高等学校计算机科学与技术应用型教材

C++
CHENGXU SHEJI

C++程序设计（第2版）

主　编◎邵兰洁　副主编◎马睿　李寰

北京邮电大学出版社
www.buptpress.com

高等学校计算机科学与技术应用型教材

C++程序设计

（第2版）

主　编　邵兰洁
副主编　马　睿　李　寰

北京邮电大学出版社
www. buptpress. com

内 容 简 介

本书以CDIO理念为指导，以项目驱动为主线，通过一个项目——学生信息管理系统的面向对象程序编制，全面而深入浅出地介绍了标准C++面向对象的程序设计技术。内容包括：C++对C语言的扩充、类与对象、继承与组合、多态性与虚函数、友元、静态成员、运算符重载、模板与STL、输入/输出流、异常处理、图形界面C++程序设计等。

本书内容丰富，通俗易懂，实用性强。它以一个综合性的案例贯穿始终，引导读者理解和领会面向对象程序设计的思想、技术、方法和要领。按照教材的引导一步步完成案例程序的编制，可以让读者在编程实践中提高自身的实践能力、自主学习能力、创新思维能力。

本书是按照应用型本科教学的基本要求编写的，既可用作高等院校计算机及相关专业本科生的面向对象程序设计课程教材，也适合用作具有C语言基础，想学习面向对象编程技术的自学者和广大程序设计人员的参考用书。

图书在版编目（CIP）数据

C++程序设计 / 邵兰洁主编. -- 2版. -- 北京：北京邮电大学出版社，2013.8
ISBN 978-7-5635-3575-0

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

中国版本图书馆CIP数据核字（2013）第165811号

书　　名：C++程序设计(第2版)
主　　编：邵兰洁
责任编辑：王丹丹
出版发行：北京邮电大学出版社
社　　址：北京市海淀区西土城路10号(邮编：100876)
发 行 部：电话：010-62282185　传真：010-62283578
E-mail：publish@bupt.edu.cn
经　　销：各地新华书店
印　　刷：北京联兴华印刷厂
开　　本：787 mm×1 092 mm　1/16
印　　张：18.75
字　　数：489千字
印　　数：1—3 000册
版　　次：2009年7月第1版　2013年8月第2版　2013年8月第1次印刷

ISBN 978-7-5635-3575-0　　定　价：38.00元

前　言

当今，面向对象编程技术是软件开发领域的主流技术，该技术从根本上改变了人们以往设计软件的思维方式，它把数据和对数据的操作封装起来，集抽象性、封装性、继承性和多态性于一体，可以帮助人们开发出高可靠性、可复用、易修改、易扩充的软件，极大地降低了软件开发的复杂度，提高了软件开发的效率，尤其适用于功能庞大而复杂的软件开发。C++为面向对象编程技术提供全面支持，是主流的面向对象程序设计语言，在当前软件开发领域占据重要地位。全国各地高校计算机及相关专业基本上都开设了该课程，目的是让学生掌握面向对象程序设计的基本知识和基本技能，学会利用C++语言进行面向对象程序的编写，解决一般应用问题，并为后续专业课程的学习奠定程序设计基础。

C++由C语言发展而来，它在C语言的基础上进行功能扩充，增加了面向对象的机制。无论从编程思想、代码效率、程序的可移植性和可靠性，还是从语言基础、语言本身的实用性来讲，C++都是面向对象程序设计语言的典范。学好C++，不仅能够用于实际的程序设计，而且有助于理解面向对象程序设计的精髓，再来学习诸如Java、C#之类的面向对象程序设计语言也就简单了。

但是，目前的大多数C++教材在内容安排上都是既介绍C++的面向过程程序设计（这里绝大部分是在介绍原来C语言的内容），又介绍C++的面向对象程序设计。这样的教材对于没有C语言基础的读者来说是合适的。可是目前有不少高校是把C语言和C++分别作为独立的两门课，尤其对计算机科学与技术专业、软件工程专业的学生来说，这样的安排更合理些。所以需要以C语言为起点的C++教材，这样可以节省教学时间。本书就是应这种需要而产生的。本书的特点如下：

