

普通高等教育机电类规划教材

C++程序设计

——从面向过程到面向对象

主 编 胡学龙

副主编 陈 峻 刘其明

参 编 景雪琴 聂世澄

主 审 潘志庚



机械工业出版社

本书以面向对象程序设计语言 C++为主线, 兼顾无 C 语言程序设计基础的读者顺利地接受 C++的思想。C++语言兼容 C 语言的面向过程程序设计, 保持 C 语言高效、简洁的特点, 又适合于软件开发者的思维方式, 可重用性、可扩充性强, 可读性、可理解性更好, 已成为软件开发者优选的程序设计语言之一。本书配有习题集和上机指导书《C++程序设计学习指导与上机实践》。本书循序渐进, 深入浅出, 重点突出, 难点分解, 可读性强, 可作为大专院校学生、计算机等级考试者的教材, 也可供计算机应用开发人员作为参考资料。

图书在版编目 (CIP) 数据

C++程序设计: 从面向过程到面向对象/胡学龙主编. - 北京: 机械工业出版社, 2001.8

普通高等教育机电类规划教材

ISBN 7-111-08976-6

.C... .胡... . C 语言 - 程序设计 - 高等学校 - 教材 . TP312

中国版本图书馆 CIP 数据核字 (2001) 第 031692 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 王世刚 张祖凤

封面设计: 责任印制:

印刷·新华书店北京发行所发行

2001 年 7 月第 1 版第 1 次印刷

1000mm × 1400mm B5 · 11 印张 · 428 千字

0 001—6 000 册

定价: 26.00 元

本书内容如有更改或与实际操作不符, 恕不另行通知
凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换
本社购书热线电话 (010) 68993821、68326677-2527

普通高等教育机电类规划教材编审委员会

主任委员：邱坤荣

副主任委员：黄鹤汀

左健民 高文龙

王晓天 蔡慧官

秘 书：周骥平

委 员：(排名不分先后)

沈世德 周骥平

徐文宽 唐国兴

韩雪清 戴国洪

李纪明 吴建华

鲁屏宇 王 钧

赵连生

序

人类满怀激情刚刚跨入充满机遇与挑战的 21 世纪。这个世纪是经济全球化、科技创新国际化的世纪,是新经济占主导地位的世纪,是科学技术突飞猛进、不断取得新突破的世纪。这个世纪对高等教育办学理念、体制、模式、机制和人才培养等各个方面都提出了全新的要求,培养的人才必须具备新思想、新观念、不断创新、善于经营和开拓市场、其团队精神等素质。

机械高等工程教育是我国高等教育的重要组成部分,21 世纪对它的挑战同样是严峻的。随着现代科学技术的迅猛发展,特别是微电子技术、信息技术的发展,它们与机械技术紧密结合,从而形成传统制造技术、信息技术、自动化技术、现代管理技术等交融、渗透的先进制造技术,使制造业和制造技术的内涵发生了深刻的变化。面向 21 世纪的机械制造业正从以机器为特征的传统技术时代迈向以信息为特征的系统技术时代。制造技术继续沿着 20 世纪 90 年代展开的道路前进。制造技术和自动化水平的高低已成为一个国家或地区经济发展水平的重要标志。而目前我国的制造技术与国际先进水平还有较大差距,亟需形成我国独立自主的现代制造技术体系。面对这一深刻的变化和严峻的形势,我们必须认真转变教育思想,坚持以邓小平同志提出的“三个面向”和江泽民同志提出的“四个统一”为指导,以持续发展为主题,以结构优化升级为主线,以改革开放为动力,以全面推进素质教育和改革人才培养模式为重点,以构建新的教学内容和课程体系、深化方法和手段改革为核心,努力培养素质高、应用能力强、富有创新精神和特色的应用性的复合型人才。

基于上述时代背景和要求,由机械工业出版社教编室、江苏省教育厅(原江苏省教委)、江苏省以及部分省外高等工科院校成立了教材编审委员会,并组织编写了机械工程及自动化专业四个系列成套教材首批 31 本,作为向新世纪的献礼。

这套教材力求具有以下特点:

