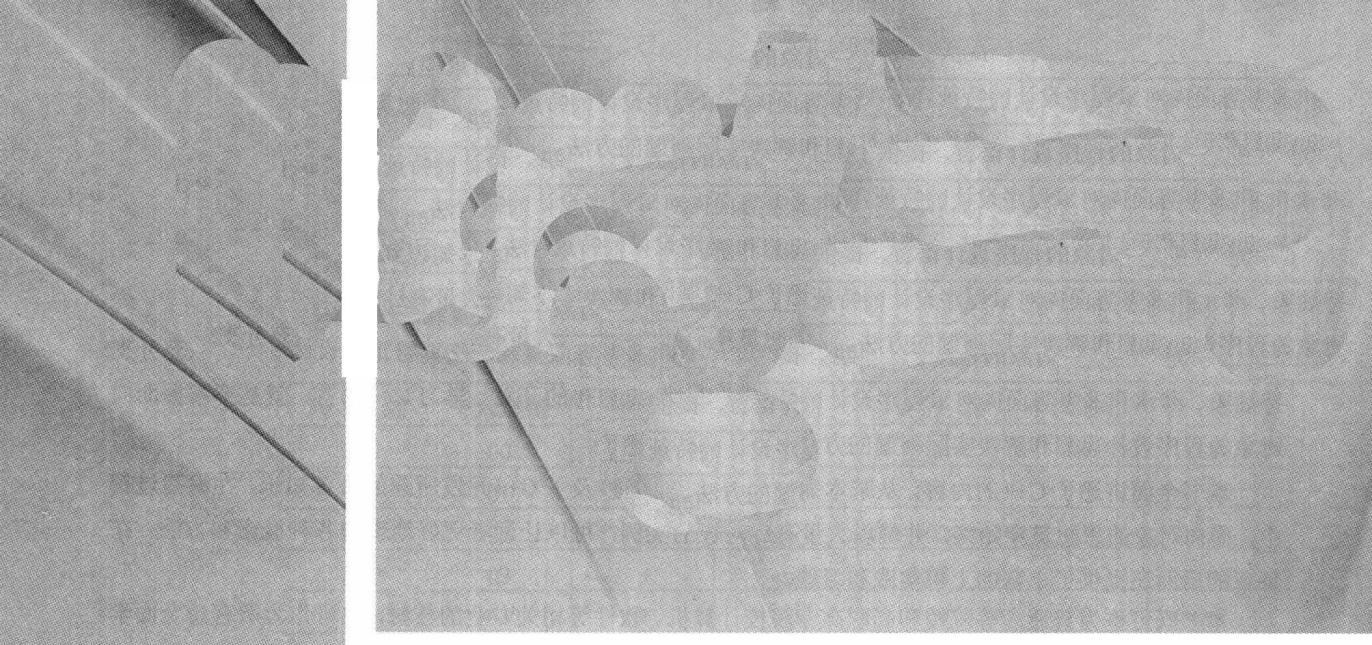
C++

面向对象程序设计

李丽平　丁宏伟◎主　编
石彦芳　刘丽华◎副主编

- 直接提出面向对象的程序设计思想
- 概念讲解形象、贴切、透彻
- 案例丰富，结构清晰，讲解通俗，深入浅出
- 免费提供配套教学资源

清华大学出版社



C++

面向对象程序设计

李丽平　丁宏伟◎主　编

石彦芳　刘丽华◎副主编

清华大学出版社

北京

内 容 简 介

C++是一种面向对象的程序设计语言，提供了类、模板、函数重载和运算符重载设计等功能，充分支持抽象、继承和多态等面向对象程序设计的特征，方便大型软件的开发。学习C++语言，就是要掌握面向对象的程序设计思想和解决实际问题的方法。

本书全面讲述了C++的内容，从基本知识到核心概念，涉及了C++开发所需的必备知识。在编写过程中，面向对象的思想贯穿始终，并辅以大量有针对性的实例，可以让读者更好地理解各种概念和方法。在每章的后面还提供了丰富的上机实践和习题。