（1）重点突出，内容取舍合理。本书重点讲解C++的面向对象程序设计，同时还介绍了C++在面向过程方面对C语言的扩充。

（2）通俗易懂、深入浅出。本书力求用通俗易懂的语言、生活中的现象来阐述面向对象的抽象的概念，以减少初学者学习C++的困难，深入浅出，便于自学。

（3）强调示例程序的可读性和标准化。本书的所有示例程序均遵循程序员所应该遵循的一般编程风格，如变量名、函数名和类名的命名做到“见名知义”，采用缩排格式组织程序代码并配以尽可能多的注释等，程序可读性强。同时每个示例程序均在Visual C++6.0和Visual Studio 2012下调试通过，并给出运行结果。所有示例程序均按照标准C++编写，力求培养学生从一开始就写标准C++程序的习惯。

（4）强调示例程序的实用性。本书示例程序都是经过精心设计的，实用性强，力求解决理论与实际应用脱离的矛盾，从而达到学有所用的目的。

（5）重视学生实际编程能力的培养。本书以CDIO工程教育模式所倡导的“基于项目

的学习”理念为指导，精心设计了一个贯穿全书大部分章节的项目——学生信息管理系统。随着学习进程的推进，不断地运用所学的面向对象的C++程序设计技术完成、完善该系统的功能，最后形成一个比较完整的系统。学生可借此理解面向对象的编程思想，掌握面向对象程序设计的方法，提高自身的实践能力、自主学习能力、创新思维能力。

(6) 特别关注内容提醒。凡是需要学生特别关注的内容，书中都用带阴影的文字标记，以引起学生的注意。

(7) 提供配套的上机指导与习题解答。配套的上机指导可以为课程上机提供方便，习题解答方便读者自查。

全书共分11章。第1章为面向对象程序设计概述，本章从学生信息管理系统项目的面向过程程序设计出发，讨论了传统的面向过程程序设计方法的不足，进而引出面向对象程序设计方法，介绍面向对象程序设计的编程思想、面向对象程序设计的基本概念、面向对象程序设计的优点。第2章为C++对C语言的扩充，主要介绍C++在面向过程方面对C语言功能的扩充。第3～10章介绍C++的面向对象程序设计，包括类与对象、继承与组合、多态性与虚函数、友元、静态成员、运算符重载、模板与STL、输入/输出流和异常处理等内容。第11章为图形界面C++程序设计，演示基于对话框和基于单文档图形界面C++程序的设计步骤，让读者体验图形界面C++程序的开发过程，消除开发窗口程序的神秘感。

本书第1、4、8、9、11章和项目案例代码由邵兰洁编写，第2、3章由李寰编写，第5、7章由马睿编写，第6章由史迎春编写，第10章由母俐丽编写。全书由邵兰洁统稿、审稿。

在本书编写过程中阅读参考了国内外大量的C++书籍，这些书籍已被列在书后的参考文献中，在此谨向这些书籍的作者表示衷心的感谢。

为方便读者学习和教师教学，本书配有以下辅助资源：

※ 例题的全部程序代码；

※ 配套的PPT电子课件；

※ 配套的上机指导与习题解答；

※ 全部习题程序代码。

除配套的《C++程序设计上机指导与习题解答》外，其他资源均可从北京邮电大学出版社的网站(www.buptpress.com)上进行下载或发邮件到shaolanjie@126.com向编者索取。

由于编者水平有限，书中难免存在疏漏和不足之处，恳请读者批评指正。

编　者

目　录

第1章 面向对象程序设计概述

面向对象程序设计与面向过程程序设计有着本质的区别。面向过程程序设计以功能为中心，数据和操作数据的函数相分离，程序的基本构成单位是函数。而面向对象程序设计以数据为中心，数据和操作数据的函数被封装成一个对象，与外界相对分隔，对象之间通过消息进行通信，使各对象完成相应的操作，程序的基本构成单位是对象。

