.3.2 文件的打开与关闭	166
9.3.3 文件的读写.....	169
小结.....	175
习题.....	175
 第 10 章 异常处理	178
10.1 异常处理概述.....	178
10.2 C++ 的异常处理机制	179
10.2.1 异常处理的形式.....	179
10.2.2 异常处理的执行过程.....	180
小结.....	182
习题.....	182
 第 11 章 综合应用实例	184
11.1 面向对象程序设计的一般方法.....	184
11.2 综合实例：学生管理系统	185
11.2.1 问题描述.....	185
11.2.2 类设计.....	185

11.2.3	核心控制设计	185
11.2.4	源代码	185
第 12 章	上机实验	215
12.1	C++ 集成开发环境	215
12.1.1	进入和退出 Visual C++ 6.0 集成开发环境	215
12.1.2	C++ 程序的开发过程	216
12.2	上机实验	220
12.2.1	C++ 程序设计基础练习实验	220
12.2.2	类和对象实验	220
12.2.3	构造函数和析构函数实验	221
12.2.4	函数重载和运算符重载实验	221
12.2.5	派生类与继承实验	222
12.2.6	多态性与虚函数实验	222
12.2.7	函数模板与类模板实验	223
12.2.8	输入输出的格式控制实验	223
12.2.9	异常处理实验	223
12.2.10	综合练习实验	224
附录 A	ASCII 字符表	225
参考文献		227

第 1 章 程序设计及 C++ 概述

本章主要介绍程序设计的基本概念、算法的基本概念、C++ 语言(简称 C++)的特点、简单的 C++ 程序及 C++ 程序的开发。通过本章的学习,要求做到:

- 了解程序、程序文件、程序设计等基本概念。
- 了解算法的概念及其特征,熟悉常用的描述算法的方法。
- 了解 C++ 的基本特点。
- 熟练掌握简单 C++ 程序的基本结构,熟练掌握 C++ 程序开发的过程,学会最简单的 C++ 程序开发。

1.1 程序设计的基本概念

1.1.1 程序和程序文件

程序就是用程序设计语言对要完成的任务进行描述。

一个程序一般应包含以下两方面内容:一是对数据的描述,在程序中要指定数据的类型和数据的组织形式,即数据结构;二是对操作步骤的描述,也就是算法。

程序必须以文件的方式存储在计算机中,存放程序的文件称为程序文件。对于 C 语言,源代码程序文件约定的扩展名为 c。

1.1.2 程序设计和程序设计语言

1. 程序设计

所谓程序设计就是把解题步骤用程序设计语言描述出来的工作过程。简单的程序设计一般包含以下几个部分:

1) 确定数据结构

根据任务提出的要求、指定的输入数据和输出结果,确定存放数据的数据结构。

2) 确定算法

针对存放数据的数据结构来确定解决问题、完成任务的步骤。

3) 编写源代码程序

根据确定的数据结构和算法,使用选定的程序设计语言编写程序代码,简称编程。

4) 调试和运行程序

通过对程序的调试消除由于疏忽而引起的语法错误或逻辑错误;用各种可能的输入数据对程序进行测试,使之对各种合理的数据都能得到正确的结果,对不合理的数据能进行适当的处理。

2. 程序设计语言

人们利用计算机解决实际问题,一般要编写程序。程序设计语言就是用户用来编写程

序的语言,它是人与计算机之间交换信息的工具。

程序设计语言一般分为机器语言、汇编语言和高级语言三类。

1) 机器语言

机器语言是最底层的计算机语言。用机器语言编写的程序,计算机硬件可以直接识别和运行。在用机器语言编写的程序中,每一条机器指令都是二进制形式的指令代码。在指令代码中一般包括操作码和地址码,其中操作码告诉计算机作何种操作,地址码则指出被操作的对象。例如代码 10000000 表示加法操作,而代码 10010000 表示减法操作。