- (1) 科学定位。本套教材主要用于应用性本科人才的培养。
- (2) 强调实际、实践、实用,体现“浅、宽、精、新、用”。所谓“浅”,就是要深浅适度;所谓“宽”,就是知识面要宽些;所谓“精”,就是要少而精,不繁琐;所谓“新”,就是要跟踪应用学科前沿,跟踪技术前沿,推陈出新,反映时代要求,反映新理论、新思想、新材料、新技术、新工艺;所谓“用”,就是要理论联系实际,学以致用。
- (3) 强调特色。就是要体现一般工科院校的特点、特色,符合一般工科院校的实际教学要求,不盲目追求教材的系统性和完整性。
- (4) 以学生为本。本套教材尽量体现以学生为本、以学生为中心的教育思想,不

为教而教，要有利于培养学生自学能力和扩展、发展知识能力，为学生今后持续创造性学习打好基础。

当然，本套教材尽管主观上想以新思想、新体系、新面孔出现在读者面前，但由于是一种新的探索以及其他可能尚未认识到的因素，难免有这样那样的缺点甚至错误，敬请广大教师和学生以及其他读者不吝赐教，以便再版时修正和完善。

本套教材的编审和出版得到了机械工业出版社教编室、江苏省教育厅以及各主审、主编和参编学校的大力支持和配合，在此，一并表示衷心感谢。

普通高等教育机械工程及自动化专业机电类规划教材编审委员会

主任 邱坤荣

前 言

程序设计是大学计算机教育的重要组成部分和培养创新能力的工具。在众多的高级语言中,C++语言无疑在鼓励面向对象编程方面、在运行效率方面、在理解计算机的基本概念方面、在用户的使用比例和在工程应用中跨系统软件和应用软件的覆盖面等方面都占有独特的地位,已经成为许多院校计算机专业和非计算机专业第一门算法语言。

多年来,本教材的编写者在程序设计的教学、科研和工程应用的实践中积累了一些经验,获得了多项省级教学成果,相关课程多次被评为省级优秀课程。我们认为,在区分层次、区分阶段进行面向对象技术教学的过程中,面向对象程序设计(OOP)是一个重要环节。贯彻以读者为中心,注重实效是我们编写该教材的出发点。为此我们介绍了学习 C++程序设计的注意事项,采用了循序渐进、重点突出、难点分解、实例说明、图表助学、C++语言与 C 语言对比、每章对重点和难点进行小结等方法尽可能地做到深入浅出。书中的源程序均在 Visual C++ 6.0 编程环境中通过。为了给读者在阅读技术资料 and 调试程序时提供方便,本书还提供了 C++程序设计方面的常用词汇中英文对照表。对照表提供了按英文字母排序、按汉语拼音排序两种索引方式,查找方便。

我们希望本书能成为初学 C++程序设计语言的入门教材,同时也可供 C 语言程序设计教学使用。通过本书和配套辅助教材《C++程序设计学习指导与上机实践》(吴访升主编)的学习,读者可在较短的时间内完成 C 到 C++、面向过程到面向对象的过渡。本书也可作为计算机等级考试、软件技术培训用教材。

本书由胡学龙担任主编,陈峻、刘其明担任副主编,景雪琴、聂世澄参加了编写。具体分工如下:胡学龙编写了绪论、第一章、第五章第十一节至十五节、第八章第八节、第九章、第十章、第十一章第一节至第四节、各章小结、附录。陈峻编写了第二章,第五章第一节至第十节。刘其明编写了第三章、第四章。景雪琴编写第六章、第七章第四节、第八章第一节至第七节。聂世澄编写了第七章第一节至第三节和第五节至第九节、第十一章第五节。在各作者分工合作的基础上,最后由胡学龙对全书进行了统稿。

浙江大学计算机系、CAD&CG 国家重点实验室博士生导师潘志庚教授担任主审。潘志庚教授在百忙之中审阅了书稿,对本书提出了许多很好的建议和改进意见,为保证本书的质量起到了关键作用。

本书的编写得到扬州大学第一批教材建设基金、“面向 21 世纪教学内容和课程体系改革”项目经费、首批教学改革试点专业建设项目经费的资助。同时也得到机械工业出版社和江苏省教育厅有关领导、专家的指导。本书在编写工作中得到了周国华、

