



21世纪高校计算机应用技术系列规划教材

谭浩强 主编

C++面向对象程序设计

陈维兴 林小茶 编著

★系统介绍C++面向对象程序设计的基本概念、

基本语法和编程方法。

★书中提供了大量实例、

习题和上机实验题，

使读者能深刻理解、

领会面向对象程序设计的特点和风格，

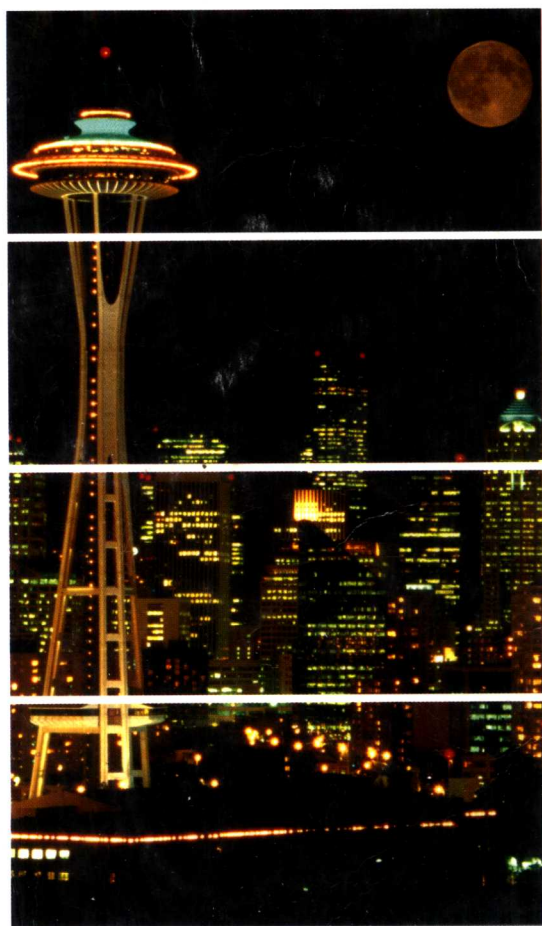
掌握其方法和要领。

★本书以应用为目的，

注重培养应用能力。

★本书适合于高校学生学习C++程序设计的教材，

也可作为C++语言自学者的参考书。



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



谭浩强 主编

21 世纪高校计算机应用技术系列规划教材

C++面向对象程序设计

陈维兴 林小茶 编著

中国铁道出版社

2005 · 北京

内 容 简 介

本书全面系统地介绍了 C++ 面向对象程序设计的基本概念、基本语法和编程方法。较详尽地讲述了 C++ 语言面向对象的基本特性, 包括类、对象、派生类、继承、多态性、虚函数、模板、流类库等。全书提供了大量实例、习题和上机实验题, 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。

本书以应用为目的, 注重培养应用能力。本书可以作为大学本科应用型专业的学生以及高职高专学生学习 C++ 程序设计课程的教材, 也可作为 C++ 语言自学者的教材或参考书。

图书在版编目 (CIP) 数据

C++ 面向对象程序设计/陈维兴, 林小茶编著. —北京: 中国铁道出版社, 2004. 3

(21 世纪高校计算机应用技术系列规划教材)

ISBN 7-113-05770-5

I. C… II. ①陈…②林… III. C 语言-程序设计-高等学校-教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 014883 号

书 名: C++ 面向对象程序设计

作 者: 陈维兴 林小茶

出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 魏 春

责任编辑: 苏 茜 黄园园 秦绪好

封面设计: 白 雪

印 刷: 遵化市胶印厂印刷

开 本: 787×1092 1/16 印张: 22.5 字数: 539 千

版 本: 2004 年 4 月第 1 版 2005 年 2 月第 2 次印刷

印 数: 5001~8000 册

书 号: ISBN 7-113-05770-5/TP·1144

定 价: 29.00 元

版权所有 侵权必究

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

21 世纪高校计算机应用技术系列规划教材

编委会名单

主 任： 谭浩强

副主任： 陈维兴 严晓舟

委 员： （以下排名按姓氏字母的先后顺序为序）

安淑芝	安志远	侯冬梅	李雁翎	秦建中
秦绪好	宋 红	宋金珂	孙中胜	魏 春
魏善沛	熊伟建	薛淑斌	赵乃真	訾秀玲

丛书序言

21 世纪是信息技术高度发展并且得到广泛应用的时代,信息技术深刻地改变了人类的生活、工作和思维方式。每一个人都应当学习信息技术、应用信息技术。人们平常习惯说的计算机教育其内涵实际上已经发展为信息技术教育,内容主要包括计算机和网络的基本知识和应用。

