

大学本科教材·计算机教学丛书

C++

程序设计

主 编：刘 娜 娜

参编者：刘 娜 娜 迟 立 颖 张 光 妲

陈 淑 鑫 刘 恩 军 张 景 林

主 审：邓 文 新



北京航空航天大学出版社

TP312/3154

2009

大学本科教材·计算机教学丛书

C++程序设计

主 编：刘娜娜

参编者：刘娜娜 迟立颖 张光姐
陈淑鑫 刘恩军 张景林

主 审：邓文新

北京航空航天大学出版社

内 容 简 介

本书全面、系统地介绍了 C++ 程序设计的基本概念、语法和编程方法,详尽地讲述了 C++ 语言的数据类型、基本语句及数组指针和引用、函数的作用、类和对象、继承和派生、多态性、虚函数和模板等内容。通俗易懂,由浅入深,重点突出,例题丰富,注重实际应用。内容精炼,结构合理,便于自学。每章后均备有相当数量的习题,可供学习者练习使用。本书不仅可作为高等学校 C++ 程序设计课程的教材,还可作为 C++ 语言的自学教材和参考书。

图书在版编目(CIP)数据

C++ 程序设计/刘娜娜主编. —北京:北京航空航天大学出版社,2009.3

ISBN 978-7-81124-597-4

I. C… II. 刘… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 208092 号

C++ 程序设计

刘娜娜 主编

责任编辑 许传安

*

北京航空航天大学出版社出版发行

北京市学院路 37 号(100191) 发行部电话:010-82317024 传真:010-82328026

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

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787×1092 1/16 印张:18 字数:461 千字

2009 年 3 月第 1 版 2009 年 3 月第 1 次印刷 印数:4 000 册

ISBN 978-7-81124-597-4 定价:28.00 元



总 前 言

随着科学技术、文化、教育、经济和社会的发展,计算机教学进入了我国历史上最火热的年代,欣欣向荣。就计算机专业而言,全国开办计算机本科专业的院校在2004年之初505所,到2006年已经发展到771所。另外,在全国高校中的非计算机专业,包括理工农医以及文科(文史哲法教、经管、文艺)等专业,按各自专业的培养目标都融入了计算机课程的教学。过去出版界出版了一大批计算机教学方面的各类教材,满足了一定时期的需求,但是还不能完全适应计算机教学深化改革的要求。

面对《国家科学技术中长期发展纲要(2006年—2020年)》制订的信息技术发展目标,计算机教学也要随之进行改革,以便提高培养质量。教学要改革,教材建设必须跟上。面对各层次、各类型的学校和各类型的专业都要开设计算机课程,就应有多样化的教材,以适应各专业教学的需要。北京航空航天大学出版社是以出版高等教育教材为主的,愿对计算机教学的教材建设做出贡献。

为计算机类教材的出版,北京航空航天大学出版社成立了“大学本科教材·计算机教学丛书”编审委员会。出版计算机教材,得到了北京航空航天大学计算机学院的大力支持。该院有三位教育部高等学校计算机科学与技术教学指导委员会(下称教指委)的成员参加编审委员会的工作。其他成员是北京航空航天大学、北京交通大学等6所院校和中科院计算技术研究所对计算机教育有研究的教指委成员、专家、学者和出版社的领导。

我们组织编写、出版计算机课程教材,以大多数高校实际状况为基点,使其在现有基础上能提高一步,追求符合大多数高校本科教学适用为目标。按照教指委制订的计算机科学与技术本科专业规范和计算机基础课教学基本要求的精神,我们组织身居教学第一线,具有教学实践经验的教师进行编写。在出书品种和内容上,面对两个方面的教学:一是计算机专业本科教学,包括计算机导论、计算机专业技术基础课、计算机专业课等;二是非计算机专业的计算机基础课程的本科教学,包括理工农医类、文史哲法教类、经管类、艺术类等的计算机课程。