本书既可作为普通高等院校和高职高专院校计算机、软件等相关专业的教材，也可作为所有想全面学习C++开发技术的人员和使用C++进行开发的工程技术人员的工具书。

图书在版编目（CIP）数据

C++面向对象程序设计/李丽平，丁宏伟主编. —北京：清华大学出版社，2011.9

ISBN 978-7-302-27002-7

I. ①C…　II. ①李…　②丁…　III. ①C语言-程序设计　IV. ①TP312

中国版本图书馆CIP数据核字（2011）第193141号

责任编辑：贾小红
封面设计：刘　超
版式设计：文森时代
责任校对：姜　彦
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：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185×260　**印　张**：19　**字　数**：439千字
版　　次：2011年9月第1版　　**印　　次**：2011年9月第1次印刷
印　　数：1～4000
定　　价：34.00元

产品编号：044208-01

前　言

C++语言是当今 IT 领域最流行的程序设计语言之一，广泛应用于系统软件及各种大型应用软件的开发。目前，国内高校普遍开设了“面向对象程序设计”类的课程，一些院校更是将 C++语言作为程序设计语言课程的首选。为此，我们在总结多年的教学、培训及开发实践经验的基础上编写了本书。

本书针对程序设计的初学者，以面向对象的程序设计思想为主线，以通俗易懂的方式介绍 C++语言，引导读者以最自然的方式将人类惯有的面向对象的思维方法运用到程序设计中来。本书的宗旨是培养读者面向对象编程的基本能力，因此，在知识体系设计与章节安排上独具匠心，并通过先进的教学理念和深入浅出的讲解风格，循序渐进地展开教学内容。本书具有以下特点：

1．直接提出面向对象的设计思想，读者更容易接受与理解

由于 C++语言既支持面向过程的程序设计方法，又支持面向对象的程序设计方法，所以传统的 C++教材都是先从面向过程的设计思想开始讲授。这样做的缺点是，从面向过程转向面向对象时，读者接受起来会非常困难。本书直接讲授面向对象的程序设计思想，读者更容易接受和理解。

2．概念讲解形象、贴切、透彻，适合初学者学习

本书语言形象生动，在讲解各类概念时，多给予了形象、具体的解释，并且通过实例做了进一步阐述，使读者不仅能知其然，还能知其所以然，在第一次接触这些概念时就能迅速掌握。

3．实例丰富，加深读者的理解

本书在讲解知识点时，贯穿了大量有针对性的实例，使读者在实际的编程体验中能更好地理解各种概念和方法，加深其对 C++语言内涵和精髓的理解程度。

本书共分为 12 章，全面介绍了 C++面向对象程序设计的有关概念与语法，每个章节后还配备了上机实践与习题。书中所有实例程序均在 Visual C++ 6.0 上运行通过。

各章内容如下：

第 1 章　引入面向对象方法。主要介绍面向对象的程序设计方法与思想，并通过一个简单的 C++示例使读者对 Visual C++ 6.0 运行环境有一定的了解。

第 2 章　类和对象的初步认识。主要介绍类和对象的概念以及定义方法；C++的变量与函数、运算符与表达式。

第 3 章　类和对象的提高篇。在第 2 章的基础上进一步讨论类和对象，包括构造函数、析构函数和静态成员等。

第 4 章　流程控制。重点介绍流程控制语句，包括顺序控制语句、选择控制语句和循环控制语句。利用这些流程控制语句，可以让程序的执行逻辑更合理，编码更简单。另外，还简单介绍了变量的作用域。

第 5 章 数组与指针。主要介绍一维数组的定义、初始化与引用；字符数组；指针的用法以及函数参数的传递方式等。

第 6 章 友元。友元机制是对封装机制的补充，它给了程序员更大的灵活性，可以提高程序的运行效率。本章主要介绍友元函数与友元类。

第 7 章 多态性。主要介绍静态多态性，包括函数重载与运算符重载。

第 8 章 继承性与派生类。继承是面向对象程序设计的重要特征，是使代码可以复用的最重要的方法之一。本章详细介绍 C++继承和派生的方法。

第 9 章 动态多态性。多态性是面向对象程序设计的重要特征之一。本章重点介绍动态多态性，包括虚函数，抽象类等。

第 10 章 异常。主要介绍 C++中异常的概念以及处理异常的方法。

第 11 章 模板。主要介绍模板的概念，包括函数模板与类模板。

第 12 章 文件的输入与输出。主要介绍文件、文件流的概念，以及如何从文件中输入数据并将处理的结果输出到文件等。

本书由李丽平、丁宏伟主编，石彦芳、刘丽华任副主编，赵清晨参编。其中，第 6、7、9、11 章由李丽平编写，第 1、2、8 章由丁宏伟编写，第 4、5 章由石彦芳编写，第 3、10 章由刘丽华编写，第 12 章由赵清晨编写。全书的整理、审校工作由李丽平、丁宏伟负责。

