新手学编程系列

打开C++程序设计大门的金钥匙

新手学 C++

袁 超等 编著

7小时多媒体视频讲解

DVD-ROM

◎ 由浅入深：从基本概念开始讲解，逐步深入到实际开发
◎ 重点突出：着重介绍了C++中的类、重载、继承和多态等特性
◎ 实例丰富：讲解知识点时穿插了265个实例，有较强的实用性
◎ 面向就业：提供了常见面试题，帮助读者了解入职面试的相关知识
◎ 视频教学：提供了7小时多媒体教学视频，学习起来更加直观

北京希望电子出版社
Beijing Hope Electronic Press
www bhp com cn

新手学编程系列

新手学 C++

袁 超等 编著

7 小时多媒体视频讲解

DVD-ROM

北京希望电子出版社
Beijing Hope Electronic Press
www bhp com cn

内 容 简 介

本手册以标准 C++为蓝本，全面介绍了 C++基础知识及常用工具的用法。

本手册共分四篇，主要内容涉及：预备知识、C++概述、变量和基本类型、表达式、语句、数组、字符串、指针、函数、标准库类型、类、重载操作符、继承性、多态性、命名空间、模板、异常处理与错误等内容。

作者依据多年的 C++使用经验，总结出学习 C++的初学者最需要的知识与学习方法，帮助学习者用最少的时间来获得最大的学习效果。文中通过大量实例讲解 C++的知识点，精选的实例简洁、通俗，非常容易学懂与掌握。本手册适合 C++的初学者，也可作为大专院校计算机、软件工程相关专业学生使用。

本光盘内容为实例源代码、语音视频教学及电子教案（PPT）。

需要技术支持的读者，请与北京清河 6 号信箱（邮编：100085）发行部联系，电话：010-62978181（总机）转发行部、010-82702675（邮购），传真：010-82702698，E-mail：tbd@bhp.com.cn。

新手学 C++ / 袁超等编著. —北京希望电子出版社，2010
（新手学编程系列）
ISBN 978-7-89498-990-1

Ⅰ. 新… Ⅱ. 袁… Ⅲ. C++—程序设计

责任编辑：周凤明　/ 责任校对：马　君
责任印刷：双　青　/ 封面设计：一品设计

北京希望电子出版社 出版
北京市海淀区上地三街 9 号金隅嘉华大厦 C 座 611
邮政编码：100085
http://www.bhp.com.cn

北京四季青双青印刷厂印刷

北京希望电子出版社发行　各地新华书店经销

*

2010 年 1 月第 1 版　开本：787mm×1092mm 1/16
2010 年 1 月第 1 次印刷　印张：32.375
印数：1—3 000　字数：753 千字

定价：49.80 元（配 1 张 DVD）

前　言

程序设计语言比比皆是，人们为什么要学习 C++语言？由于 C++语言具有超强的适应性，学会了 C++语言，再学习其他语言就都不是问题了。

为了方便广大读者学习 C++语言，笔者花费了半年时间进行写作。本手册全面介绍了 C++方方面面的知识，并以实例方式讲解使用 C++语言开发应用程序的方法与技巧。学完本手册，力求使读者具有项目编程的本领。

特点

1．实例丰富，实用性强

本手册通过大量生动的实例来讲解知识点，既避免了学习的枯燥也可以加深对知识点的理解，还可以立即通过实践手册中的实例来体会编程的乐趣，增强了初学者的信心。另外，还对实用性强的知识进行了详细介绍，方便以后的实践。

2．讲解通俗易懂，案例融合情景

通俗易懂的讲解可以大大帮助初学者入门。如果开始讲解的内容过于高深，势必造成看不懂的结果，所以本手册对知识点进行了通俗易懂的讲解，使读者不仅知道怎么做，还要知道为什么这样做。

另外，本手册在知识点讲解时引入了相应的情景，如引入了动物园给一群猴子分桃子的情景，来学习使用求余操作符计算剩余的桃子，使学习更加的有乐趣。

3．内容充实，层次分明

本手册对 C++中的基础知识点进行了全面讲解，并对重要内容进行了详细讨论，并逐步加深，其中涵盖了基础知识点、标准库的使用、类的三大特性以及部分高级主题。

在学习基本知识点之后可以编写简单的程序，学习标准库可以使程序更加强大，本手册对类的三大特性用了较大的篇幅进行讲解，读者可以初步感受到面向对象编程的乐趣，最后通过高级主题的内容，可以使编程的方法更加的多样化。

适合的读者

- C++的编程爱好者。
- 从事软件开发的程序员。
- 大中专院校的学生。
- 社会培训班学生。

作者

本手册主要由袁超主笔。其他参与编写的人员有卜庆玲、冯曼菲、匡妍娜、雷成健、李

小波、刘浩然、刘会神、马震、齐志华、舒军、孙大林、王辉、王沛、王石、王晓悦、熊英、张杰、袁福庆、赵显琼、韩延峰、李刚、张佳楠、张金霞、孔鹏、张昆、陈刚、姚志鹃等，在此一并表示感谢。