对多数人来说,学习计算机的目的是为了利用计算机这个现代化工具去处理工作和面临的各种问题,使自己能够跟上时代前进的步伐,同时要在学习的过程中努力培养自己的信息素养,使自己具有信息时代所要求的科学素质,站在信息技术发展和应用的前列,推动我国信息技术的发展。

学习计算机课程,有两种不同的方法,一是从理论入手;一是从实际应用入手。不同的人有不同的学习内容和学习方法。大学生中的多数人将来是各行各业中的计算机应用人才。对他们来说,不仅需要解决**知道什么**,更重要的是**会做什么**。因此要以应用为目的,注重培养应用能力,大力加强实践环节,激励创新意识。

根据实际教学的需要,我们组织编写这套“**21 世纪高校计算机应用技术系列规划教材**”。顾名思义,这套丛书的特点是突出应用技术,面向实际应用。在选材上,根据实际应用的需要决定内容的取舍,坚决舍弃那些现在用不到、将来也用不到的内容。在叙述方法上,采取“**提出问题——介绍解决问题的方法——归纳结论和概念**”的三部曲,这种从实际到理论、从具体到抽象、从个别到一般的方法,符合人们的认识规律,实践证明已取得了很好的效果。

本丛书采取模块化的结构,根据需要确定一批书目,也就是提供一个课程菜单供各校选用,以后根据信息技术的发展和教学的需要,不断地补充和调整。只要教学有需要,我们就组织编写新的教材,不受任何框框的限制。我们的指导思想是面向实际,面向应用,面向对象。这样比较灵活,能满足不同学校、不同专业的需要。希望各校的老师把你们的要求反映给我们,我们将会尽最大努力满足大家的要求。

本丛书可以作为大学计算机应用技术课程教材以及高职高专、成人高校和面向社会的培训班的教材,也可作为学习计算机的自学教材。

参加本丛书策划和编写工作的专家和老师有:谭浩强、陈维兴、严晓舟、薛淑斌、秦建中、安淑芝、安志远、赵乃真、李雁翎、宋红、周永恒、熊伟建、宋金珂、陈元春、冯继生、姚怡、沈洪、沈添、李尊朝、王晓敏、侯冬梅、訾秀玲、魏善沛、孙中胜、王丙义、程爱民、史秀璋、李振银、刘涛、李宁等。此外参加本丛书编辑和其他工作的还有:魏春、秦绪好、张艳芳、戴薇、郭晓溪、马建、姜淑静、杨东晓、于静等。对于他们的智慧、奉献和劳动表示深切的谢意。中国铁道出版社以很高的热情和效率组织了丛书的出版工作。在组织编写出版的过程中,得到全国高等院校计算机基础教育研究会和各高等院校老师的热情鼓励和支持,对此谨表衷心的感谢。

本丛书如有不足之处,请各位专家、老师和广大读者不吝指正。

谭浩强谨识

2003 年 2 月于清华园

前 言

面向对象程序设计是不同于传统程序设计的一种新的程序设计范型。它对降低软件的复杂性,改善其重用性和维护性、提高软件的生产效率,有着十分重要的意义。因此面向对象的程序设计被普遍认为是程序设计方法学的一场实质性的革命。

C++语言是在C语言基础上扩充了面向对象机制而形成的一种面向对象程序设计语言,它除了继承了C语言的全部优点和功能外,还支持面向对象程序设计。C++现在已成为介绍面向对象程序设计的首选语言。学习C++不仅可以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,而且可以使读者掌握一种十分流行和实用的程序设计语言。

近年来许多高等院校纷纷将面向对象程序设计及面向对象技术正式列入教学计划,作为必修课或选修课,许多有识之士也纷纷把目光转向面向对象程序设计。

鉴于以上情况,我们在多年教学和科研的基础上编写了这本教材,旨在使读者迅速迈入面向对象程序设计的大门,掌握C++程序设计的基本技能和面向对象的概念与方法,并能编写出具有良好风格的程序。本书的特点是:通俗易懂,适于自学;由浅入深,便于理解;例题丰富,重点突出。