由于作者水平有限，书中难免有不足之处，恳请读者批评指正。

编　者

2011 年 8 月

目　录

第 1 章　引入面向对象方法

20 世纪 90 年代以来，面向对象的程序设计（Object Oriented Programming，OOP）方法迅速地在全世界流行，并一跃成为程序设计的主流技术。现在，越来越多的软件开发人员采用面向对象的程序语言进行软件开发，这主要是因为面向对象的程序开发模式更接近于人的思维活动，可以在很大程度上提高开发人员的编程能力，并有效减少软件的后期维护成本。

本章主要介绍面向对象的程序设计方法，并通过一个简单的 C++示例使读者对 Visual C++ 6.0 的运行环境有一定的了解。面向对象的程序设计方法和我们之前接触到的编程方法有很大的区别，并且在理解上有一些难点，希望读者能认真体会。

1.1　程序设计基础

1.1.1　计算机语言的种类

我们都知道，使用计算机进行工作时，需要通过操作相应的软件来实现。例如，使用 Word 软件可以进行文字处理，使用 Excel 软件可以进行数据统计和表格处理，使用数据库软件可以实现大量数据的存储与管理等。这些软件都是由专业的软件开发人员设计和编写的。

一般来说，日常工作中遇到的问题多数都可以借助现成的应用软件完成，但有时仍然需要为某个具体问题自行开发一些软件。特别是在工程应用领域，经常会遇到大量的具体问题，因此在使用通用软件时，不仅效率低下，而且可能无法完成任务。在这种情况下，自行编制一些具有针对性的应用软件可能是解决问题的唯一方法。

为计算机编写软件时需要使用程序设计语言。目前可用的计算机语言种类非常多，总的来说可以分成机器语言、汇编语言和高级语言 3 大类。

早期的程序员们使用机器语言编写程序。机器语言由 0、1 二进制代码构成，是计算机唯一能够识别和执行的语言，因此执行效率极高。但由于机器语言难以记忆和识别，因此手工编写机器语言程序非常繁琐，且非常容易出错。

为了克服机器语言难读、难编、难记和易出错的缺点，人们将一些使用频率较高的算法，用能帮助人们记忆的英文缩写来表示（如用 ADD 表示加法），于是出现了汇编语言，也叫助记符语言。汇编语言比用机器语言的二进制代码编程要方便些，在一定程度上简化了编程过程。

但汇编语言和机器语言在本质上是相同的，都属于低级语言，即都是面向机器的语言。这类语言和具体机器的指令系统有着密切的关系，由于不同型号的计算机其汇编语言指令系统是不相同的，因此，针对同一问题编制的汇编语言程序在不同种类的计算机间是互不

相通的。而且，使用汇编语言，要求程序员必须十分熟悉计算机的硬件结构和其工作原理，这对于非计算机专业的人员来说是很难做到的，对计算机的应用推广也非常不利。

高级语言指的是像 C/C++、Basic、Pascal 等与具体机器无关的语言。采用高级语言进行编程，程序员就不再需要了解计算机的内部结构，而只要按照程序的语法编写程序即可。高级语言具有更强的表达能力，可方便地表示数据的运算和程序的控制结构，能更好地描述各种算法，而且易于学习和掌握，是目前绝大多数编程者的选择。

高级语言所编制的程序称为源程序，不能直接被计算机识别和执行，必须经过转换才能被执行。转换的方式有解释和编译两种。

解释方式类似于日常生活中的口译，即一边将应用程序源代码由相应语言的解释器“翻译”成目标代码（机器语言程序），一边执行，因此效率比较低，而且不能生成可独立执行的可执行文件，应用程序不能脱离其解释器，但这种方式比较灵活，可以动态地调整和修改应用程序。