因编者水平有限，书中错漏之处在所难免，恳请广大读者指正，相关问题请与martt0656@163.com联系。

编著者

目 录

第 1 篇 简介

第 2 篇 基础入门

第 4 篇 高级主题

第1篇　简介

第 1 章　开始

“C++”这个词到底读作什么？国内的程序员通常读作“C 加加”，而国外的程序员通常读作“C plus plus”。在深入学习 C++之前，有必要对 C++的发展有一个基本的了解。为什么会有 C++？它是从何而来的？以及将来会怎么发展？本章将对 C++作一个简单的介绍。学完本章后读者将会对 C++的来龙去脉有一个大致的了解。学习 C++，就不得不提面向对象编程，本章的任务之一就是让读者初步了解什么是面向对象编程，以及面向对象编程有什么特点。

1.1　C++简史

下面将对 C++的发展历史进行介绍，也许读者在看本书之前已经学习过 C 语言，这将对学习 C++更加有益。如果读者没有学过 C 语言也不要紧，本书将一步步地对 C++进行详细介绍。

1.1.1　C++发展历程

C++源于 C 语言，而 C 语言是在 B 语言的基础上发展起来的。可以说 B 语言是 C++语言之父。

1960 年出现了 ALGOL 60 语言，它是一种面向问题的高级语言，这种语言离硬件设备比较远，不宜用来编写系统程序。为了克服 ALGOL 60 离硬件设备较远的缺点，1963 年英国剑桥大学在 ALGOL 60 语言的基础上推出了 CPL（Combined Program- ming Language）语言。

CPL 语言比 ALGOL 60 语言在硬件方面更接近了一些，但规模比较大，难以实现。为了克服这个缺点，1967 年英国剑桥大学的 Matin Richards 对 CPL 语言进行了简化，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，又做了进一步简化，设计出更简单、并且很接近硬件设备的 B 语言（取 BCPL 的第一个字母）。似乎 B 语言已经很先进了，有了这么多的优点，并且，人们用 B 语言编写了第一个 UNIX 操作系统，并在 PDP-7 上实现了此系统。1971 年，人们又在 PDP-11/20 上实现了 B 语言，并编写了 UNIX 操作系统。

前面提到的仅仅是 B 语言的优点，但由于它过于简单、功能有限，所做的事情很少，这时设计一个功能强大的语言就成为最急切的需要，1972 年至 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（取 BCPL 的第二个字母）。

C 语言既保持了 BCPL 和 B 语言的优点：语言精练，而且接近硬件，又克服了它们的缺点：语言过于简单，而且数据没有类型。最初设计 C 语言的目的是为了给 UNIX 操作系统提

供一种工作语言，所以 UNIX 操作系统里面的编程用到的就是 C 语言。

现在，C 语言已风靡全世界，成为世界上应用最广泛的几种计算机语言之一，中国的很多大学首选的计算机语言就是 C 语言，笔者最先学习的计算机语言也是 C 语言。

然而，即使克服了前面一些语言的缺点，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，这就造成用 C 语言开发大程序比较困难，这时似乎又有一件事要做了，那就是再开发一种新语言来克服 C 语言的缺点，的确，现实也是这样的。为了克服 C 语言存在的缺点，贝尔实验室的 Bjarne Stroustrup 博士及其同事对 C 语言进行了改进和扩充，将“类”的概念引入到 C 语言中，构成了最早的 C++语言（1983 年），这也就是最早期的 C++语言，称为带类的 C，Bjarne Stroustrup 博士自然就成为 C++之父。

后来，Stroustrup 和他的同事们又将运算符重载、引用、虚函数等特性引入了 C++，并且使它更加精练。1989 年推出了 AT& TC++ 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）一起对其进行了标准化工作，并于 1998 年正式发布了 C++语言的国际标准 ISO/IEC:98-14882。

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，所以 C++成为了软件开发的热门语言之一。同时，C++之父 Bjarne Stroustrup 博士为了使以前人们编写的数百万行的 C 语言的程序代码还能继续使用，将 C 语言设计成为 C++的一个子集，这就使得 C 代码不经修改就可被 C++编译通过。目前，C++受到越来越多的重视，而且应用十分广泛，成为商业软件开发中占统治地位的语言。