本章从学生信息管理系统项目的面向过程程序设计出发，讨论了传统的面向过程程序设计方法的不足，进而引出面向对象程序设计方法，介绍面向对象程序设计的编程思想、基本概念及面向对象程序设计的优点。最后简单介绍面向对象的软件开发。

1.1 面向过程程序设计

面向过程程序设计的基本思想：功能分解、逐步求精、模块化、结构化。当要设计一个目标系统时，首先从整体上概括出整个系统需要实现的功能，然后对系统的每项功能进行逐层分解，直到每项子功能都足够简单，不需要再分解为止。具体实现系统时，每项子功能对应一个模块[①]，模块间尽量相对独立，通过模块间的调用关系或全局变量而有机地联系起来。下面举例说明面向过程程序设计方法的应用。

【例 1-1】 运用面向过程程序设计方法设计一个学生信息管理系统。该系统要管理的学生信息包括：学号(Num)、姓名(Name)、性别(Sex)、出生日期(Birthday)、三门课成绩〔英语(English)、数据结构(DataStructure)、C++程序设计(CPlusPlus)〕、总成绩(Sum)、平均成绩(Average)，学生信息表如表 1-1 所示。该系统要具有如下功能。

(1) 显示学生信息：显示全部学生的信息。

(2) 查询学生信息：查询指定学生的信息(指定学号或姓名)，查询结果直接显示在屏幕上。

(3) 添加学生信息：对学生信息进行添加。

(4) 修改学生信息：按指定学号对学生信息进行修改。

(5) 删除学生信息：按指定学号对学生信息进行删除。

(6) 统计学生成绩：统计每个学生的总成绩和平均成绩，或统计所有学生某一门课的总成绩和平均成绩。

---

① 一个模块就是一个程序段，是能够实现某一功能，可以独立地进行编制、测试和维护的程序单位。

（7）学生信息排序：按学号、总成绩或某一门课成绩排序。

（8）备份学生信息：把所有学生信息备份一份。

表 1-1　学生信息表

学号	姓名	性别	出生日期	英语成绩	数据结构成绩	C++成绩	总成绩	平均成绩
20120101001	邓光辉	男	1994-02-05	87	88	90	265	88.3
20120101002	杜丽丽	女	1995-09-20	79	80	75	234	78.0
20120101003	姜志远	男	1995-11-08	68	84	70	222	74.0
20120101004	张大伟	男	1993-08-05	70	67	82	219	73.0
…	…	…	…	…	…	…	…	…

从例 1-1 来看，该学生信息管理系统具有 8 项子功能，其中需要进行功能分解的有(2)、(6)、(7)。对于“(2)查询学生信息”子功能，可以继续分解为按学号查询、按姓名查询两项子功能，这两项子功能已足够简单，不需要再继续分解。对于“(6)统计学生成绩”子功能，可以继续分解为统计每个学生的总成绩和平均成绩、统计所有学生某一门课的总成绩和平均成绩两项子功能，这两项子功能已足够简单，也不需要再继续分解。对于“(7)学生信息排序”子功能，可以继续分解为按学号排序、按总成绩排序和按某一门课成绩排序 3 项子功能，这 3 项子功能也不需要再继续分解。

我们已经学习过的 C 语言是一种支持面向过程程序设计的计算机语言，这里选择 C 语言实现该系统。将该系统的每一项子功能对应设计成一个 C 函数，各个函数及函数间的调用组成程序。