对于不同的计算机硬件(主要是 CPU),其机器语言是不同的。因此,针对一台计算机所编写的机器语言程序不能在另一台计算机上运行。由于机器语言程序是直接针对计算机硬件的,因此它的执行效率比较高,能充分发挥计算机的速度性能。但是,用机器语言编写程序的难度比较大,容易出错,而且程序的直观性比较差,也不容易移植。

2) 汇编语言

为便于理解与记忆,人们采用能帮助记忆的英文缩写符号(称为指令助记符)来代替机器语言指令代码中的操作码,用地址符号来代替地址码。例如 ADD 表示加法运算操作码, SUB 表示减法运算操作码。用指令助记符及地址符号书写的指令称为汇编指令(也称符号指令),而用汇编指令编写的程序称为汇编语言源程序。汇编语言又称为符号语言。

汇编语言也是与具体使用的计算机相关的。由于汇编语言采用了助记符,因此,它比机器语言直观,容易理解和记忆。用汇编语言编写的程序也比机器语言程序易读、易检查、易修改。但是,计算机不能直接识别源程序,必须由一种专门的翻译程序将汇编语言源程序翻译成机器语言程序后,计算机才能识别并执行。这种翻译的过程称为“汇编”,负责翻译的程序称为汇编程序。

3) 高级语言

机器语言和汇编语言都是面向机器的语言,一般称为低级语言。低级语言对机器的依赖性太大,用它们开发的程序通用性差,普通的计算机用户也很难胜任这一工作。

随着计算机技术的发展以及计算机应用领域的不断扩大,从 20 世纪 50 年代中期开始逐步发展了面向过程的程序设计语言,称为高级语言。高级语言与具体的计算机硬件无关,其表达方式接近于被描述的问题,人们易于接受和掌握。用高级语言编写程序要比低级语言容易得多,并大大简化了程序的编制和调试,使编程效率得到大幅度地提高。高级语言的显著特点是独立于具体的计算机硬件,通用性和可移植性好。

必须指出,用任何一种高级语言编写的程序(称为源程序)都要通过编译程序翻译成机器语言程序(称为目标程序)后计算机才能执行,或者通过解释程序边解释边执行。

面向对象的编程语言与以往各种编程语言的根本不同点在于,它设计的出发点就是为了能更直接地描述客观世界中存在的事物(这些事物称为对象)以及它们之间的关系。

开发一个软件是为了解决某些问题,这些问题所涉及的业务范围称为该软件的问题域。面向对象的编程语言将客观事物看作具有属性和行为(或称为服务)的对象,通过抽象找出同一类对象的共同属性和行为,形成类。通过类的继承派生和多态可以很方便地实现代码重用,大大缩短了软件开发周期,并使得软件风格统一。因此,面向对象的编程语言使程序能够比较直接地反映问题域的本来面目,软件开发人员能够利用人类认识事物所采用的一般思维方法来进行软件开发。

在面向对象的编程语言中应用最广泛的是在 C 语言基础上扩充出来的 C++。由于 C++ 对 C 语言兼容,而 C 语言又早已被广大程序员所熟知,所以,C++ 也就理所当然地成为应用最广的面向对象程序语言。

1.2 面向对象的软件开发过程

在整个软件开发过程中,编写程序只是其中的一小部分工作。软件开发的真正决定性因素来自前期概念问题的提出,而非后期的实现问题。只有分析、理解和正确表达了应用问题的内在实质,才能做出好的设计,然后,才是具体的编程实现。

早期的软件开发所面临的问题比较简单,从要解决的问题到编程实现并不是太难的事。随着计算机应用领域的扩展,计算机所处理的问题日益复杂,软件系统的规模和复杂度空前扩大,以至于软件的复杂性和其中包含的错误已达到软件人员无法控制的程度,这就是 20 世纪 60 年代初期的“软件危机”。软件危机的出现,促进了软件工程学的形成与发展。

学习面向对象的程序设计,首先应该对软件开发和维护的全过程有一个初步了解。因此,在本小节将首先简要介绍一下面向对象的软件开发过程。面向对象的软件开发过程主要包括如下内容:面向对象的分析(OOA)、面向对象的设计(OOD)、面向对象的编程(OOP)、面向对象的测试(OOT)和面向对象的软件维护(OOSM)。