教材的编写注重以下几点。

1. 基础性。具有基础知识和基本理论,以使学生在专业发展上具有潜力,便于适应社会的需求。
2. 先进性。融入计算机科学与技术发展的新成果;瞄准计算机科学与技术发展的新方向,内容应具有前瞻性。这样,以使学生扩展视野,以便与科技、社会发展的脉络同步。
3. 实用性。一是适应教学的需求;二是理论与实践相结合,以使学生掌握实用技术。

编写、出版的教材能否适应教学改革的需求,只有师生在教与学的实践中做出评价,我们期望得到师生的批评和指正。

“大学本科教材·计算机教学丛书” 编审委员会成员

主 任 马殿富

副主任 麦中凡

陈炳和

委 员 (以音序排列)

陈炳和 邓文新 金茂忠

刘建宾 刘明亮 罗四维

卢湘鸿 马殿富 麦中凡

乔少杰 谢建勋 熊 璋

张 莉



前 言

随着计算机技术的不断发展和应用领域的拓宽,软件开发的规模也在不断的扩大。C++程序设计在开发应用程序和解决实际问题中得到了广泛应用。

本书是作者根据多年的教学经验,在总结了教学中的遇到的各种问题的基础上编写而成的。主要特点是各章节结构清晰,内容由浅入深,通俗易懂,重点突出,适合初学者使用;每章配备了大量的习题,题型丰富,注重实际应用。

本书系统地讲述了C++语言的基本概念和编程方法,全书共分12章。第1章主要讲述面向对象程序设计的基本思想、引入面向对象的概念、C++程序的构成和Visual C++集成环境的使用。第2章主要讲数据类型、运算符和表达式,它是C++程序设计的基础,是C++程序的基本组成部分。第3章为预处理与程序结构,主要介绍了C++语言的基本程序结构,是C++程序的基本框架。第4章为数组,是C++中的一种典型的构造结构的数据类型,主要讲述了它的使用方法。第5章为函数,所有C++的程序都是由函数构成的。本章主要讲述了函数的定义及函数间的数据传递问题。第6章为指针,主要讲述了利用指针变量访问各种数据类型的方法以及指针在参数传递中的应用。从第7章开始讲述面向对象程序设计,主要介绍类的三大特性。第7章讲述了类和对象的基础,介绍了类和对象的定义和建立使用的方法。第8章为类和对象的应用,主要介绍了类和指针、数组的关系。通过例题进一步讲述类和对象的应用。第9章为类的继承和派生,主要讲述类的继承和派生的概念和简单应用。第10章为多态性和虚函数,主要通过例题介绍多态的概念,以及虚函数的应用。第11章为C++的I/O流类库,介绍了C++中数据流的概念和对外部设备的输入和输出操作。第12章为模板,主要介绍了函数模板和类模板的概念以及简单的应用。

本书所有的例题和习题都在Visual C++6.0环境下运行通过。这些例题和习题对学习、掌握C++程序的设计方法和编程技巧有很大的帮助。

全书由刘娜娜、迟立颖、张光姐、陈淑鑫、刘恩军、张景林编写。其中,刘娜娜编写第7~8章、第9章9.1~9.2节,迟立颖编写第10~11章、第12章的12.2节及附录Ⅳ,张光姐编写第3章、第6章,陈淑鑫编写第2章、第5章及附录Ⅲ,刘恩军编写第4章、第9章9.3节,张景林编写第1章、第12章12.1节及附录Ⅰ、Ⅱ。全书由刘娜娜、迟立颖统