陆峰等研究生的热情帮助，他们在参与部分章节的撰写和校对方面付出了辛勤劳动。在此一并表示衷心感谢。

由于我们在编写统编教材方面还缺少经验，加之对整个计算机教育的要求把握不够，书中一定会有不少疏忽谬误之处，恳请读者批评指正。

编 者

2001年7月于扬州大学

目 录

前言

绪论	1
第一节 程序设计在计算机科学 和技术中的作用	1
第二节 面向过程设计与面向 对象程序设计	3
第三节 算法及其表示方法	5
第四节 程序设计是从事创新 活动的工具	6
第五节 学习程序设计的一般方法	7
第一章 C++语言概述	9
第一节 C++语言发展简史	9
第二节 C++语言的特点	10
第三节 C++程序的结构	11
第四节 C++语言的基本组成	13
一、C++语言的基本符号	13
二、C语言的词法单位	14
第五节 C++程序集成开发环境 (IDE)	15
一、C++编程环境与源程序 处理过程	15
二、C++编译基本原理	16
本章小结	18
第二章 基本数据类型、运算符及表达式	19
第一节 C++语言的基本数据类型	19
一、整型数据	20
二、字符型数据	22
三、字符型变量	23
四、字符串型数据	24
五、变量赋初值	25
六、常数类型说明符 const	25

第二节 运算符及表达式	26
一、运算符的优先级与结合性	26
二、赋值运算符	27
三、算术运算符	29
四、关系运算符	29
五、逻辑运算符	30
六、自增与自减运算符	31
七、逗号运算符	32
八、强制类型转换运算符	32
第三节 表达式的类型及其运算	33
一、表达式的类型	33
二、类型转换	33
第四节 位运算符及位运算	34
一、按位与运算	35
二、按位或运算	35
三、按位异或运算	36
四、取反运算符	37
五、左移运算符	37
六、右移运算符	38
本章小结	38
第三章 控制结构	39
第一节 顺序结构程序设计	39
一、程序的结构与控制	39
二、C++语言的语句	40
三、数据的输入输出	41
四、程序举例	55
第二节 选择结构程序设计	57
一、条件语句	57
二、条件运算符	63
三、switch 语句	63

四、程序举例	65	第三节 函数值与 return 语句	118
第三节 循环结构程序设计	69	一、函数的返回值	118
一、for 循环语句	69	二、return 语句的功能与用法	118
二、while 循环语句	72	第四节 函数的参数	120
三、do ~ while 语句	72	一、形式参数与实在参数	120
四、循环的嵌套	74	二、参数值的传递	120
五、break、continue 与 goto 语句	76	三、数组名作为函数参数	121
六、程序举例	80	第五节 函数的调用	123
本章小结	85	二、对被调函数的声明	
第四章 数组	86	与函数原型	125
第一节 数组的定义格式	86	三、函数嵌套调用	127
一、一维数组	86	四、函数递归调用	128
二、二维数组	87	第六节 变量的存储种类	132
第二节 数组元素	88	一、局部变量	133
第三节 数组的初始化	88	二、全局变量	136
一、一维数组的初始化	88	第七节 内部函数与外部函数	142
二、二维数组的初始化	90	第八节 宏定义	144
第四节 数组的应用	92	一、符号常数定义	144
第五节 字符数组及字符串	101	二、带参数的宏定义	149
一、字符数组的定义和初始化	101	第九节 文件包含	151
二、字符数组的应用	103	第十节 条件编译	153
第六节 字符串处理函数	103	第十一节 缺省参数的函数	155
一、字符串输入函数 gets()	104	第十二节 内置函数	156
二、字符串输出函数 puts()	104	第十三节 函数名重载	156
三、字符串连接函数 strcat()	105	第十四节 作用域限定运算符	157
四、字符串比较函数 strcmp()	106	第十五节 多文件程序的编译和连接	158
五、字符串复制函数 strcpy()	107	本章小结	158
六、计算字符串长度函数 strlen	108	第六章 指针	160
第七节 字符数组的应用	109	第一节 指针及指针变量	160
本章小结	110	一、指针的概念	160
第五章 函数与编译预处理	111	二、指针操作符	161
第一节 库函数的调用	111	三、指针的运算	162
一、数学函数	112	第二节 指针与数组	168
二、输入输出 (I/O) 函数	113	一、一维数组与指针	168
三、字符函数和字符串函数	114	二、用数组名和用指针变量	
第二节 函数的定义	114	指向数组的区别	171