本教材共分10章,第1章概述了面向对象程序设计的基本概念。第2章介绍了C++对C语言在非面向对象方面的扩充。第3~9章详述了C++支持面向对象程序设计的基本方法,包括类、对象、派生类、继承、多态性、模板、流类库等。第10章介绍了面向对象程序设计的一般方法和技巧,并安排了两个应用实例,供读者借鉴。在附录中给出了上机实验题,供读者上机练习。带有“*”号的章节是提高部分,读者可根据自己的需要,予以有选择地学习。

作为本书的姐妹篇,我们将同时出版本教程的习题解答和实验指导书,给出了教程中所有习题的参考答案及每个实验题的参考程序,供教师和学生授课与学习时参考。

本教程第1~9章由陈维兴教授编写,第10章和附录以及习题由林小茶副教授编写。全书由陈维兴组织编写并统稿。

在本书编写和出版过程中,全国高等院校计算机基础教育研究会会长谭浩强教授给予了指导和把关,在此表示衷心的感谢。

在本书的编写和出版过程中还得到了郑玉明教授、陈宝福教授、杨道沅教授、陈昕博士以及胡燕萍老师的帮助和支持,在此表示诚挚的感谢。

由于编者水平有限,书中难免还存在一些缺点和错误,殷切希望广大读者批评指正。

编 者
2004年3月

目 录

第 1 章 面向对象程序设计概述.....	1
1-1 什么是面向对象程序设计.....	2
1-1-1 新的程序设计范型.....	2
1-1-2 面向对象程序设计的基本概念.....	2
1-1-3 面向对象程序设计的基本特征.....	5
1-2 为什么要使用面向对象程序设计.....	9
1-2-1 传统程序设计方法的局限性.....	9
1-2-2 面向对象程序设计的主要优点.....	10
1-3 面向对象程序设计的语言.....	12
1-3-1 面向对象程序设计语言的发展概况.....	12
1-3-2 几种典型的面向对象程序设计语言.....	13
1-4 本章小结.....	13
习题.....	14
第 2 章 C++基础.....	15
2-1 C++的产生和特点.....	16
2-1-1 C++的产生.....	16
2-1-2 C++的特点.....	16
2-2 C++程序的结构特性.....	17
2-2-1 一个简单的 C++示例程序.....	17
2-2-2 C++程序的结构特性.....	18
2-2-3 C++程序的编辑、编译和运行.....	19
2-3 C++在非面向对象方面的扩充.....	19
2-3-1 注释与续行.....	19
2-3-2 输入输出流.....	20
2-3-3 灵活的变量说明.....	21
2-3-4 结构、联合和枚举名.....	22
2-3-5 函数原型.....	22
2-3-6 const 修饰符.....	24
2-3-7 void 型指针.....	26
2-3-8 内联函数.....	27
2-3-9 带有缺省参数值的函数.....	29
2-3-10 函数重载.....	30
2-3-11 作用域标识符::.....	32



C++ 面向对象程序设计

2-3-12 无名联合.....	33
2-3-13 强制类型转换.....	33
2-3-14 new 和 delete	33
2-3-15 引用	36
2-4 本章小结	41
习题	42
第3章 类和对象（一）	45
3-1 类的构成	46
3-1-1 从结构到类.....	46
3-1-2 类的构成.....	47
3-2 成员函数的声明.....	48
3-3 对象的定义和使用.....	50
3-3-1 类与对象的关系.....	50
3-3-2 对象的定义.....	51
3-3-3 对象中成员的访问.....	52
3-3-4 类成员的访问属性.....	54
3-3-5 对象赋值语句.....	55
3-3-6 类的作用域.....	56
3-4 构造函数与析构函数.....	57
3-4-1 构造函数.....	57
3-4-2 成员初始化表.....	60
3-4-3 缺省参数的构造函数.....	62
3-4-4 析构函数.....	63
3-4-5 缺省的构造函数和缺省的析构函数	66
3-4-6 重载构造函数.....	69
3-4-7 拷贝构造函数.....	71
3-4-8 浅拷贝和深拷贝.....	77
* 3-5 对象的生存期	80
3-6 本章小结	83
习题	83
第4章 类和对象（二）	87
4-1 自引用指针 this.....	88
4-2 对象数组与对象指针.....	90
4-2-1 对象数组.....	90
4-2-2 对象指针.....	93
* 4-2-3 指向类的成员的指针	95
4-3 向函数传递对象.....	97
4-3-1 使用对象作为函数参数.....	97