编译方式可以在应用程序执行之前就将源代码“翻译”成目标代码（机器语言程序），因此其目标程序可以脱离其语言环境独立执行，使用方便，效率较高。但应用程序一旦需要修改，必须先修改源代码，再重新编译生成新的目标文件才能执行，只有目标文件而没有源代码，修改起来很不方便。现在大多数的编程语言都是编译型的，如 C++、Delphi 等。

1.1.2 面向对象程序设计

程序设计指的是设计、编写和调试程序的方法与过程。程序是软件的本体，软件的质量主要通过程序的质量来体现，因此，研究一种切实可行的程序设计方法至关重要。

1．面向过程的程序设计方法

所谓“面向过程”，是指从功能的角度分析问题，将待解决的问题分解成若干个功能模块，每个功能模块描述一个操作的具体过程。结构化程序设计方法是一种典型的面向过程的设计方法，它由 E • W.dijkstra 在 1969 年提出，是以模块化设计为中心，将待开发的软件系统划分为若干个相互独立的模块，使完成每一个模块的工作变得单纯而明确，为设计一些较大的软件打下良好的基础。

结构化程序设计方法的核心包括以下几个方面。

（1）自顶向下、逐步求精的开发方法。结构化程序设计方法将分析问题的过程划分成若干个层次，每一个新的层次都是对上一个层次的细化，即步步深入，逐层细分。

（2）模块化的组织方式。结构化程序设计方法将整个系统分解成若干个模块，每个模块实现特定的功能，最终的系统将由这些模块组装而成。模块之间通过接口传递信息，模块划分应尽可能达到高内聚，低耦合。

（3）结构化的语句结构。结构化程序设计方法只使用顺序、选择和循环 3 种基本结构进行程序设计，任何算法功能都可以通过这 3 种基本结构组合、嵌套构成。

结构化程序设计仍然存在诸多问题，如抽象级别较低、封装性较差、软件代码重用程度较低和软件维护困难等。针对结构化程序设计的缺点，人们提出了面向对象的程序设计

方法。

2．面向对象的程序设计方法

所谓“面向对象”，是指以对象为中心分析、设计和构造应用程序的机制。面向对象的程序设计方法，则是指用面向对象的方法指导程序设计的整个过程。

（1）对象

对象是面向对象程序设计方法中最基本和最核心的概念。

现实世界中，任何事物都是一个对象，它可以是一个有形的、具体存在的事物，如一本书、一辆汽车、一个工厂，甚至一个地球；它也可以是一个无形的、抽象的概念或事件，如学校的校规、企业规章、一场乒乓球比赛、一次到商场的购物过程等。对象既可以很简单，也可以很复杂，复杂的对象可以由若干简单的对象构成，整个世界可以认为是一个非常复杂的对象。

不同的对象具有不同的特征和功能。例如，工厂具有工厂的特征和功能，购物过程具有购物过程的特征和功能。由此可见，现实世界中的对象具有如下 3 个特征。

① 有一个名字用来唯一标识该对象。

② 用一组状态来描述对象的某些特征。

③ 用一组操作来实现其功能。

例如，有一个学生对象，姓名叫王小五，性别为男，学历为大专，专业为软件开发与设计，可从事软件开发、软件测试的工作。这里，“王小五”是这个对象的名字，“男性”、“软件开发与设计专业”和“大专学历”是这个学生的特征，“能从事软件开发、软件测试的工作”是这个学生具有的能力（功能）。

面向对象程序设计中的“对象”和现实世界中“对象”的概念相似，指描述其状态的数据以及对这些数据施加的一组操作封装在一起构成的统一体。简单地说，对象就是数据和操作的封装体。现实世界中，对象的能力通常称为操作或行为；面向对象的程序设计方法中，对象的能力通常称为方法或服务，对象的状态数据通常称为属性。在各种不同的支持面向对象的高级语言中，数据和操作的术语是不同的。在 C++语言中，属性称为数据成员，而服务称为成员函数。图 1-1 形象地描绘了对象。