假定该系统要管理的学生信息在外存中以文件的形式存储，文件名为 students.dat；在内存中以顺序表的形式存储，存放在一个一维数组 stud 中。把存放在外存中的学生信息读入到内存用一个自定义函数 ReadData 来完成。实现该系统的 C 程序框架如下：

```
/*学生信息管理系统C语言源代码 student.c*/
#include <stdio.h>                          /*包含输入/输出头文件*/
#include <string.h>                         /*包含字符串处理头文件*/
#include <stdlib.h>                         /*包含定义C语言标准库函数的头文件*/
#define MAXSIZE 100                         /* 能够处理的最多学生人数*/
typedef struct {                            /*用于存放学生生日信息的结构体*/
    int year;
    int month;
    int day;
}Date;
typedef struct {                            /*用于存放学生信息的结构体*/
    char Num[12];                           /*学号,由11位数字组成*/
    char Name[11];                          /*姓名,最多可包含5个汉字*/
    char Sex;                               /*性别,M代表男性,F代表女性*/
    Date Birthday;                          /*出生日期*/
    float English, DataStructure, CPlusPlus;  /*三门课成绩*/
    float Sum, Average;                     /*总成绩、平均成绩*/
}Student;
/*全局变量定义*/
```

```
Student stud[MAXSIZE];                    /*用于存放读入内存中的所有学生信息的全局数组*/
int count = 0;                            /*用于存放实际学生人数的全局变量*/
…(其他全局变量定义略)
/*各自定义函数原型声明*/
void ReadData( );                 /*读入学生信息到全局数组 stud 中*/
void WriteData( );        /*把全局数组 stud 中的学生信息写入外存学生信息文件 student.dat 中*/
void Display( );                  /*显示学生信息*/
void Search( );                   /*查询学生信息*/
int SearchNum(char *num);         /*按学号查询学生信息*/
int SearchName(char *name);       /*按姓名查询学生信息*/
…(其他自定义函数原型声明略)
void Backup( );                   /*备份学生信息*/
void main( )                      /*main 函数*/
{    /*系统功能以菜单的形式提供给用户*/
  int choice;   /* choice 变量用于保存用户对功能菜单的选择结果*/
  ReadData( );  /*程序开始运行时,首先把存放在外存中的学生信息读入内存*/
  for( ; ; )
  {  /*显示系统功能菜单*/
     printf("**************学生信息管理系统******************\n");
     printf("*************************************************\n");
     printf("************* 1.显示学生信息 ******************\n");
     printf("************* 2.查询学生信息 ******************\n");
     …
     printf("************* 8.备份学生信息 ******************\n");
     printf("************* 0.退出系统     ******************\n");
     printf("*************************************************\n");
     printf("            请选择要执行的操作(0~8):");
     scanf("%d", &choice);
     switch(choice){
     case 1: Display( ); break;
     case 2: Search( ); break;
     …(其他 case 语句略)
     case 8: Backup( ); break;
     case 0: /*退出系统*/
          printf("程序执行完毕,按任意键退出…\n");
          WriteData( );          /*程序执行完毕时,保存内存的学生信息到外存文件*/
          getch( );
          exit(0);
     default: /*用户选择错误时的处理*/
          printf("选项错误,无此功能,请重新选择\n");
          getch( );/*让屏幕暂停,以观察结果*/
          break;
     }/*switch 结束*/
     while( getchar( ) != '\n' ) ; /*为了避免下次输入出错,需要清除键盘缓冲区*/
     printf("按任意键继续…\n");
     getch( ); /*让屏幕暂停,以观察结果*/
     system( "cls" ); /*调用 DOS 命令 cls 来清除屏幕*/
  }/*for 结束*/
```

```
}/*main 函数结束*/
/*各自定义函数实现代码*/
void ReadData( ){  …(代码略)    }
void WriteData( ){  …(代码略)    }
void Display( ){  …(代码略)    }
void Search( ){  …(代码略)    }
…(其他自定义函数实现代码略)
```