稿,邓文新教授审阅。另外,在编写过程中,参考了一些中、外文图书资料和网上资料,在此对这些书的作者和译者提供的帮助表示衷心的感谢。

尽管在编写此书的过程中作者付出了艰辛和努力,但由于水平有限,书中难免有疏漏之处,敬请读者批评指正。

编者

2008 年 12 月



目 录

第 1 章 C++ 语言概述	1
1.1 面向对象程序设计的基本思想	1
1.1.1 面向对象程序设计	1
1.1.2 C++ 对面向对象程序设计方法的支持	2
1.2 C++ 程序的构成和书写形式	3
1.2.1 C++ 程序的构成和书写形式	3
1.2.2 C++ 程序设计风格	4
1.3 C++ 的字符集和关键字	5
1.3.1 字符集	5
1.3.2 标识符	5
1.3.3 关键字	5
1.4 C++ 程序的编写和实现	5
1.4.1 最简单的 C++ 程序	5
1.4.2 C++ 程序的实现	8
1.4.3 C++ 上机实践	9
习题一	11
第 2 章 数据类型、运算符和表达式	13
2.1 基本数据类型	13
2.1.1 整型(int)	14
2.1.2 浮点型数据(float)	15
2.1.3 字符型数据(char)	15
2.1.4 布尔类型(bool)	15
2.1.5 空类型(void)	16
2.2 常 量	16
2.2.1 常量定义	16
2.2.2 数值常量	16
2.2.3 字符常量	17
2.2.4 符号常量	19
2.3 变 量	20
2.3.1 变量定义	21
2.3.2 变量说明	21
2.3.3 常变量	22
2.4 运算符	22
2.4.1 算术运算符	23



2.4.2	关系运算符	24
2.4.3	逻辑运算符	24
2.4.4	位运算符	25
2.4.5	赋值运算符	26
2.4.6	其他运算符	27
2.5	表达式	29
2.5.1	表达式的种类	29
2.5.2	表达式中的类型转换	31
	习题二	32
第3章	预处理与程序结构	36
3.1	预处理	36
3.1.1	文件包含	36
3.1.2	条件编译	37
3.1.3	宏定义	39
3.2	顺序结构	42
3.3	选择结构	45
3.3.1	if 语句	45
3.3.2	switch 语句	48
3.4	循环结构	50
3.4.1	while 循环语句	51
3.4.2	do-while 循环语句	52
3.4.3	for 循环语句	53
3.4.4	循环嵌套	54
3.4.5	转向语句	55
	习题三	57
第4章	数 组	66
4.1	数组概述	66
4.2	一维数组	66
4.2.1	一维数组的定义	66
4.2.2	一维数组的应用	68
4.3	二维数组	70
4.3.1	二维数组的定义	70
4.3.2	二维数组的初始化	71
4.3.3	二维数组的应用	71
4.4	字符数组	74
4.4.1	字符数组的定义	74
4.4.2	字符数组的初始化	74
4.4.3	字符串	75
4.4.4	字符数组的应用	77



习题四	79
第5章 函 数	84
5.1 函数的定义与声明	84
5.1.1 函数的定义	84
5.1.2 函数的声明	85
5.2 函数的调用	87
5.2.1 函数值和 return 语句	87
5.2.2 函数调用的方式	87
5.2.3 函数参数	90
5.3 内联函数	92
5.4 函数重载	93
5.5 函数的嵌套调用和递归调用	95
5.5.1 函数的嵌套调用	95
5.5.2 函数的递归调用	96
5.6 作用域	100
5.6.1 变量的存储类别	100
5.6.2 全局变量与局部变量	102
5.6.3 外部变量	103
5.6.4 内部函数与外部函数	104
5.7 C++ 的系统函数	105
习题五	106
第6章 指 针	112
6.1 指针概念	112
6.1.1 地 址	112
6.1.2 指 针	112
6.2 指针的定义及使用	113
6.2.1 指针变量的定义	113
6.2.2 指针运算	116
6.3 指针与数组	119
6.3.1 用指针访问数组	119
6.3.2 指针与字符串	122
6.4 指针与函数	123
6.4.1 指向函数的指针	123
6.4.2 指针用做函数参数	124
6.4.3 返回值为指针的函数	125
6.5 C++ 语言的动态存储分配	126
习题六	127
第7章 类和对象的基础	133
7.1 类与对象的定义	133