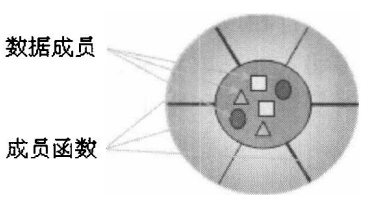


图 1-1　对象

（2）类

类是面向对象程序设计方法中的另一个重要概念。现实世界中，类是对一组相似对象的抽象描述。例如，作为学生对象，有张小三、李小四、王小五等，每个对象有不同的性别、专业和学历特征，有从事不同行业的能力。而学生类则是对学生这类对象所应具有的

共同特征和能力集合的抽象描述，即学生这类对象应具有性别、专业和学历特征，应具有从事某种行业的能力。

类和对象之间的关系是抽象和具体的关系。类是对多个对象进行综合抽象的结果，一个对象是类的一个实例。例如，“学生”是一个类，它是由千千万万个具体的学生抽象而来的一般概念。而具体到某一个学生对象王小五，则是学生类的一个实例。图 1-2 描述了类和对象之间的关系。

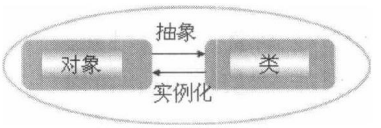


图 1-2　类和对象之间的关系

在面向对象的程序设计方法中，类是具有相同属性和服务的一组相似对象的抽象，或者说，类所包含的属性和服务描述了一组对象的共有属性和服务。也可以这么理解，类是建立某个具体对象时使用的“模型”或“模板”。编程时，总是先声明类，再由类生成对象。因为类是建立对象的“模板”，按照该“模板”可以建立多个具体的对象或实例。

这就好比月饼的制作过程：先雕刻一个有凹下图案的木模，然后抹油并将事先揉好的面塞进木模里，用力挤压后，将木模反扣在桌上，一个漂亮的月饼就制作好了。反复操作，可以制作出多个外形一模一样的月饼。这里，木模就好比是“类”，月饼就好比是“对象”。

（3）消息

现实世界中的对象不是孤立存在的实体，它们之间存在着各种各样的联系，正是它们之间的相互作用、联系和连接，才构成了世间各种不同的系统。同样，在面向对象方法中，对象之间也需要联系，称为对象的交互。面向对象程序设计技术必须提供一种机制，允许一个对象与另一个对象的交互，这种机制称为消息传递。

以实际生活为例，每个人可以为他人服务，也可以要求他人为自己服务。当需要别人为自己服务时，必须告诉他们我们需要的是什么服务，也就是说，要向其他对象提出请求，其他对象接到请求后，才会提供相应的服务。

在面向对象方法中，一个对象向另一对象发出的请求称为消息。当对象接收到发向它的消息时，就调用有关的方法，执行相应的操作。消息是一个对象要求另一对象执行某个操作的规格的说明，通过消息传递才能完成对象之间的相互请求或相互协作。例如，有一个教师对象张大三和一个学生对象李小四，李小四可以发出消息，请求张大三演示一个实验，当张大三接收到这个消息后，确定应完成的操作并执行。

一般情况下，称发送消息的对象为发送者或请求者，称接收消息的对象为接收者或目标对象。对象中的联系只能通过消息传递来进行。接收者或目标对象只有在接收到消息时，才能被激活，然后根据消息的要求完成相应的功能。

（4）面向对象程序设计的基本特征

面向对象程序设计方法模拟人类习惯的解题方法，代表了计算机程序设计的新颖的思维方法。这种方法的提出是对软件开发方法的一场革命，是目前解决软件开发面临困难的

最有希望、最有前途的方法之一。面向对象程序设计具有以下4个基本特征。

① 抽象

抽象是人类认识问题最基本的手段之一，它忽略了一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象是对复杂世界的简单表示，强调感兴趣的信息，忽略不重要的信息。例如，在设计一个学籍管理系统的过程中，考察某个学生对象时，只关心他的姓名、学号和成绩等，而可以忽略他的身高、体重等信息。