4-3-2	使用对象指针作为函数参数	98
4-3-3	使用对象引用作为函数参数	99
4-4	静态成员	100
4-4-1	静态数据成员	100
4-4-2	静态成员函数	106
4-4-3	通过普通指针访问静态成员	109
4-5	友元	110
4-5-1	友元函数	110
4-5-2	友元成员	113
4-5-3	友元类	114
4-6	对象成员	116
4-7	常类型	121
4-7-1	常引用	121
4-7-2	常对象	121
4-7-3	常对象成员	123
4-8	C++程序的多文件组成	125
4-9	程序举例	126
4-10	本章小结	132
	习题	133
第 5 章	继承与派生类	139
5-1	继承与派生类	140
5-1-1	为什么要使用继承	140
5-1-2	派生类的声明	141
5-1-3	基类成员在派生类中的访问属性	142
5-1-4	派生类对基类成员的访问规则	143
5-2	派生类的构造函数和析构函数	150
5-2-1	派生类构造函数和析构函数的执行顺序	150
5-2-2	派生类构造函数和析构函数的构造规则	151
* 5-3	调整基类成员在派生类中的访问属性的其他方法	156
5-3-1	同名成员	156
5-3-2	访问声明	158
5-4	多继承	161
5-4-1	多继承的声明	161
5-4-2	多继承的构造函数与析构函数	163
5-4-3	虚基类	166
5-5	赋值兼容规则	171
5-6	程序举例	174
5-7	本章小结	179



习题	180
第 6 章 多态性与虚函数	187
6-1 多态性概述	188
6-1-1 多态的分类	188
6-1-2 多态的实现	188
6-2 虚函数	189
6-2-1 虚函数的引入	189
6-2-2 虚函数的作用和定义	190
* 6-2-3 虚析构造函数	194
6-2-4 虚函数与重载函数的关系	195
6-2-5 多继承与虚函数	196
6-2-6 虚函数举例	197
6-3 纯虚函数和抽象类	198
6-3-1 纯虚函数	198
6-3-2 抽象类	200
6-4 程序举例	200
6-5 本章小结	206
习题	206
第 7 章 运算符重载	209
7-1 运算符重载概述	210
7-1-1 运算符重载的好处	210
7-1-2 运算符重载的规则	212
7-2 运算符重载函数的两种形式	213
7-2-1 友元运算符函数	213
7-2-2 成员运算符函数	220
7-2-3 成员运算符函数与友元运算符函数的比较	223
7-3 几个常用运算符的重载	225
7-3-1 单目运算符“++”和“--”的重载	225
7-3-2 赋值运算符“=”的重载	229
* 7-3-3 下标运算符“[]”的重载	231
* 7-3-4 函数调用运算符“()”的重载	233
* 7-3-5 运算符 new 和 delete 的重载	235
7-4 类型转换	237
7-4-1 系统预定义类型间的转换	237
* 7-4-2 类类型与系统预定义类型间的转换	238
7-5 本章小结	242
习题	242

第 8 章 模板.....	245
8-1 模板的概念	246
8-2 函数模板与模板函数.....	247
8-2-1 函数模板的说明.....	247
8-2-2 函数模板的使用.....	247
8-2-3 用户定义的参数类型.....	250
* 8-2-4 函数模板的异常处理.....	251
8-3 类模板与模板类.....	252
8-4 程序举例	257
8-5 本章小结	264
习题	265
第 9 章 C++的输入和输出	269
9-1 C++为何建立自己的输入输出系统.....	270
9-2 C++的流库及其基本结构.....	270
9-2-1 C++的流	270
* 9-2-2 流类库	271
9-3 预定义类型的输入输出.....	273
9-3-1 无格式输入输出.....	273
9-3-2 格式化输入输出.....	275
9-4 用户自定义类型的输入输出.....	283
9-4-1 重载输出运算符“<<”	284
9-4-2 重载输入运算符“>>”	285
9-5 文件的输入输出.....	286
9-5-1 文件的打开与关闭.....	286
9-5-2 文件的读写.....	289
9-6 程序举例	294
9-7 本章小结	297
习题	298
*第 10 章 面向对象程序设计方法与实例	301
10-1 面向对象程序设计的一般方法和技巧	302
10-1-1 问题分析和功能定义.....	303
10-1-2 对象（类）设计及实现.....	303
10-1-3 核心控制设计.....	306
10-1-4 编码与测试.....	306
10-1-5 进化	306
10-2 设计实例	306
10-2-1 举例：模拟网上购书的结帐功能	306