从上述学生信息管理系统的C程序框架可以看出，运用面向过程程序设计方法所设计出来的程序，数据和操作数据的函数是分离的。所有数据都是公用的，一个函数可以使用任何一组数据，而一组数据又能被多个函数所使用。用面向过程程序设计方法所设计出来的程序模型如图1-1所示。

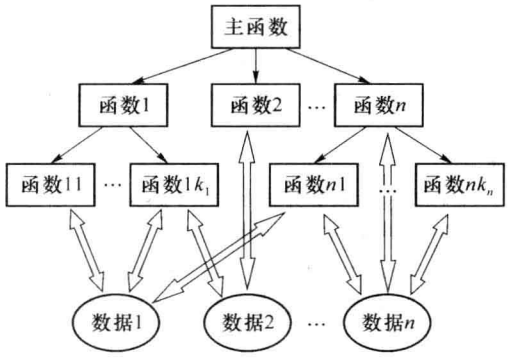

图1-1　面向过程程序设计的程序模型

面向过程的结构化程序设计在20世纪60年代末70年代初从一定程度上缓解了当时的“软件危机”，它在处理较小规模的程序时比较有效。但是，随着人们对大规模软件需求的增长，面向过程的结构化程序设计逐渐显示出它的不足，具体表现在以下几方面。

1. 程序设计困难，生产率低下

面向过程的程序设计是围绕功能进行的，用一个函数实现一项功能。所有数据都是公用的，一个函数可以使用任何一组数据，而一组数据又能被多个函数所使用(见图1-1)。程序设计者必须仔细考虑每一个细节，在什么时候需要对什么数据进行操作。当程序规模较大、数据很多、操作种类繁多时，程序设计者往往感到难以应付。就如工厂的厂长直接指挥每一个工人的工作一样，一会儿让某车间的某工人在A机器上用X材料生产轴承，一会儿又让另一车间的某工人在B机器上用Y材料生产滚珠……显然这是非常劳累的，而且往往会遗漏或搞错，所以面向过程程序设计只适用于规模较小的程序。

2. 数据不安全

在面向过程的程序中，所有数据都是公用的，谁也没有办法限制其他程序员不去修改全局数据，也不能限制其他程序员在函数中定义与全局数据同名的局部变量。因为面向过程的程序设计语言并没有提供这样一种数据保护机制。当程序规模较大时，这个问题尤其突出。

3. 程序修改困难

当某个全局数据的数据结构修改时，所有操作该全局数据的函数都要进行修改。特别是当程序的功能因用户需求的变化而改变时，程序修改的难度更大，很有可能会导致程序的重新设计。

4. 代码重用程度低

运用面向过程程序设计方法所设计出来的程序，其基本构成单位为函数，故代码重用的力度最大也只能到函数级。对于今天的软件开发来说，这样的重用力度显得非常不够。

针对面向过程程序设计的不足，人们提出了面向对象程序设计方法。

1.2 面向对象程序设计

面向对象程序设计思想的出发点是思考面向过程的程序设计为什么不能有效地解决大规模程序设计的问题？实际上，面向过程的程序设计是从计算机的角度出发去解决问题，换句话说，就是用计算机的观点去观察世界。计算机的观点和人类的思维方式是有很大区别的，从计算机的角度出发，根据对系统的分析，将系统按功能划分为一个一个的模块。在这个过程中，很多现实世界里的整体被分割成了若干个部分，这样就使得问题在从现实世界到计算机世界的转换过程中出现一定的差距。当问题较小或简单时，这种差距可以很容易地通过程序进行弥补。但当问题规模变大时，这种差距就难以弥补了。

面向对象方法的出发点是尽可能地模拟人类的思维方式去描述现实世界中的问题，使软件开发的方法尽可能接近人类认识、解决现实世界中问题的方法，使得问题在从现实世界向计算机世界转换过程中的差距尽可能地小，也就是让描述问题的问题空间和实现解法的解空间在结构上尽可能地一致。面向对象方法已经在当今的软件开发中占据了主流的位置。下面简单介绍面向对象的编程思想。