1.2.1 分析

在分析阶段,要从问题的陈述着手,建立一个说明系统重要特性的真实情况模型。为理解问题,系统分析员需要与客户一起工作。系统分析阶段应该扼要精确地抽象出系统必须做什么,而不是关心如何去实现。

面向对象的系统分析,直接用问题域中客观存在的事物建立模型中的对象,无论是对单个事物还是对事物之间的关系,都保留它们的原貌,不做转换,也不打破原有界限而重新组合,因此能够很好地映射客观事物。

1.2.2 设计

在设计阶段,是针对系统的一个具体实现运用面向对象的方法。其中包括两方面的工作,一是把 OOA 模型直接搬到 OOD,作为 OOD 的一部分;二是针对具体实现中的人机界面、数据存储、任务管理等因素补充一些与实现有关的部分。

1.2.3 编程

编程是面向对象的软件开发最终落实的重要阶段。在 OOA 和 OOD 理论出现之前,程序员要写一个好的面向对象的程序,首先要学会运用面向对象的方法来认识问题域,所以 OOP 被看作一门比较高深的技术。现在,OOP 的工作比较简单了,认识问题域与设计系统成分的工作已经在 OOA 和 OOD 阶段完成,OOP 工作就是用一种面向对象的编程语言把 OOD 模型中的每个成分书写出来。

尽管如此,学习面向对象的程序设计仍然要注重学习基本的思考过程,而不能仅仅学习程序的实现技巧。

1.2.4 测试

测试的任务是发现软件中的错误,任何一个软件产品在使用之前都要经过严格的测试。在面向对象的软件测试中继续运用面向对象的概念与原则来组织测试,以对象的类作为基本测试单位,可以更准确地发现程序错误,提高测试效率。

1.2.5 维护

无论经过怎样严格的测试,软件中通常还是会存在错误,因此软件在使用的过程中,需要不断地维护。

使用面向对象的方法开发的软件,其程序与问题域是一致的,软件工程各个阶段的表示是一致的,从而减少了维护人员理解软件的难度。无论是发现了程序中的错误而追溯到问题域,还是因需求发生变化而追踪到程序,道路都是比较平坦的;而且对象的封装性使一个对象的修改对其他对象的影响很少。因此,运用面向对象的方法可以大大提高软件维护的效率。

1.3 C++ 概述

1.3.1 C++ 的发展与特点

C 语言是 20 世纪 80 年代以来迅速推广使用的一种程序设计语言。C 语言既具有高级语言的特点,表达力丰富,可移植性好;又具有低级语言的一些特点,能够很方便地实现汇编级的操作,目标程序效率较高。最初的 C 语言是贝尔实验室为了描述和实现 UNIX 操作系统、为之提供一种工作语言而设计的。因此十分适合于编写系统软件。但由于 C 语言的优异特性,各种应用程序的开发也纷纷采用 C 语言,使得 C 语言在各个应用领域中迅速扩展。其应用面之广,生命力之强,是其他高级语言无法相比的。

C++ 语言(简称 C++) 的早期版本叫做“带类的 C”,是 1980 年在贝尔实验室开发的。C++ 的重要目标就是在保留 C 语言原有精华的基础上提供全面的面向对象的编程支持,使得程序的结构更加清晰、更容易维护和扩充,同时又不丧失其高效性。C++ 对 C 语言做了大量的扩充,它在 C 语言的基础上增加了常值(const)数据、inline 函数、引用(reference)、函数及操作符重载等新的表现手段以及更严格的类型检查机制,同时支持面向对象概念,如对象、类、属性和方法、派生类与继承等等。C++ 也跟其前身 C 语言一样,已被广泛应用。例如,Visual C++ 和 C++ Builder 这两种广为应用的可视化开发工具都是以 C++ 为脚本语言的。