抽象在系统分析、系统设计以及程序设计的发展中一直起着重要的作用。在面向对象程序设计方法中，对一个具体问题的抽象分析的结果，是通过类来描述和实现的。

例如，在学籍管理系统中，对学生进行归纳、分析，抽取出其中的共性，可以得到姓名、学号和成绩等共同的属性，它们组成了学生的数据抽象部分；数据输入、修改和输出等共同的行为，则构成了学生的行为抽象部分。

如果开发一个学生健康档案程序，所关心的特征就有所不同了。可见，即使对同一个研究对象，由于所研究问题的侧重点不同，也可能产生不同的抽象结果。

② 封装

在现实世界中，所谓封装就是把某个事物包围起来，使外界不知道该事物的具体内容。在面向对象程序设计中，封装是指把数据和实现操作的代码集中起来放在对象内部，并尽可能隐蔽对象的内部细节。对象好像是一个不透明的黑盒子，表示对象属性的数据和实现各个操作的代码都被封装在黑盒子里，从外面是看不见的，更不能直接访问或修改这些数据及代码。使用一个对象时，只需知道它向外界提供的接口形式而无需知道它的数据结构细节或实现操作的算法。

对象的封装机制可以将对象的使用者与设计者分开，使用者不必知道对象行为实现的细节，只需要使用设计者提供的接口让对象去执行。封装的结果实际上隐藏了复杂性，并提供了代码重用性，从而降低了开发一个软件系统的难度。

③ 继承

继承在现实生活中是一个很容易理解的概念。例如，每个人都从父母身上继承了一些特性，如种族、血型和眼睛的颜色等。我们身上的特性来自于父母，也可以说，父母是我们所具有的属性的基础。

再以动物学中对动物继承性的研究为例。图 1-3 说明了哺乳动物、狗和柯利狗之间的继承关系。哺乳动物是一种热血、有毛发、用奶哺育幼仔的动物；狗是哺乳动物，具有哺乳动物的所有特性，同时还具有区别于其他哺乳动物（如猫、大象等）的特征，如有犬牙、食肉、特定的骨骼结构、群居等；柯利狗是尖鼻子、身体红白相间、适合放牧的狗，同样具有狗的所有特征。

在继承链中，每个类继承了它前一个类的所有特性，图 1-3 中的继承关系是：狗是哺乳动物，柯利狗是狗，即狗类继承了哺乳动物类的特性，柯利狗类继承了狗类的特性。

以面向对象程序设计的观点，继承所表达的是类之间相关的关系，这种关系使得某一类可以继承另外一个类的特征和能力。例如，可以定义一个描述哺乳动物的类 Mammal，通过继承机制定义类 Dog，类 Dog 自动拥有类 Mammal 的所有特征和能力，并且在定义类

Dog 时，除了继承类 Mammal 的所有特征和能力外，还可添加新的特征和能力以区别于其他哺乳动物。这时，称被继承的类 Mammal 为基类或父类或超类，而称继承类 Dog 为 Mammal 的派生类或子类。同时也可以说，类 Dog 是从类 Mammal 中派生出来的。

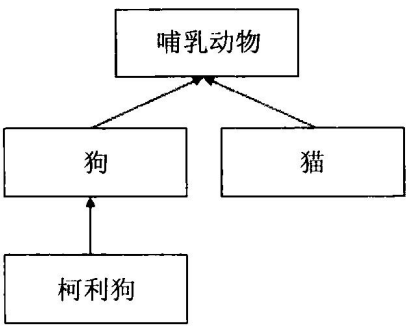


图 1-3　动物学中的继承性

图 1-3 中，狗类是从哺乳动物类派生出来的，而柯利狗类又是从狗类派生出来的，这就构成了类的层次。哺乳动物类是狗类的直接基类，是柯利狗类的间接基类，柯利狗不但继承它的直接基类的所有特性，还继承它的所有间接基类的特征。

面向对象程序设计为什么要提供继承机制？或者说继承有什么作用？继承的作用有两个：一是避免公用代码的重复开发，减少代码和数据冗余；二是通过增强一致性来减少模块间的接口和界面。

如果没有继承机制，那么每次软件开发都要从零开始，并且类的开发者在构造类时“各自为政”，使类与类之间没有什么联系，分别是一个个独立的实体。继承使程序不再是毫无关系的类的堆砌，而是具有良好的结构。