三、多维数组与指针.....	174	四、消息.....	228
第三节 指针与字符串.....	178	五、继承.....	228
一、字符串的指针表示.....	178	第二节 类的结构.....	229
二、用数组名和用指针处理		一、类的定义.....	229
字符串的区别.....	179	二、类与结构体的异同.....	230
三、指针数组.....	180	第三节 对象.....	231
第四节 多级指针.....	182	第四节 构造函数与析构函数.....	238
第五节 指针与函数.....	183	一、构造函数.....	238
一、指针作为函数的参数.....	183	二、析构函数.....	243
二、函数的入口地址和函数的指针.....	187	三、实现复制的构造函数.....	246
三、返回指针值的函数.....	189	四、构造函数和对象成员.....	249
第六节 函数 main()的参数.....	191	第五节 友元函数.....	252
第七节 new 和 delete 操作符.....	193	第六节 类和对象的作用域.....	256
一、new 和 delete 操作符的		第七节 引用.....	257
使用方法.....	193	一、引用的概念.....	257
二、使用 new 和 delete 操作符		二、独立引用.....	257
应注意的事项.....	195	三、作为参数的引用.....	258
本章小结.....	198	四、引用返回值.....	260
第七章 结构体与共用体.....	199	第八节 类自引用及 this 指针.....	261
第一节 结构体的定义.....	199	本章小结.....	265
第二节 结构体的使用方法.....	200	第九章 继承与派生类.....	266
第三节 指向结构体的指针.....	202	第一节 继承的基本概念.....	266
第四节 指针在链表中的应用.....	203	第二节 继承与数据成员的可视性.....	266
一、单向无序链表的操作.....	205	第三节 派生类的构造.....	267
二、单向有序链表的操作.....	209	一、继承基类私有成分.....	267
第五节 结构体与函数.....	214	二、继承基类公有成分.....	269
第六节 共用体.....	217	三、继承基类保护成分.....	272
第七节 枚举类型.....	219	第四节 成员访问控制.....	272
第八节 用 typedef 自定义类型.....	221	第五节 类属类简介.....	280
第九节 位段.....	222	第六节 多重继承.....	282
本章小结.....	226	本章小结.....	285
第八章 类与对象.....	227	第十章 运算符重载与虚拟函数.....	286
第一节 类和对象的基本概念.....	227	第一节 多态性的基本概念.....	286
一、类.....	227	第二节 早期联编和滞后联编.....	286
二、对象.....	227	第三节 运算符重载.....	287
三、方法.....	228	一、运算符重载的意义.....	287

二、运算符重载的实现.....	288	一、文件流类体系.....	316
三、运算符重载实例.....	295	二、文件的打开.....	316
第四节 虚拟函数.....	296	三、文件的关闭.....	318
一、虚拟函数的实现.....	296	四、文本文件的读写操作.....	318
二、空虚函数.....	301	五、二进制文件读写操作.....	319
三、纯虚函数与抽象类.....	304	第四节 RAM 流.....	319
本章小结.....	306	第五节 C 语言的文件系统.....	321
第十一章 文件与流类库.....	307	一、C 文件系统的概念.....	321
第一节 文件和流的基本概念.....	307	二、文件的打开与关闭.....	322
一、文件的概念.....	307	三、文件的输入输出.....	323
二、流的概念.....	308	四、文件的定位与状态.....	330
第二节 基本流类体系.....	308	本章小结.....	332
一、流类体系.....	309	附录 A C++在非 C 的增强.....	333
二、预定义的流与流运算符.....	309	附录 B 常用词汇中英文对照表.....	335
三、流的格式控制.....	310	参考文献.....	340
第三节 文件流.....	316		

绪 论

在工程实践中，传统的设计概念源远流长，而程序设计技术则是伴随着计算机的产生和发展不断前进。仅仅几十年的历史，程序设计已经成为大学计算机教育的基石。在信息时代掌握至少一门程序设计语言是新世纪人才必须具备的业务素质之一，它使计算机文化类的常识型、操作型课程过渡到思维型、设计型课程，在创新教育中起着显著的作用，对大学生的知识和能力的提高都有重大的影响。