在 C 语言发展过程中,逐步克服了早期版本规范性不足的弱点,发展到现在的 C++,已今非昔比,其规范性和实用性都达到一个新的水平。C++ 现在得到了越来越广泛的应用,它继承了 C 语言的优点,并有自己的特点,主要特点如下:

(1) C++ 全面兼容 C 语言,这就使许多 C 语言代码不经修改就可以为 C++ 使用,用 C 语言编写的众多的库函数和实用软件可以用于 C++ 中。

(2) 用 C++ 编写的程序可读性更好,代码结构更为合理,可直接地在程序中映射问题

空间的结构。

(3) 生成代码的质量高,运行效率仅比汇编语言代码段慢 10%~20%。

(4) 从开发时间、费用到形成软件的可重用性、可扩充性、可维护性和可靠性等方面有了很大的提高,使得大中型的程序开发项目变得容易很多。

(5) 支持面向对象的机制,可方便地构造出模拟现实问题的实体和操作。

1.3.2 简单的 C++ 程序

下面给出一个简单的 C++ 程序,通过该程序可以对 C++ 程序有一个初步的了解。

【例 1.1】 输出信息“Hello World!”。

```
//helloworld.cpp
#include<iostream.h>
void main()
{
    cout<<"Hello World!"<<endl;
}
```

【程序分析】

C++ 程序的基本结构包括注释、编译预处理命令和程序主体。

(1) 注释有两种形式:第一种是“//”,注释内容紧跟其后,且须在一行内完成,如本例中的第一行;第二种是“/*”和“*/”,注释内容位于两者之间,可占多行。注释内容不参与编译,是专门为提高程序可读性而设计的。

(2) 编译预处理命令是“#”之后的指令。一条命令需在一行内完成,本例中的 `#include <iostream.h>` 称为文件包含命令,其作用是将文件 `iostream.h` 的内容成为本程序文件的一个组成部分。

(3) 程序主体由一个或多个函数组成,在所有函数中,只有一个函数是程序执行的入口,这个函数称为主函数,在控制台程序中,主函数定义为 `main()` 函数。本例的程序中即只包含一个 `main()` 函数,它表示了程序执行的开始点。

任意一个函数的描述由函数体来完成。函数体由“{}”括起,函数体由语句序列组成,每个语句由“;”结束。本例的函数体只有一条语句:

```
cout<<"Hello World!"<<endl;
```

语句的功能是将字符串“Hello World!”输出到屏幕。`cout` 是一个输出流对象,是 C++ 系统预定义的对象,包含许多有用的输出功能。输出操作由插入操作符“<<”配合完成。

本例在 Visual C++ 6.0 下运行成功,运行结果如图 1.1 所示。

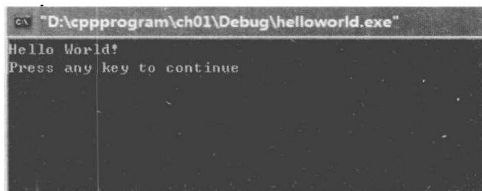


图 1.1 例 1.1 程序运行结果

1.3.3 C++ 程序的开发过程

C++ 程序的开发与其他高级语言源程序的开发原理是一样的,一般都要经过下列 4 个步骤:编辑、编译、连接和运行。

1. 编辑

编辑是应用程序开发的第一步,工作内容就是输入、修改程序。通常使用的编译系统都是集成化的,开发一个应用程序的所有工作,包括编辑、编译、连接和测试运行都可以通过它来完成。因此,虽然可以用任何一种文字处理软件来完成程序编辑工作,但一般还是使用编译系统内嵌的文字处理模块更方便一些。通过程序编辑而得到的程序称为源程序。源程序以纯文本格式保存在源程序文件中,C++ 程序文件的扩展名为 cpp。

2. 编译

编译是应用程序开发的第二步,工作内容就是分析程序中的源程序,生成目标程序,并保存在目标程序文件中。目标程序文件与相应的源程序文件的主名相同,但扩展名是 obj。虽然称为目标程序,而实际上是半成品,是不可执行的。在目标程序中还没有为函数、变量等分配具体的存储地址。