C++ 面向对象程序设计

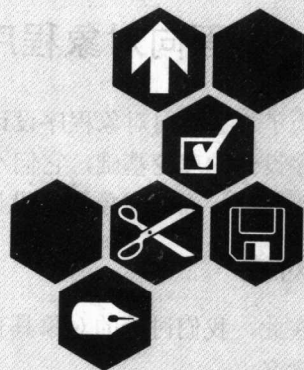
10-2-2 举例：模拟 ATM 机为用户提供服务	315
10-3 本章小结	335
习题	336
附录 上机实验题	337
实验一 C++基础练习	337
实验二 C++简单程序设计练习	337
实验三 类与对象(一)	338
实验四 类与对象(二)	339
实验五 派生类与继承	339
实验六 虚函数与多态性	340
实验七 函数模板与类模板	341
实验八 输入输出的格式控制	341
实验九 文件的输入与输出	342
实验十 综合练习	343
参考文献	345

1

面向对象程序设计概述

本章导读

20 世纪 90 年代以来面向对象程序设计 (Object Oriented Programming, 简称 OOP) 异军突起, 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, 面向对象程序设计的思想已经被越来越多的软件设计人员所接受, 不仅因为它是一种最先进的新颖的计算机程序设计思想, 更主要的是这种新的思想更接近人的思维活动, 人们利用这种思想进行程序设计时, 可以很大程度地提高编程能力, 减少软件维护的开销。面向对象程序设计方法是通过增加软件的可扩充性和可重用性来提高程序员的编程能力的。这种思想与我们以前使用的方法有很大的不同, 并且在理解上有一些难点, 希望本章的内容能对读者有所帮助。





1-1 什么是面向对象程序设计

1-1-1 新的程序设计范型

面向对象程序设计是一种新的程序设计范型 (Paradigm)。程序设计范型是指设计程序的规范、模型和风格，它是一类程序设计语言的基础。一种程序设计范型体现了一类语言的主要特征，这些特征能用以支持应用领域所希望的设计风格。不同的程序设计范型有不同的程序设计技术和方法学。

面向过程程序设计范型是使用较广泛的程序设计范型，这种范型的主要特征是，程序由过程定义和过程调用组成，即 程序 = 过程 + 调用。基于面向过程程序设计范型的语言称为面向过程性语言，如 C、PASCAL、Ada 等都是典型的面向过程性语言。函数式程序设计范型也是较为流行的程序设计范型，它的主要特征是，程序被看作“描述输入与输出之间关系”的数学函数。LISP 是支持这种范型的典型语言。除了面向过程程序设计范型和函数式程序设计范型外，还有许多其他的程序设计范型，如模块程序设计范型（典型语言是 Modula）、逻辑式程序设计范型（典型的语言是 PROLOG）、进程式程序设计范型、类型系统程序设计范型、事件程序设计范型、数据流程序设计范型等。

面向对象程序设计是一种新型的程序设计范型。这种范型的主要特征是：

程序 = 对象 + 消息

面向对象程序的基本元素是对象，面向对象程序的主要结构特点是：第一，程序一般由类的定义和类的使用两部分组成，在主程序中定义各对象并规定它们之间传递消息的规律。第二，程序中的一切操作都是通过向对象发送消息来实现的，对象接收到消息后，启动有关方法完成相应的操作。一个程序中涉及到的类，可以由程序设计者自己定义，也可以使用现成的类（包括类库中为用户提供的类和其他人已构建好的）。尽量使用现成的类，是面向对象程序设计范型所倡导的程序设计风格。