第一节 程序设计在计算机科学和技术中的作用

程序设计语言，又称编程语言，在计算机科学和技术中具有以下三大功能：

(1) 软件描述作用。软件包括程序、数据和相关文档等。语言的优劣直接影响到程序的内容，自然会影响到程序及其软件的质量。如 PASCAL 语言，严谨流畅，清晰易懂，对数据结构和算法的教学十分适宜；FORTRAN 语言为公式翻译程序设计语言，为科技工作者进行科学计算创造了条件；C 语言原先为辅助开发 UNIX 操作系统而设计，主要用于系统程序设计，由于其高效性后来也拓宽到应用程序设计；C++语言在 C 语言的基础上增加了面向对象的成分，成为 C 程序员乐于接受的面向对象程序设计语言；JAVA 语言面向 Internet 上的 WWW，适用于网络和分布式计算，其简明性、面向对象性、安全性、易移植性、多平台操作、多线程等性能十分显著，成为普及速度最快的语言。

(2) 软件交流工具。软件反映的内容如果仅存在于软件开发者的大脑中是无法交流的。先进的设计思想必须通过编程语言才能在计算机上实现。同样，软件只有用语言描述才能得到社会上广泛的应用。例如：由于 C++和 JAVA 语言的先进性使得它们推出后便成为面向对象语言中的主流语言。

(3) 软件发展和计算机发展水平的标志。代表性语言是软件发展的重要标志。如：FORTRAN 是第一个高级语言，其出现大大减轻了科技人员的计算工作量；LISP 语言推动了表处理技术的发展；SIMULA 67 引入了“对象”和“类”的概念，成为第一个面向对象语言（或称对象式语言）；随后，SMALLTALK、EIFFEL、C++和 JAVA 等语言各具特色，成为典型的面向对象语言。

图 0-1 表示了一些重要的程序设计语言之间的衍生关系。其中：

(1) 具有对象概念的程序设计语言有三类：基于对象的（object-based）、基于类的（class-based）和面向对象的（object-oriented）。第一类语言只有“对象”的概念，而没有将对象的共性抽象化进一步形成“类”；第二类语言引入了“类”，将数据以及对数

据的操作合为一体，更加符合自然域的求解要求；第三类语言提供了类的继承机制，使类形成了层次结构，这对软件的可复用性提供了有力支持。可见，类及其继承是面向对象的标志。有些语言虽然不能称为面向对象语言，但在某些特征上为面向对象语言的产生奠定了基础。

(2) 按照语言的成分性质可分为顺序语言（一般应用场合）并发（用于并行计算）和分布语言（用于分布式计算）。

(3) 按计算机的设计背景可分为冯·诺依曼式语言和非冯·诺依曼式语言。一般语言为前者，而 1977 年提出的函数式语言 FP 以非冯·诺依曼机为其设计背景，属于非冯·诺依曼语言。