继承机制为程序员提供了组织、构造和重用类的一种手段。继承使得基类的属性和操作被派生类重用，在派生类中只需描述其基类中没有的属性和操作即可。这样就避免了公用代码的重复开发，增加了程序的可重用性，减少了代码和数据的冗余。

继承机制是面向对象方法的关键技术。这是因为类的继承性所构成的层次关系和人类认识客观世界的过程和方法吻合，从而使得人们能够用和认识客观世界一致的方法来设计软件。

④ 多态性

面向对象程序设计的另一个重要特性是多态性。所谓多态，是指一个名字有多种语义。下面我们考察多态性问题的一个类比问题。假设一辆汽车停在了属于别人的车位上，司机可能会听到这样的要求：请把车挪开。司机在听到请求后，所做的反应应该是把车开走；在家里，一把小椅子挡住了孩子的去路，她可能会请求妈妈：请把小椅子挪开，妈妈过去搬起小椅子，放到一边。在这两件事情中，司机和妈妈的工作都是挪开一样东西，但是他们在听到请求以后的行为是截然不同的。对于挪开这个请求，还可以有更多的行为与之对应。“挪开”从字面上看是相同的，但由于作用的对象不同，操作的方法也就不同。

与此类似，面向对象程序设计中的多态性是指不同的对象收到相同的消息时产生多种

不同的行为方式。例如，有一个窗口（Window）类对象，还有一个扑克牌（PlayingCard）类对象，现在对它们都发出“移动”的消息，“移动”操作在 Window 类对象和 PlayingCard 类对象上可以有不同的行为。

C++语言支持两种多态性，即编译时的多态性和运行时的多态性。编译时的多态性是通过重载来实现的，运行时的多态性是通过虚函数来实现的（详细内容见第 7 章和第 9 章）。多态性增强了软件的灵活性和重用性，为软件的开发和维护提供了极大的便利，尤其是采用了虚函数和动态联编机制后，允许用户以更为明确、易懂的方式去建立通用的软件。

（5）面向对象程序设计的优点

面向对象程序设计方法是软件开发史上的一个重要里程碑。这种方法从根本上改变了人们以往设计软件的思维方式，程序员将精力集中于要处理对象的设计和研究上，极大地减少了软件开发的复杂性，提高了软件开发的效率。面向对象程序设计主要具有以下优点。

① 真实的建模

因为我们生活在对象世界中，面向对象程序设计方法能更精确地模仿现实世界，符合人们习惯的思维方法，便于分解大型的、复杂多变的问题。

② 可提高程序的可重用性

重复使用一个类，可以方便地构造出软件系统，加上继承机制，极大地提高软件的开发效率。

③ 可改善程序的可维护性

用传统程序设计语言开发出来的软件很难维护，这是长期困扰人们的一个严重问题，是软件危机的突出表现。但面向对象程序设计方法所开发的软件可维护性较好。在面向对象程序设计中，对对象的操作只能通过消息传递来实现，所以只要消息模式即对应的方法界面不变，方法体的任何修改都不会导致发送消息的程序的修改，这显然为程序的维护带来了方便。此外，类的封装和信息隐藏机制使得外界对其中的数据和程序代码的非法操作成为不可能，这也就大大地减少了程序的错误率。

由于面向对象程序设计具有上述优点，它是目前解决软件开发面临难题的最有希望、最有前途的方法之一。

1.2　C++语言的产生和特点

1.2.1　C++语言的产生

FORTRAN 语言是世界上第一种计算机高级语言，诞生于 1954 年，其后出现了多种计算机高级语言，其中使用最广泛、影响最大的是 BASIC 语言和 C 语言。

BASIC 语言于 1964 年在 FORTRAN 语言的基础上简化而成，它是为初学者设计的小型高级语言。C 语言是一种高效的编译型结构化程序设计语言，于 1972 年由美国贝尔实验室的 Dennis Ritchie 研制成功。C 语言最初用作 UNIX 操作系统的描述语言，其功能强、性能好，能像汇编语言那样高效、灵活，又支持结构化程序设计。随着 UNIX 操作系统的广