C++语言是在前面几种语言的基础上，摒弃缺点，保留优点发展而成的语言，是集众家精华所成，但它又有自己独特的优点，是一门功能强大的语言。

C++语言是人们公认的难学的语言之一，原因就是由于它过于强大。但读者只要坚持努力学习是能学会的。

1.1.2　应先学习 C 语言吗

学习 C++之前，有必要先学习 C 语言吗？有些人认为，学 C++之前，最好学一下 C 语言。首先要明确的是，我们要学的是 C++语言，而不是 C 语言，C++语言是一个完整的语言，因此完全可以直接学习 C++语言。当然，以前学过 C 语言更好，这样学 C++更容易入门，但是如果没有学过 C 语言也不要紧，因为我们要学的是 C++，本书的目的就是让以前没有学过其他语言的读者能够快速入门。

在上一节 C++发展历程中已经介绍了，C++语言是从 C 语言的基础上发展起来的，C++根除了 C 语言中存在的问题，并保证与 C 语言兼容。有人说 C++语言是 C 语言的一个超集，虽然说得不完全，但从另一个侧面说明了可以直接学习 C++语言。绝大多数的 C 语言代码无需修改就可以直接在 C++程序中使用，这样便使得 C 程序员更容易学习 C++语言。现在不用担心了吧？

1.1.3 C++与面向对象编程（OOP）

本节开始之前，有关C++的概念可能是第一次接触，所以如果发现自己不懂不要紧，因为本书的目的就是让大家弄懂这些东西。既然学习C++就不得不提面向对象编程（OOP）。本节主要介绍C++与面向对象编程（OOP）的相关知识。

C++语言是一种应用较广的面向对象的程序设计语言，使用它可以实现面向对象的程序设计。面向对象的设计与面向过程的设计是有很大区别的，面向对象的程序设计在面向过程的程序设计的基础上有了一个质的飞跃。要学会面向对象的程序设计，首先要学会一种面向对象的语言，即要学会用VC编程，就要先有C++的基础，而学习C++语言首先要认识它面向对象的特性和实现面向对象的方法。

面向对象编程（Object Oriented Programming，OOP，面向对象程序设计）是一种计算机编程架构。它的出现以20世纪60年代的simula语言为标志。20世纪80年代中后期，面向对象程序设计逐渐成熟，被计算机界理解和接受，人们又开始进一步考虑面向对象的开发问题。

传统的软件结构和设计方法难以适应软件生产自动化的要求，因为它以过程为中心进行功能组合，软件的扩充和复用能力很差。

对象是对现实世界实体的模拟，因此能更容易地理解需求，即使用户和分析者之间具有不同的教育背景和工作特点，也可以很好地进行沟通。

当首次接触C++时，总会碰到一些在C语言从未见过的概念，如类、对象、抽象、封装、继承、多态性、虚函数等。这些概念是C++所具有的，现在就让我们来看一下面向对象的特征。总的来说，面向对象有三个基本特征，即封装、继承和多态，当然还有其他的特征，在此仅作简要说明。

1．数据封装

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。面向过程的程序设计方法与面向对象的程序设计方法在对待数据和函数关系上是不同的，在面向对象的程序设计中，将数据和对该数据进行合法操作的函数封装在一起作为一个类的定义，数据将被隐藏在封装体中，该封装体通过操作接口与外界交换信息，如图1.1所示。

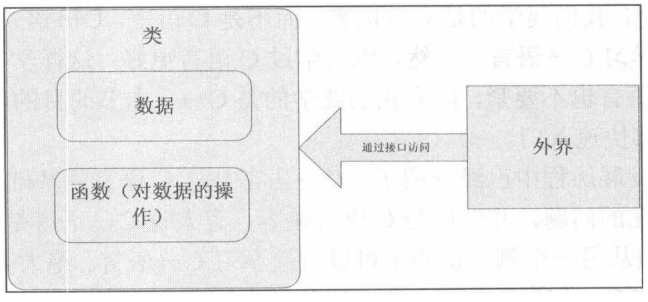

图1.1 类的封装

对象被说明为具有一个给定类的变量，类类似于C语言中的结构，在C语言中可以定义结构，但这种结构中包含数据，而不包含函数。C++中的类是数据和函数的封装体。在C++

中，结构可作为一种特殊的类，它虽然可以包含函数，但是它没有私有或保护的成员。举个简单的例子：人就可以是一个类，它包括头、手和脚等基本的组成成分，还包括相应的活动，如跑步、吃饭和睡觉等。

对象就是一个具体的人，如老师就是一个对象。老师是一个实实在在的人，看得见摸得着，老师的手就是老师的手而不是学生的手，老师的脚就是老师的脚也不是学生的脚。老师吃饭就是老师吃饭，吃完饭老师就饱了，不会出现因为老师吃完饭后学生就饱了的情况，如图 1.2 所示是关于人的类和它一个老师的例子。