3. 连接

连接是应用程序开发的第三步,工作内容是将若干目标程序加以归并、整理,为所有的函数、变量分配具体的存储地址,生成可执行程序,并保存在可执行程序文件中。可执行程序文件与相应的目标程序文件及源程序文件的主名相同,但扩展名为 exe。如果说编译产生的是半成品,则连接就是把相关的半成品组装成最终产品——可执行程序。在被组装的这些半成品中,一部分是由程序设计者自己提供的,一部分是由合作者提供的,还有一部分是编译系统提供的。

4. 运行

运行是应用程序开发的第四步。一个 C++ 程序经过编译和连接后生成可执行文件。运行可执行文件即可得到源程序的运行结果。通过分析运行结果是否与要求相符,若相符则完成程序的开发,若不成功,重回第一步修改源程序及后面的步骤,直到成功为止。

C++ 程序的开发过程可用图 1.2 表示。

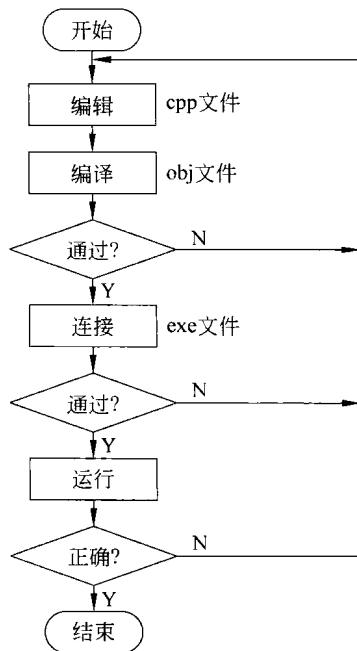


图 1.2 C++ 程序开发过程

小 结

本章简要地介绍了程序、程序文件、程序设计、程序设计语言、面向对象的软件开发过程等基本知识,以及 C++ 的特点及发展过程,通过一个简单的实例说明 C++ 程序的基本结构。C++ 程序开发的 4 个步骤是程序的编辑、编译、连接和运行。

习 题

一、填空题

1. C++ 的前身是_____语言。
2. 一个 C++ 程序的基本结构包含_____,_____和_____三大部分。
3. 若一应用程序的源程序文件的文件名为 test. cp, 经编译而生成的目标文件的文件名为_____。
4. 在 C++ 中有两种形式的注释: 第一种是“/ * ”和“ * /”, 第二种是_____。
5. C++ 程序是由一系列_____组成的, 一个 C++ 程序的源程序必须含有一个名叫_____的函数。
6. 面向对象的软件开发过程包括_____,_____,_____,_____和_____。
7. C++ 程序的开发过程分为_____,_____,_____和_____四个阶段。

二、单项选择题

1. 关于 C++ 和 C 语言的关系描述中, 错误的是()。
A. C 语言是 C++ 的一个子集
B. C 语言和 C++ 都是面向对象的语言
C. C++ 与 C 语言兼容
D. C++ 对 C 语言做了些改进
2. 下面描述错误的是()。
A. C++ 是一种面向结构化的程序设计语言
B. C++ 是一种面向对象的程序设计语言
C. C++ 是一种通用的程序设计语言
D. C 语言是一种面向对象的程序设计语言
3. 一个 C++ 程序的执行是()。
A. 从 main() 函数开始, 到 main() 函数结束
B. 从第一个函数开始, 到最后一个函数结束
C. 从任意一个函数开始, 到 main() 函数结束
D. 从 main() 函数开始, 到最后一个函数结束
4. 将目标文件进行(), 可以得到可执行文件。
A. 编辑 B. 编译 C. 连接 D. 运行
5. C++ 源程序文件默认扩展名为()。
A. cpp B. vcc C. c D. c++

三、编程题

1. 设计一个程序, 在屏幕在显示信息“Good Morning!”。
2. 设计一个程序, 计算地球万有引力。