7.1.1 类的定义	133
7.1.2 对象的定义	135
7.2 对象的初始化	137
7.2.1 构造函数	137
7.2.2 析构函数	138
7.2.3 复制构造函数	139
7.3 成员函数的特性	141
7.3.1 内联函数和外联函数	141
7.3.2 成员函数重载	142
7.3.3 设置参数的默认值	144
7.4 静态成员	145
7.4.1 静态数据成员	145
7.4.2 静态成员函数	146
7.5 友元	147
7.5.1 友元函数	147
7.5.2 友元类	148
7.6 类的作用域与对象的生存期	150
7.6.1 类的作用域	150
7.6.2 对象的生存期	152
第8章 类和对象的应用	157
8.1 类和指针	157
8.1.1 指向类对象的指针	157
8.1.2 指向类的成员的指针	158
8.1.3 this 指针	160
8.2 类和数组	161
8.2.1 对象数组	161
8.2.2 对象指针数组	162
8.2.3 指向对象数组的指针	163
8.3 常类型	164
8.3.1 常对象	164
8.3.2 常成员函数和常数据成员	164
8.4 子对象与堆对象	165
8.4.1 子对象	166
8.4.2 堆对象	167
习题八	169
第9章 继承性和派生类	173
9.1 基类和派生类	173
9.1.1 派生类的定义格式	173
9.1.2 派生类的继承特性	175



9.1.3 单继承派生类的构造函数和析构函数	178
9.2 多重继承	180
9.2.1 多重继承的概念	180
9.2.2 多重继承的声明	182
9.2.3 多重继承的构造函数和析构函数	182
9.3 虚基类	184
9.3.1 虚基类的作用	184
9.3.2 虚基类的定义	185
9.3.3 虚基类的构造函数和初始化	187
习题九	188
第10章 多态性和虚函数	194
10.1 函数重载	194
10.2 运算符重载	196
10.2.1 运算符重载的必要性	196
10.2.2 运算符重载的规则	196
10.2.3 运算符重载的实例	196
10.3 静态联编和动态联编	202
10.3.1 静态联编	202
10.3.2 动态联编	203
10.4 虚函数	204
10.5 纯虚函数和抽象类	206
10.5.1 纯虚函数	206
10.5.2 抽象类	208
10.6 虚析构函数	209
习题十	211
第11章 C++的 I/O 流类库	218
11.1 标准输出	219
11.1.1 预定义的插入符	219
11.1.2 put()成员函数	220
11.1.3 write()成员函数	221
11.2 标准输入	222
11.2.1 预定义提取符	222
11.2.2 get()成员函数	223
11.2.3 read()成员函数	225
11.3 格式控制 I/O 操作	226
11.3.1 流的格式化标志	227
11.3.2 格式化输出函数	229
11.3.3 操作子	229
11.4 插入符和提取符的重载	231



11.5 字符串流.....	232
11.5.1 ostringstream 类的构造函数.....	233
11.5.2 istream 类的构造函数.....	233
11.6 磁盘文件的 I/O 操作.....	234
11.6.1 磁盘文件的打开和关闭.....	234
11.6.2 文件流状态的判别.....	236
11.6.3 文本文件的读和写.....	238
11.6.4 二进制文件的读和写.....	240
11.6.5 随机文件的读和写.....	241
习题十一.....	243
第 12 章 模 板	249
12.1 函数模板.....	249
12.1.1 函数模板.....	249
12.1.2 模板函数.....	250
12.1.3 模板实参的省略.....	251
12.2 类模板.....	254
12.2.1 类模板的概念与定义.....	254
12.2.2 模板类.....	254
12.2.3 模板类的继承与派生.....	256
习题十二.....	258
附录 I C++ 的关键字列表.....	263
附录 II C++ 运算符的优先级和结合性.....	265
附录 III 常用系统函数.....	267
附录 IV ASCII 码表.....	270
参考文献.....	274