需要说明的是，某一种程序设计语言不一定与一种程序设计范型相对应。实际上存在有具备两种或多种范型的程序设计语言，即混合型语言。例如 C++ 就不是纯粹的面向对象程序设计范型，而是面向过程程序设计范型和面向对象程序设计范型的混合范型程序设计语言。

1-1-2 面向对象程序设计的基本概念

为了掌握面向对象程序设计技术，我们从最基本的概念入手。本节介绍的内容是面向对象程序设计的理论基础，它们不依赖于具体的程序设计语言，也就是说，无论使用哪种面向对象语言进行面向对象程序设计，本节内容都有指导意义。

1. 对象

在此，我们讨论的对象具有两方面的含义，即在现实世界中的含义和面向对象程序设计中的含义。

在现实世界中，任何事物都是对象。它可以是一个有形的具体存在的事物，例如一粒米、
可以是物体，也可以是抽象的

一辆汽车，甚至一个地球。它也可以是一个无形的、抽象的事件，例如一次演出、一场球赛、一次出差等。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，整个世界都可以认为是一个非常复杂的对象。

现实世界中的对象既具有静态的属性（或称状态），又具有动态的行为（或称操作、功能）。例如，每个人都有姓名、性别、年龄、身高、体重等属性，都有吃饭、走路、睡觉、学习等行为。所以在现实世界中，对象一般可以表示为：属性+行为。

现实世界中的对象，具有以下特性：

- (1) 每一个对象必须有一个名字以区别于其他对象。
- (2) 用属性来描述它的某些特征。
- (3) 有一组操作，每个操作决定对象的一种行为。
- (4) 对象的操作可以分为两类：一类是自身所承受的操作，一类是施加于其他对象的操作。

在面向对象程序设计中，对象是描述其属性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。对象可以认为是：数据+操作。对象所能完成的操作表示它的动态行为，通常也把操作称为方法。

为了帮助读者理解对象的概念，图 1-1 形象地描绘了具有 3 个操作的对象。

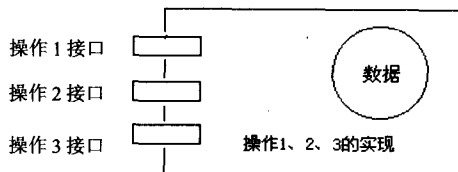


图 1-1 3 个操作对象

下面我们用一个录音机比喻一个对象，通俗地说明对象的某些特点。

录音机上有若干按键，如 Play（播放）、Rec（录音）、Stop（停止）、Rew（倒带）等，当人们使用录音机时，只要根据自己的需要如放音、录音、停止、倒带等按下与之对应的键，录音机就会完成相应的工作。这些按键安装在录音机的表面，人们通过它们与录音机交互。我们无法（当然也没必要）操作录音机的内部电路，因为它们被装在机壳里，录音机的内部情况对于用户来说是隐蔽的，不可见的。

一个对象很像一台录音机。当在软件中使用一个对象的时候，只能通过对对象与外界的接口来操作它。对象与外界接口也就是该对象向公众开放的操作，使用对象向公众开放的操作就好像使用录音机的按键，只需知道该操作的名字（如录音机的键名）和所需要的参数（用于提供附加信息或设置状态，好像听音乐前先装录音带并把录音带转到指定位置），根本无需知道实现这些操作的方法。事实上，实现对象操作的代码和数据是隐藏在对象内部的，一个对象好像是一个黑盒子，表示它内部状态的数据和实现各个操作的代码，都被封装在这个黑盒子内部，在外面是看不见的，更不能从外面去访问或修改这些数据或代码。

使用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内部实现算法，不仅使得对象的使用变得非常简单、方便，而且具有很高的安全性和可靠性。可见面向对象程序设计中的对象来源于现实世界，更接近人们的思维。



2. 类

在实现世界中，“类”是一组具有相同属性和行为的对象的抽象。例如，张三、李四、王五，……，虽然每个人的性格、爱好、职业、特长等各有不同，但是他们的基本特征是相似的，都具有相同的生理构造，都能吃饭、说话、走路等，于是把他们统称为“人”类，而具体的每一个人是人类的一个实例，也就是一个对象。

类和对象之间的关系是抽象和具体的关系。类是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。同理，桌子、教师、计算机等都是类。