泛应用，C 语言赢得了程序员们的青睐，到了 20 世纪 80 年代已经广为流行，成为一种应用广泛的程序设计语言。

但 C 语言也存在如下局限性。

（1）C 语言类型检查机制较弱，这使得程序中的一些错误不能在编译阶段被发现。

（2）C 语言本身几乎没有支持代码重用的机制，这使得各个程序的代码很难被其他程序所用。

（3）C 语言不适合大型项目的开发，当项目的规模达到一定程度时，程序员很难控制项目的复杂性。

为满足日益增长的软件开发需求，1980 年，美国贝尔实验室的 Bjarne Stroustrup 博士及其同事开始对 C 语言进行改编，在其基础上增加了面向对象的特性，开发出一种过程性与面对象性结合的程序设计语言。最初，他们把这种新的语言叫做“带类的 C”， 1983 年，这种语言被正式命名为 C++。

C++语言继承了 C 语言的原有精髓，如高效率、灵活性等，增加了对开发大型项目颇为有效的面向对象机制，弥补了 C 语言不支持代码重用、不适宜开发大型项目的不足，成为一种既可用于表现过程模型，又可用于表现对象模型的优秀的程序设计语言。

1.2.2 C++语言的特点

C++语言得到了越来越广泛的应用，它继承了 C 语言的优点，并拥有一些自己的特点，主要表现在以下几个方面。

（1）C++语言包含了 C 语言的全部特征、属性和优点。可以认为 C 语言是 C++语言的一个子集。C++语言全面兼容 C 语言，许多 C 语言代码不经修改就可以直接为 C++语言所用，用 C 语言编写的众多库函数和实用软件也可以自由地应用于 C++环境中。

（2）C++语言增加了面向对象的编程机制，可以方便地构造出模拟现实问题的实体和操作。C++语言和 C 语言的本质区别在于：C++语言是面向对象的，而 C 语言是面向过程的。

（3）用 C++语言编写的程序可读性更好，代码结构更为合理，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）节省开发时间和开发费用，且所开发软件的可重用性、可扩充性、可维护性和可靠性等性能有了很大的提高，使得大中型项目的开发设计变得更加容易。

目前，C++语言已经成为广泛使用的通用程序设计语言，国内外使用和研究 C++语言的人迅猛增加，优秀的 C++版本和配套的工具软件不断涌现。

1.3 C++程序中的类和对象

1.3.1 C++程序中的类

C++语言是在 C 语言的基础上扩充了面向对象机制而形成的一种面向对象的程序设计语言，支持类、对象、派生、继承和多态等概念和语言机制。下面给出一个简单的 C++示

例，以便读者对 C++中的类能有一个初步的了解。

例 1-1　一个简单的 C++示例。

```
#include <iostream.h>
//类的声明部分
class Car
{
private:
      char color[10];
public:
      void honk() //类 Car 中的成员函数
      {
            //语句
            cout<<"BEEP BEEP!";
      }
};
```

上述代码的说明如下：

❶ class 关键字用来声明一个类，大括号用来指明类体的开始和结束，分号用来结束类的声明。例如，

```
class Car
{
    …… ……
};
```

❷ class 关键字之后为类的名字。这里，Car 是类名。类名必须遵守一定的命名规则，并应尽量遵守命名惯例。

- 命名规则：类名必须由字母、数字和下划线（_）组成，且不能以数字开头；关键字（如 if、class 等）不可用作类名。
- 命名惯例：类名应该是有意义的；类名最好是名词；如果类名包含一个以上的单词且不用下划线的话，则类名中每个单词的第一个字母应采用大写。例如，描述职工家属的类名可以为 Employee Dependent。
- 命名规则是创建类时必须要遵守的，而命名惯例则不是强制性的，是开发人员在编码过程中应遵循的约定，或者说是初学者应努力养成的良好习惯。

❸ 在Car这个类中，color是其数据成员，用来描述Car的颜色属性。

❹ 对象通过传递消息和对消息的响应发生彼此交互，C++中传递消息的任务可通过成员函数来完成。函数是为响应消息而执行特定任务的一组语句。对象的函数称为成员函数，需要在类体内声明。例如，

```
class Car
{
    //成员函数
    void honk()
    {
```