(4) 逻辑式语言是处理逻辑问题的语言，广泛应用于关系数据库、数理逻辑、抽象问题求解、自然语言理解等领域。

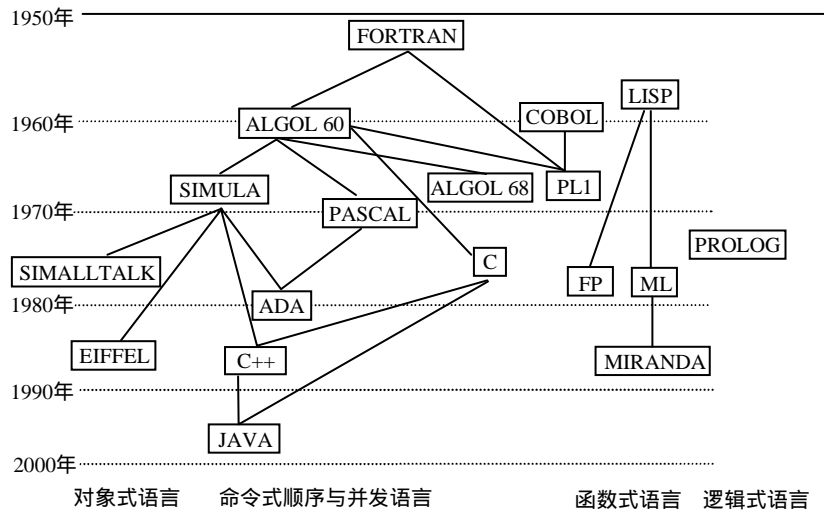


图 0-1 重要语言的衍生关系

目前高等院校所开设的各种计算机语言的主要特点如表 0-1 所示。

表 0-1 常用计算机语言的特点

语 种	主 要 特 点
BASIC 语言	简便易学，是一种大众化语言。许多大学曾开设过 True BASIC。目前在可视化、多媒体应用环境下的程序开发语言是 VB（Visual BASIC）
FORTRAN 语言	第一个程序设计高级语言，计算功能强大，是典型的面向过程程序设计语言（POPL），其最新版本 FORTRAN 90 增强了在数据结构的表示和处理方面的能力
PASCAL 语言	充分体现了算法、结构化设计方法和数据结构的关系，适用于计算机语言教学
C 语言	“中级语言”的代表，既用来编写系统软件，又可编写应用软件。工程应用广泛，许多学校和专业将其选为第一算法语言
C++语言	在 C 语言基础上实现了面向对象程序设计机制，是面向对象程序设计语言（OOPL）的典型代表之一，VC++（Visual C++）是建立在可视化、多媒体应用环境下的程序开发语言

(续)

语 种	主 要 特 点
XBASE, FoxPro 语言	数据库用语言, 它包括 dBase、FoxBase、FoxPro。VFP (Visual FoxPro) 是 FoxPro 在多媒体应用环境下的程序开发语言, VFP3.0 是第一个面向对象的微机数据库的软件开发工具
Delphi 语言	一种非过程化第四代语言 (4GL), 在可视化开发环境下, 具有强大的功能组件和数据库开发工具
LISP 语言	人工智能用语言, 通常用于计算机专业的本科生和研究生学习人工智能等课程用
JAVA 语言	网络计算用语言, 随着 Internet 的普及, 其作用将越来越大, 一些院校的专业将其设为专业选修课
Ada 语言	军方等要求高可靠性、实时、安全、大型软件场合下选用的软件, 被认为功能最强、最能体现软件工程特点的语言
汇编语言	面向特定机器的符号语言, 是一种低级语言, 具有高效等特点, 适用于编写系统程序、控制程序。要求编程者必须了解计算机的内部结构, 它是学习微机原理、接口技术、单片机等课程的基础语言

第二节 面向过程程序设计 with 面向对象程序设计

按照用户的要求, 编程语言可分为过程语言和非过程语言。相应地, 面向过程的设计 (POP, procedure-oriented programming) 关心的是用程序表达对问题的求解过程, 用户可以显式地指明一系列可执行指令 (或语句) 来表达程序的处理过程。凡是用户不能显式地指明这一过程的语言称为非过程语言。

传统的编程方法是面向过程的程序设计。此方法以问题的处理过程为中心, 对问题的求解步骤是根据问题所依据的数据结构来确定算法, 编制程序, 求得结果。20 世纪 70 年代人们采用了结构化程序设计 (structured programming) 来对复杂的大型问题进行分解, 逐步精化, 直至其算法表示可用计算机实现为止。该方法对提高程序的可理解性、可维护性等性能提供了一种重要的软件开发方法, 使面向过程技术更加成熟。

由于一般问题都比较复杂, 在 POP 中, 为控制程序的复杂性, 采用功能分解的方法, 将复杂度较高的问题分解为复杂度较低的问题, 这样逐层分解直到具有单一功能为止。这样分解以后的模块单位在程序设计中称为“过程 (procedure)”或“函数 (function)”。POP 一般与结构化程序设计相配合, 采用“自顶向下 (top-down)”、“逐层精化 (refine)”和“模块化 (modular)”等分解手段。这里“自顶向下”与“逐层精化”指先进行总体设计, 将复杂的问题分层处理, 实现功能上的分解 (function decomposition)。“模块化”指经过分解以后的功能可以用“功能模块”具体实现。各个模块可以分配给不同的程序员编制。以功能为目标分解一个大任务为各个子任务, 从而便于组织多人协作完成复杂的软件工程。