1.2.1 面向对象的编程思想

具体地讲，面向对象编程的基本思想如下。

(1) 客观世界中的事物都是对象(object)，对象之间存在一定的关系。

面向对象方法要求从现实世界客观存在的事物出发来建立软件系统，强调直接以问题域(现实世界)中的事物为中心来思考问题和认识问题，并根据这些事物的本质特征和系统责任，把它们抽象地表示为系统中的对象，作为系统的基本构成单位。这可以使系统直接映射到问题域，保持问题域中的事物及其相互关系的本来面目。

(2) 用对象的属性(attribute)描述事物的静态特征，用对象的操作(operation)描述事物的行为(动态特征)。

(3) 对象的属性和操作结合为一体，形成一个相对独立、不可分的实体。对象对外屏蔽其内部细节，只留下少量接口，以便与外界联系。

(4) 通过抽象对对象进行分类，把具有相同属性和相同操作的对象归为一类，类是这些对象的抽象描述，每个对象是其所属类的一个实例。

(5) 复杂的对象可以用简单的对象作为其构成部分。

(6) 通过在不同程度上运用抽象的原则，可以得到一般类和特殊类。特殊类继承一般类的属性与操作，从而简化系统的构造过程。

(7) 对象之间通过传递消息进行通信，以实现对象之间的动态联系。

(8) 通过关联表达类之间的静态关系。

为了让大家对面向对象的编程思想有更深入地理解，下面先对面向对象的基本概念进行阐述。

1.2.2 面向对象的基本概念

1. 对象

可以从两个角度来理解对象。一个角度是现实世界，另一个角度是我们所建立的软件系统。

现实世界中客观存在的任何一个事物都可以看成一个对象。或者说，现实世界是由千千万万个对象组成的。对象可以是有形的，如汽车、房屋、张三等；也可以是无形的，如社会生活中的一种逻辑结构，如学校、军队，甚至一篇文章、一个图形、一项计划等都可视作对象。

对象可大可小。如学校是一个对象，一个班级、一个学生也是一个对象。同样，军队中的一个师、一个团、一个连、一个班等都是对象。

任何一个对象都具有两个要素：属性和行为，属性用于描述客观事物的静态特征，行为用于描述客观事物的动态特征。例如，一个人是一个对象，他有姓名、性别、身高、体重等属性，有走路、讲话、打手势、学习和工作等行为。一台录像机是一个对象，它有生产厂家、牌子、颜色、重量、价格等属性，有录像、播放、快进、倒退、暂停、停止等行为。一般来说，凡是具有属性和行为这两个要素的，都可以作为对象。

在一个系统中的多个对象之间通过一定的渠道相互联系，如图 1-2 所示。要使某一个对象实现某一个行为，应当向它传递相应的消息。如想让录像机开始播放，必须由人去按录像机的按键，或者用遥控器向录像机发一个信号。对象之间就是这样通过发送和接收消息互相联系的。

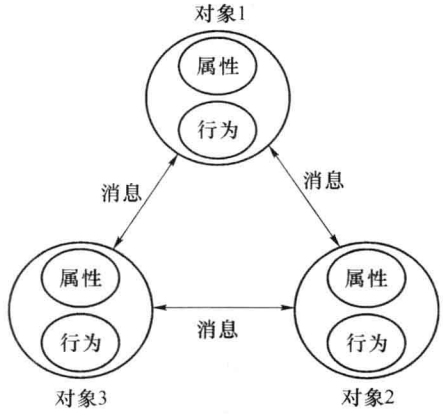

图 1-2 现实世界系统对象模型

在面向对象的软件系统中，对象是用来描述客观事物的一个相对独立体，是构成系统的一个基本单位。一个对象由一组属性和对这组属性进行操纵的一组操作组成。属性是用来描述对象静态特征的一个数据项，操作是用来描述对象行为的一个动作序列。