第1章

C++语言概述

C++语言是在C语言的基础上发展起来的。它包括了C语言的全部语言成分,同时又添加了尤为重要的面向对象编程的完全支持。它既可进行过程化程序设计,也可进行面向对象程序设计,是目前编程人员使用最广泛的语言工具。本章主要讲解面向对象程序设计的特点、C++程序的构成、语法及C++编程的基本方法和上机实践操作。



1.1 面向对象程序设计的基本思想

在面向对象的程序设计技术 OOP(Object-Oriented Programming)出现前,程序员们一般采用面向过程的程序设计方法 POP(Process-Oriented Programming)。面向过程的程序设计方法采用函数来完成对数据结构的操作,但又将函数和所操作的数据结构分离开来。但函数和它所操作的数据是密切相关的,特定的函数往往对特定的数据结构进行操作;如果数据结构发生改变,则相应的函数也要发生变化。这样用面向过程的程序设计方法来编写大程序不但难于编写,而且也难于调试、修改和维护。

面向对象的程序设计方法是对面向过程的程序设计方法的继承和发展。它汲取了面向过程的程序设计方法的优点,同时又考虑到现实世界与计算机世界的对应关系,现实世界中的实体就是面向对象程序方法中的对象。

现以常见的电视机为例来说明面向对象的程序设计方法。电视机内部有显像管、高压包、集成电路等很多复杂的元件,如果让用户直接去操作这些元件,那是相当困难的,需要有一定的专业知识才能去使用它。而现在呈现在人们面前的电视机,把内部的这些元件之间的详细构造全部封装起来,只给大家一个控制面板,就可以通过控制面板上的按钮来实现对电视机的操作,简单方便。这就是面向对象程序设计中所谓的“封装”,电视机就是“对象”,而对电视机的操作就是“方法”。下面就具体学习面向对象程序设计的方法。

1.1.1 面向对象程序设计

面向对象的程序设计是一种重要的程序设计方法。它能够有效地改进结构化程序设计存在的问题。在结构化的程序设计中,要解决某个问题,就要确定这个问题能够分解为哪些函数,数据能够分解为哪些基本的类型,即思考方式是面向机器实现的,不是面向问题的结构,需要在问题结构和机器实现之间建立联系。面向对象的程序设计方法的思考方式是面向问题的结构,认为现实世界是由对象组成的,而问题求解的方法与现实世界是对应的;因此,采用面向对象的程序设计方法来解决某个问题,则须先确定这些问题是由哪些对象组成的,这些对象之间是如何相互作用的。

面向对象程序设计将通过对数据和代码建立分块的内存区域,对程序进行模块化的一种程序设计方法。这些模块被用作样板——类,将其实例化成对象。面向对象的程序设计有 3



个特征,即封装、继承和多态。因此,面向对象程序设计方法要求语言必须具备抽象、封装、继承和多态性等关键要素。本书后几章将详细介绍面向对象的程序设计方法。

简单地讲,面向对象的分析设计方法可以分成以下4个步骤:

- (1) 找出问题中的对象和类。
- (2) 确定每个对象和类的功能,具有哪些属性,提供哪些方法等。
- (3) 找出这些对象和类之间的关系,确定对象之间的消息通信方式、类之间的继承和复合等关系。
- (4) 用程序代码实现这些对象和类。

可见,面向对象程序设计是将问题抽象成若干类,将数据与对数据的操作封装在一起,各个类之间可能存在着继承关系,对象是类的实例,程序是由对象组成的,通过对象之间相互传递消息、进行消息响应和处理来完成功能。面向对象的程序设计可以较好地克服结构化程序设计存在的问题,可以开发出健壮、易于扩展和维护的应用程序。

1.1.2 C++对面向对象程序设计方法的支持

C++和C语言均诞生于贝尔实验室。Bjarne Stroustrup博士在C语言中引入了面向对象的思想,并将这种语言命名为C++。C++是C语言的扩展,是C语言的一个超集,根据Stroustrup博士自己的说法,C++是一个“更好的C语言”。它是一种混合型的语言,既支持传统的结构化程序设计,又支持面向对象程序设计。