在面向过程程序设计中, 过程是关心的重点, 数据结构的设计服从于对过程的组织, 处于相对次要的地位。C 程序主函数 main () 的主要作用是反映程序的高层过程, 再

由这些过程调用较低层的过程，依此类推。因而面向过程的 C 程序是“函数”的集合。

与面向过程程序设计形成鲜明对比的是，面向对象的程序设计将数据和对数据的操作封装在“类”中，这样说明抽象数据类型比较自然，对实际问题的描述、对信息的隐蔽性、结构性和可读性等都比面向过程程序控制好。C++ 程序主函数 `main ()` 反映的是问题的主要对象，再由这些对象与其他对象通过消息发生联系，依此类推。显然，面向对象的 C++ 程序是“对象”的集合。

与我们将具有对象概念的程序设计语言分成三类相对应，在通向面向对象的道路上有如图 0-2 所示的 7 个台阶。其中：

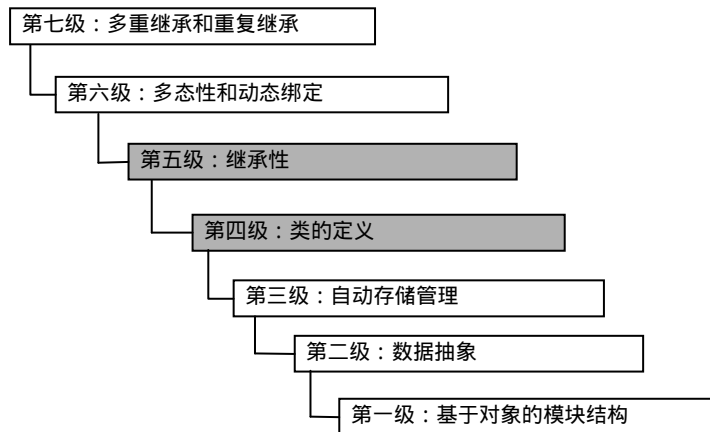


图 0-2 面向对象的 7 个台阶

(1) 基于对象的模块结构 (modular structure) 指基于数据结构而不是基于功能进行模块化。

(2) 数据抽象指对象是抽象数据类型 (Abstract Data Type, ADT) 的具体实现。

(3) 自动存储管理指对象应该能由编译系统进行分配、回收等管理。

(4) 类的定义是 ADT 的一种表示方式，它将数据及其相应操作封装起来组成一个整体，因而综合了模块和数据类型的特性。

(5) 继承性使下一层次的子类 (派生类) 对上一层次父类 (基类) 的数据成员和操作功能进行扩充和“裁剪”。继承性是软件成分可复用的基础。

(6) 多态性指同一操作可以有多个不同的定义，因此对于不同的应用场合可以有不同的表现。动态绑定指一个程序实体能指向多个类的对象。

(7) 多重继承指一个类可以是上一层次多个类的子类，而重复继承指一个类可以多次成为同一个类的子类。

面向对象技术是由面向对象程序设计起源的。在 20 世纪 90 年代由于大型软件工程、多媒体信息处理技术、Internet 技术的兴起使面向对象技术得到充分发展，它在处理图形、图象、动画、视频等海量、复杂数据等方面起到了其他方法不可替代的作用，成

为新一代软件工程的主流技术。读者从表 0-2 中不难看出不同复杂度的典型数据类型和程序设计语言之间的对应关系。

表 0-2 不同复杂度的数据类型与程序设计语言的对应性

数据类型	二进制 (0,1)	整数型、浮点实型、字符型	记录、结构	音频、图形、图像、视频、动画等多媒体类型
程序设计语言	机器语言	FORTRAN 等 POPL	PASCAL、C 等 POPL	C++ 等 OOPL