在开发软件系统时，首先要对现实世界中的对象进行分析和归纳，以此为基础来定义软件系统中的对象。

软件系统中的一部分对象是对现实世界中的对象的抽象，但其内容不是全盘照搬，这些对象只包含与所解决的现实问题有关的那些内容；系统中的另一部分对象是为了构建系统而设立的。

2. 类

类是对客观世界中具有相同属性和行为的一组对象的抽象，它为属于该类的全部对象提供了统一的抽象描述，其内容包括属性和操作。

在寻找类时，要用到一个概念：抽象。所谓抽象，是指忽略事物的非本质特征，只注意那些与当前目标有关的本质特征，从而找出事物的共性，把具有共性的事物划分为一类，得出一个抽象的概念。例如，人可以作为一个类，它是世界上所有实体人如张三、李四、王五等的抽象，而实体人张三、李四、王五等则是人这个类的具体实例。

类和对象的关系可表述为：类是对象的抽象，而对象则是类的实例，或者说是类的具体表现形式。

3. 封装

日常生活中，运用封装原理的例子很多，如录像机、VCD播放器、数码相机、手机等。就拿录像机来说，录像机里有电路板和机械控制部件，但在外面是看不到的，从外面看它只是一个“黑盒子”，在它的表面有几个按键，这就是录像机与外界的接口，人们在使用录像机时不必了解它的内部结构和工作原理，只需知道按哪一个键能执行哪种操作即可。

这样做的好处是大大降低了人们操作对象的复杂程度，使用对象的人完全可以不必知道对象内部的具体细节，只需了解其外部功能即可自如地操作对象。

在面向对象方法中，所谓“封装”是指两方面的含义：一是用对象把属性和操纵这些属性的操作包装起来，形成一个基本单位，各个对象之间相对独立，互不干扰；二是将对象中某些部分对外隐蔽，即隐藏其内部细节，只留下少量接口，以便与外界联系，接收外界的消息。这种对外界隐蔽的做法称为信息隐蔽(information hiding)。信息隐蔽还有利于数据安全，防止无关的人了解和修改数据。

4. 继承

所谓“继承”，是指特殊类自动地拥有或隐含地复制其一般类的全部属性与操作。继承具有“是一种”的含义，在图1-3中，卡车是一种汽车，轿车是一种汽车，二者作为特殊类继承了一般类“汽车”类的全部属性和操作。

在类的继承层次结构中，位于上层的类叫做一般类(也称为基类或父类)，而位于下层的类叫做特殊类(也称为派生类或子类)。

通过在不同程度上运用抽象原则，可以得到较一般的类和较特殊的类。在图1-4中，从上向下看是对运输工具的分类，而从下向上看是经过了3个层次的抽象。从该图可以看出，继承具有传递性，例如，轿车具有运输工具的全部内容。

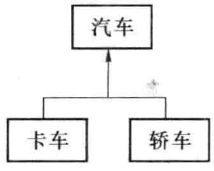

图1-3 继承示例

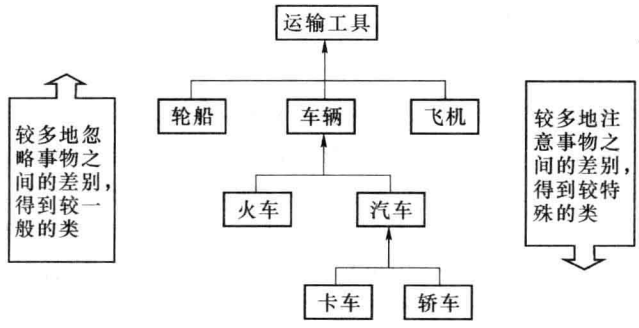

图1-4 继承的层次与抽象原则的运用

有时一个类要同时继承两个或两个以上一般类中的属性和操作，把这种允许一个特殊类具有一个以上一般类的继承模式称作多继承。图1-5给出了一个多继承示例。