C++设计的初衷是为了扩充C语言,并引入面向对象程序设计思想。C语言虽然有其强大的功能,但作为一种结构化编程语言,当程序量相当大时,其局限性不可避免地暴露出来。而C++以其对面向对象程序设计方法的支持,成为设计和开发大规模软件的强有力工具。同时,C++在设计时充分考虑了与C语言的兼容性,使用大量基于C语言的开发工作得以继承和发展,许多C语言的编写代码不须修改就可为C++语言所用,而且原来用C语言编写的众多库函数和实用程序也可以用于C++语言中。C++和C的主要区别是C++对数据抽象和面向对象程序设计方法的支持。C++允许数据抽象、支持封装、继承和多态等特征。C语言程序的设计一般采用自上而下、逐步求精的方式进行软件开发;而C++则是兼有自下而上和自上而下两种方式。目前,C++语言已被广泛用于程序设计的众多领域。实践证明,C++尤其适用于大、中型软件的开发。

1. C++支持数据封装

支持数据封装就是支持数据抽象。在C++中,类是支持数据封装的工具,对象则是数据封装的实现。

2. C++类中包含私有、公有和保护成员

C++类中可以定义三种不同访问控制权限的成员。一种是私有(private)成员,只有在类中说明的函数才能访问该类的私有成员,而在该类外的函数不可以访问私有成员;另一种是公有(public)成员,类外面也可访问公有成员,成为该类的接口;还有一种是保护(protected)成员,这种成员只有该类的派生类可以访问,其余的在这个类外不能访问。

3. C++中通过发送消息来处理对象

C++中是通过向对象发送消息来处理对象的。每个对象根据所接收到的消息的性质来决定需要采取的行动,以响应这个消息。为此送到一个对象的所有可能的消息在对象的类描



述中都需要定义。

4. C++中允许友元破坏封装性

类中的友元可以不遵守私有成员不允许该类外面的任何函数访问的规定,友元可以访问该类的私有成员,友元打破了类的封装性,可在类外定义函数称友元函数,也可以是在类外定义整个类称为友元类,这是C++面向对象的重要特征。

5. C++允许函数名和运算符重载

C++支持多态性,函数名重载和运算符重载都属于多态性。多态性是指相同的语言结构可以代表不同类型的实体,或者对不同类型的实体进行操作。C++允许一个相同的标识符或运算符代表多个不同实现的函数,被称为标识符或运算符的重载。用户可以根据需要定义标识符重载或运算符重载。

6. C++支持继承性

C++中允许单继承和多继承。一个类可以根据需要生成派生类。派生类继承了基类的所有方法,另外派生类自身还可以定义所需要的不包含在父类中的新方法。一个子类的每个对象包含有从父类那里继承来的数据成员以及自己所特有的数据成员。

7. C++支持动态联编

C++中可以定义虚函数,通过定义虚函数来支持动态联编。动态联编是多态性的一个重要特征。多态性形成由父类和子类组成的一个树型结构。

以上概述了C++在面向对象程序设计的一些主要特征支持。有关这些支持的实现将在后续章节中介绍。



1.2 C++程序的构成和书写形式

C++是一个复杂的程序设计语言。为帮助读者养成良好习惯,编写更清晰、更易读易懂、更易维护、更易测试和调试的代码,我们从编写第一个C++程序设计开始,就应该遵循一定的程序设计风格,通过长期的训练和积累,使得良好的程序设计风格成为习惯。

1.2.1 C++程序的构成和书写形式

一个C++程序可以由一个程序单位或多个程序单位构成。第一个程序单位作为一个文件。在程序编译时,编译系统分别对各个文件进行编译,因此,一个文件是一个编译单元。

在一个程序单元中,可以包括以下几个部分:

- (1) 预处理命令。
- (2) 全局声明部分(在函数外的声明部分)。在这部分中包括对用户自己定义的数据类型的声明和程序中所用到的变量的定义。
- (3) 函数。函数是实现操作的部分,因此函数是程序中必须有的和最基本的组成部分。每一个程序必须包括一个或多个函数,其中必须有一个(而且只能有一个)主函数(main函数)。

一个函数由两部分组成:

- (1) 函数首部,即函数的第一行。包括函数名、函数类型、函数属性、函数参数(形参)名、参数类型。一个函数名后面必须跟一对圆括号,函数参数可以默认。



如: `int max(int x, int y)`

(2) 函数体, 即函数首部下面的大括号内的部分。如果在一个函数中有多个大括号, 则最外层的一对 `{ }` 为函数体的范围。

函数体一般包括:

1) 局部声明部分(在函数内的声明部分)。包括对本函数中所用到的类型、函数的声明和变量的定义。对数据的声明既可以放在函数之外(其作用范围是全局的), 也可以放在函数内(其作用范围是局部的, 只在本函数内有效)。

2) 执行部分。由若干个执行语句组成, 用来进行有关的操作, 以实现函数的功能。

当然, 在某些情况下也可以没有声明部分。甚至可以既无声明部分, 也无执行部分。如 `void demp() { }`, 是一个空函数, 什么也不干, 但这是合法的。

语句包括两类。一类是声明语句, 如“`int a, b;`”, 用来向编译系统通知某些信息(如类型、函数和变量的声明或定义); 但它并不引起实际的操作, 是非执行语句。另一类是执行语句, 用来实现用户指定的操作。C++对每一种语句赋予一种特定的功能。语句是实现操作的基本成分。显然, 没有语句的函数是没有意义的。C++语句必须以分号结束, 如“`c=a+b;`”, 分号是语句的一个组成部分。注意, 没有分号的就不是语句。

一个C++程序总是从 `main` 函数开始执行的, 而不论 `main` 函数在整个程序中的位置如何。`main` 函数可以放在程序文件的最前头; 也可以放在程序文件的最后; 或在一些函数之前, 在另一些函数之后。

类(class)。类是C++新增加的重要的数据类型, 是C++对C的最重要的发展。有了类, 就可以实现面向对象程序设计方法中的封装、信息隐蔽、继承、派生、多态等功能。在一个类中可以包括数据成员和成员函数。它们可以被指定为私有的(private)和公有的(public)属性。私有的数据成员和成员函数只能被本类的成员函数所调用。

C++程序书写格式自由, 一行内可以写几个语句, 一个语句可以分写在多行上。C++程序没有行号, 也不像FORTRAN或COBOL那样严格规定书写格式(语句必须从某一列开始书写)。一个好的、有使用价值、可读性强的源程序都应当加上必要的注释。

1.2.2 C++程序设计风格

所谓程序设计风格, 是指借助于好的设计方法编写结构好的程序。在编写源程序时, 往往需要采用各种措施来提高程序的可读性、可理解性和可修改性, 以利于程序的查错、测试、维护修改及交流。

(1) 以简洁明了的方式编写C++程序, 是一种良好的程序设计习惯。通常称这种编写程序的方法为KIS(Keep it Simple, 保持简洁)。

(2) 缩排规则就是使程序的书写格式应能较好地反映出该程序的层次结构。例如, 处于同一层的语句都从同一个字符位置开始书写; 将每个函数的整个函数体在定义函数体的花括号中缩排一级, 可使程序中的函数结构更明显, 使程序更易读。

(3) 标识符选择能够反映相关功能和特征的单词来命名, 以便于见名知意。

(4) 注释是增加可读性、可理解性的常用措施。注释虽然不是程序的有效部分, 但是, 给程序添加合适的注释是一种良好的程序设计风格。注释应该视为程序的一个重要组成部分。

(5) 注意大小写英文字母, C++语言中是区分英文字母的大小写的。