第三节 算法及其表示方法

大型软件的开发必须按照软件工程学的原理来进行。一般规模的程序设计可以按照图 0-3 的步骤来进行，其中算法的设计是编写程序的基础。

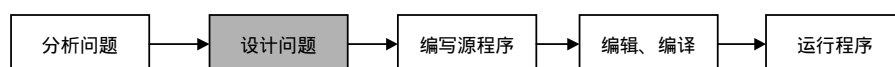


图 0-3 程序设计的一般步骤

著名计算机科学家 N. Wirth 教授曾经提出一个著名的公式：

$$\text{算法} + \text{数据结构} = \text{程序}$$

该公式说明进行程序设计要做好两方面的工作：描述数据，即表示数据的类型或数据结构 (data structure)；另一个就是对数据的操作，即算法。

根据软件技术发展的现状，可以进一步完善上述公式：

$$\text{程序} = \text{算法} + \text{数据结构} + \text{程序设计方法} + \text{语言工具和环境}$$

算法 (algorithm) 是计算机软件中的一个基本概念，它是对解决实际问题的方法和步骤的描述。设计算法是程序设计的核心，而算法语言是计算机软件的精髓。此公式充分体现进行程序设计各个方面的组成。程序设计者要逐步培养将算法、程序、软件三者有机结合在一起的能力，真正将语言作为工程工具来使用，使计算机充分发挥其效用，体现精通算法语言的目的全在于应用算法语言。

正确的算法有以下几个显著特点：

(1) 有穷性 (finiteness)。一个算法是有限步骤的描述。根据问题的复杂性程度，在物理上，它可由计算机在合理的时间内完成执行。在一般情况下，人们根据常识不难确定“合理的时间”的数量级。在算法中如果出现“死循环”等现象，就不符合算法有穷性的要求。

(2) 确定性 (certainty)。算法的每一个步骤都必须是明确的，无二义性 (ambiguity) 的。计算机只能根据程序员给它的具体明确的步骤执行，且同一个步骤不能作多种理解。这说明算法对于相同的输入只能得出相同的输出。在算法描述中如果出现“好像”之类的似是而非的语法成分，则此算法是非确定的。

(3) 有效性 (effectiveness)。又称可行性,指算法中的每一步都能有效地执行,得出确定的结果。如果一个数被 0 除就不满足有效性。

(4) 有零个或多个输入 (input)。算法一般应具有供加工的原始数据。如果算法本身能够生成数据,则可不需外界数据。

(5) 有一个或多个输出 (output)。算法实现的目标是经过加工处理后获得结果,此结果只有输出才能为用户所用。所以算法至少要有个输出。

算法的质量是一个十分重要的问题。由于在软件开发中先有算法后有程序,显然程序不可能比算法的质量更高。好的算法应达到以下几个目标:

(1) 正确可靠。不正确的程序不但不能解决问题,反而会给我们的工作带来不必要的麻烦,甚至造成重大损失。

(2) 清晰易读。在程序设计发展的早期,由于计算机的速度和存储容量都受到限制,往往将程序的效率放在第一位,随着硬件水平的迅速提高,人们对一般问题已经从“效率第一”转向为“清晰第一”,程序的可读性和可理解性成为优质软件的重要标志。

(3) 执行程序的时间效率和占用内存的空间效率要高。在有些情况下,当效率成为制约程序运行的主要因素时,编程者需要在效率和可读性之间作出折中的选择。

(4) 通用性要好,对异常情况要能作出适当的反应,留有出口。

第四节 程序设计是从事创新活动的工具

在现代信息社会,创新过程若以程序设计作为计算、模拟、显示、控制等工具,将会大大地提高效率。

(1) 程序设计中凝聚了许多现代思想方法,如信息论、系统论、控制论等,程序设计的发展过程本身也是一个创新过程。至少我们从以下几个方面可以看出程序设计的一些新思路:

系统论的观点要求程序要结构化、模块化。当程序较大时,还要按照软件工程的原则建立“黑箱”结构模型、逐步“白箱化”进行开发。目前,程序设计和工程设计已密不可分,形成“工程设计程序化,程序设计工程化”的局面。

面向对象是一种认知方法,它要求我们用一个统一的方法将认识一个系统工程的方法和设计与实现一个系统的过程的方法统一起来。面向对象的程序设计方法正是面向对象方法学在程序设计中的具体体现。

可视化思想要求程序设计语言在图形方式下,应用形象思维方式表现抽象复杂的自然现象和工程问题。现在已经出现了像 VB、VC++、VFP 等在可视化操作界面下的很受大众欢迎的程序设计语言。对 Windows 应用程序的分析、编制、调试等技术已经作为学习程序设计的重要组成部分。

混合语言编程的思想要求程序员全面地看待各种编程语言的优缺点,扬长避短,