类在现实世界中并不真正存在。例如，在地球上并没有抽象的“人”，只有一个个具体的人，如张三、李四、王五……。同样，世界上没有抽象的“学生”，只有一个个具体的学生。

在面向对象程序设计中，“类”就是具有相同的数据和相同的操作的一组对象的集合，也就是说，类是对具有相同数据结构和相同操作的一类对象的描述。例如，“学生”类可由学号、姓名、性别、成绩等表示其属性的数据项和对这些数据的录入、修改和显示等操作组成。在C++语言中把类中的数据称为数据成员。

在面向对象程序设计中，总是先声明类，再由类生成其对象。类是建立对象的“模板”，按照这个模板所建立的一个个具体的对象，就是类的实际例子，通常称为实例。打个比方，手工制作月饼时，先雕刻一个有凹下图案的木模，然后在木模上抹油，接着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，一个漂亮的图案就会出现在月饼上了。这样一个接着一个的，就可以制造出外形一模一样的月饼。这个木模就好比是“类”，制造出来的糕点好比是“对象”。

3. 消息

现实世界中的对象不是孤立存在的实体，他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，在面向对象程序设计中，对象之间也需要联系，我们称为对象的交互。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。这种机制叫消息传递。

在面向对象程序设计中的消息传递实际是对现实世界中的信息传递的直接模拟。以实际生活为例，我们每一个人可以为他人服务，也可以要求他人自己服务。当我们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。

在面向对象程序设计中，一个对象向另一个对象发出的请求被称为“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个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，有一个教师对象张三和一个学生对象李四，对象李四可以发出消息，请求对象张三演示一个实验，当对象张三接收到这个消息后，确定应完成的操作并执行之。

一般情况下，我们称发送消息的对象为发送者或请求者，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。接收对象只有在接收到消息时，才能被激活，被激活的对象会根据消息的要求完成相应的功能。

消息具有以下3个性质：

- (1) 同一个对象可以接收不同形式的多个消息，作出不同的响应。
- (2) 相同形式的消息可以传递给不同的对象，所作出的响应可以是不同的。
- (3) 对消息的响应并不是必需的，对象可以响应消息，也可以不响应。

在面向对象程序设计中，消息分为两类：公有消息和私有消息。假设有一批消息发向同一个对象，其中一部分消息是由其他对象直接向它发送的，称为公有（public）消息；另一部分消息是它向自己发送的，称为私有（private）消息。

4. 方法

在面向对象程序设计中，要求某一对象做某一操作时，就向该对象发送一个相应的消息，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。方法就是对象所能执行的操作。方法包括界面和方法体两部分。方法的界面也就是消息的模式，它给出了方法的调用协议；方法体则是实现某种操作的一系列计算步骤，也就是一段程序。消息和方法的关系是：对象根据接收到的消息，调用相应的方法；反过来，有了方法，对象才能响应相应的消息。所以消息模式与方法界面应该是一致的。同时，只要方法界面保持不变，方法体的改动不会影响方法的调用。在C++语言中方法是通过函数来实现的，称为成员函数。

1-1-3 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。本节介绍面向对象程序设计的4个基本特征。

1. 抽象

抽象是人类认识问题的最基本的手段之一。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，抽象强调感兴趣的信息，忽略了不重要的信息。例如，在设计一个学籍管理程序的过程中，考察某个学生对象时，只关心他的姓名、学号、成绩等，而对他的身高、体重等信息就可以忽略。

以一般观点而言，抽象是通过特定的实例（对象）抽取共同性质以后形成概念的过程。抽象是对系统的简化描述或规范说明，它强调了系统中的一部分细节和特性，而忽略了其他部分。抽象包括两个方面：数据抽象和代码抽象（或称为行为抽象）。前者描述某类对象的属性或状况，也就是此类对象区别于彼类对象的特征物理量；后者描述了某类对象的共同行为特征或具有的共同操作。

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。

现在以学生管理程序为例，通过对学生进行归纳、分析，抽取出其中的共性，可以得到如下的抽象描述：

- (1) 共同的属性：姓名、学号、成绩等，它们组成了学生的数据抽象部分。用C++语言的数据成员来表示，可以是：

```
char* name; int number; float score;
```