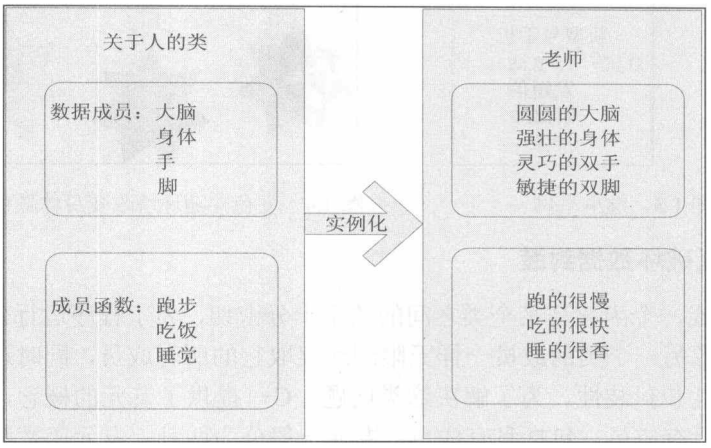

图 1.2　关于人的类的实例

上图中的“圆圆的大脑”、“强壮的身体”等仍然很抽象，实际上老师的大脑和身体的其他部位是具体的，不过我们的目的是通过简单的说明来使读者明白对象与实例的区别。

2．继承性

C++支持继承性，C++中可以允许单继承和多继承，单继承说明派生类的基类有且只有一个父类，多继承则允许有多个父类。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都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。以手机为例，现在手机的功能越来越多，新型号手机拥有旧型号手机的功能，而且还开发出了许多新的功能，这里旧型号手机就是基类，新型号手机就是派生类，新型号手机继承了旧型号手机的功能，并开发了新的功能，如图 1.3 所示。

3．多态性

当不同的对象接受到相同的消息名（或者说当不同的对象调用相同名称的成员函数）时，可能引起不同的行为（执行不同的代码）。这有点像变色龙，能根据环境的变化而改变身体的颜色，达到保护自己的目的，如图 1.4 所示。

4．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。响应消息的是一系列的方法，方法是在类定义中使用

函数来定义的，使用一种类似于函数调用的机制把消息发送到一个对象上。就像一个机器人，它会根据使用者的要求来完成特定的任务。吸尘器机器人就是一个很好的例子，当吸尘器机器人收到使用者的不同消息时，会根据事先编写好的程序，执行相应的动作。

图 1.3　继承举例

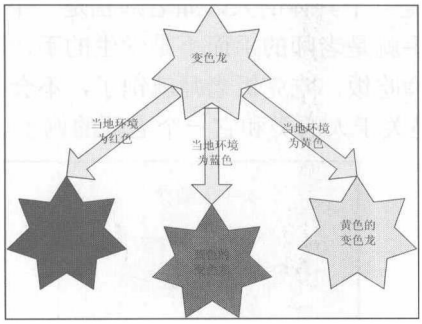

图 1.4　变色龙随环境改变身体颜色

5．友元可以破坏数据封装

有时两个类或一个函数与某个类之间的关系十分密切，为了程序运行效率着想，希望这个函数和类可以像另一个类的成员一样无限制地存取它的所有成员，同时还希望在形式上尽量保持类的独立性和封装性。为了解决这类问题，C++提供了友元的概念，一个类的友元可以存取这个类的所有成员，包括私有成员。友元一般分为两种：友元函数和友元类。友元的这种特性保证了编程的灵活性。

6．类中包含的各种成员

C++类中定义了三种不相同访问控制权限的成员。

（1）私有成员：只有在类中说明的函数才能访问的成员，而在该类外的函数不可以访问私有成员，但友元可以打破这一规则。

（2）公有成员：不仅在类中可以访问，在外面也可以访问公有成员，它成为该类的接口。

（3）保护成员：保护成员只有在该类的派生类中可以被访问，在这个类外不能被访问。

关于私有成员、公有成员与保护成员的关系如图 1.5 所示。

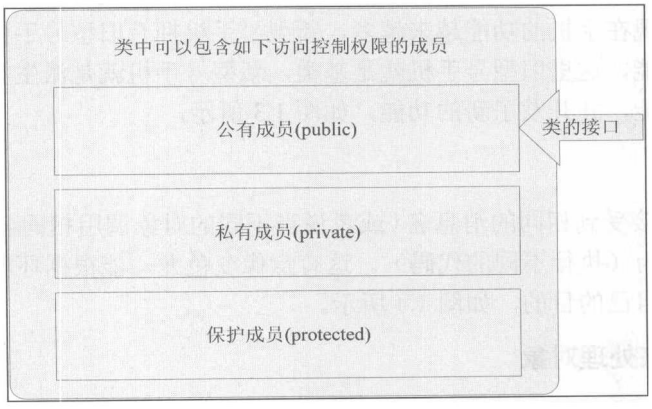

图 1.5　类中的三种不相同访问控制权限的成员