新编21世纪高等院校计算机系列规划教材

C/C++语言程序设计

北京希望电子出版社 总策划
董晓华 主 编
李 崇 姜 雷 副主编
董晓华 张立华 甘 玲 编 著

科学出版社
www.sciencep.com

新编21世纪高等院校计算机系列规划教材

C/C++语言程序设计

北京希望电子出版社 总策划
董晓华 主编
李崇 姜雷 副主编
董晓华 张立华 甘玲 编著

科学出版社
www.sciencep.com

内容简介

本书针对C语言初学者进行编写，前一部分对C语言的基本语法和常用语句进行了详细介绍，并提供大量实例加以说明。后一部分作为C语言的加深，介绍了程序设计中常用的算法和数据结构，并对C++进行了一定的介绍，对C语言的后续学习奠定了基础。

本书适合作为培养应用型人才院校的计算机及相关专业教学用书，也可作为专业人士的开发参考书，还可作为全国计算机等级考试二级（C语言）培训教材或自学参考书。

需要本书或技术支持的读者，请与北京中关村083信箱（邮编：100080）发行部联系，电话：010-82702660，62978181（总机）传真：010-82702698，E-mail：tbd@bhp.com.cn。

图书在版编目（CIP）数据

C/C++语言程序设计 / 董晓华主编. —北京：科学出版社，2005.8

ISBN 7-03-015578-5

Ⅰ. C… Ⅱ. 董… Ⅲ. C语言—程序设计 Ⅳ. TP312

中国版本图书馆CIP数据核字（2005）第050824号

责任编辑：王楠楠 / 责任校对：王春桥
责任印刷：媛 明 / 封面设计：梁运丽

科学出版社 出版
北京东黄城根北街16号
邮政编码：100717
http://www.sciencep.com
北京媛明印刷厂
科学出版社发行 各地新华书店经销
*
2005年8月第 一 版 开本：787×1092 1/16
2005年8月第一次印刷 印张：14
印数：1— 3000 字数：321 594

定价：20.00元

新编21世纪高等院校计算机系列规划教材编委会

主　任： 陈火旺　全国工科院校计算机专业教学指导委员会主任
中国工程院院士

副主任： 李国杰　中国计算机学会理事长
计算技术研究所所长

杨芙清　中国计算机学会副理事长
中国科学院院士

沈复兴　全国高等师范学校计算机教育研究会副理事长
北京师范大学信息科学学院院长

何炎祥　武汉大学计算机学院院长

桂卫华　中南大学信息科学与工程学院院长

李仁发　湖南大学计算机与通信学院院长

陆卫民　中国科学出版集团北京希望电子出版社社长

委　员：（按姓氏笔画为序）

王江晴　王行恒　甘　玲　邓志华　孙中胜　刘晓燕　匡　松
任达森　李华贵　李超锋　李节阳　李新国　李龙澍　李建平
何婷婷　何登旭　张友生　张洪瀚　罗　琳　杨　波　杨宪泽
武兆辉　陈浩杰　陈　庄　郑明红　赵振华　洪汝渝　徐建军
徐　谡　唐光海　唐霁虹　唐　雁　高　丽　阎怀志　曹永存
覃　俊　董玉萍　董晓华　谢秉元　詹国华　戴上平

秘　书： 徐建军

前 言

C语言是近年来国内外广泛推广应用的一门计算机语言，不仅为计算机专业人员所使用，而且为广泛的非专业人员所青睐。

C语言功能强大、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好。既有高级语言的优点，又有低级语言的许多特点。因此，常常被作为计算机的入门学习语言，比学习和使用汇编语言容易得多。

目前，许多高校不仅将C语言作为计算机专业课程来开设，而且还在非计算机专业中将其作为一门语言工具来学习。计算机水平考试、等级考试等都将C语言列入考试范围，用以衡量计算机专业人员和非计算机专业人员的计算机水平，许多人已经用它来开发系统软件和应用软件。

但是，C语言涉及到的概念比较复杂，使用灵活，但容易出错；因此许多初学者都感到非常困难。针对C语言比较难学的情况，作者对全书内容做了精心安排，分解难点，减小台阶，用通俗易懂的语言和丰富的例题解释清楚复杂的概念。

本书对C语言的基本语法和语句进行详细介绍，并在此基础上进行加深，介绍了相关的程序设计算法和数据结构。具体说来，有以下特点：

（1）内容编写思路新颖、注重实用。以实际问题引出概念，在例题中讲解语法及注重分析问题，便于初学者接受；同时强调实用性，在讲解语言中给出应用建议，使读者在掌握语法的同时明确它的实际用途。

（2）注重程序设计能力的培养。在介绍C语言基本知识的同时，结合实例着重介绍程序设计方法，使读者逐步建立起程序结构的概念，掌握程序设计的一般思路和方法，培养学生独立解决问题的能力。

（3）以适用于初学者为目的进行编排。知识难度控制在初学者能接受的范围内，对于哪些内容可以了解、哪些必须掌握、哪些是较深入的应用等都给出了明确的说明。语法介绍简明扼要、条理清楚、例题丰富。程序例题尽量简单易学，以适合初学者。本书所有程序都运行通过。

本书由董晓华博士主编并负责全书的统稿工作，由姜雷和李崇担任副主编工作。参加本书编写的还有：何清太、林勇、罗琳、王志平，以及曲阜师范大学（日照校区）的张立华老师和中国人民解放军总医院第三零九临床部信息科的李泰环老师。

本书在编写过程中，参考了大量的国内外文献资料，在此，谨向文献资料的作者表示感谢。

本书配套有便于教学用的电子教案，并配有四套模拟试题，以及习题参考答案。由于篇幅有限，书中没有列出这些内容，以上所有内容可到http://www.bhp.com.cn网站下载，或向TextBooks@126.com发邮件索要以上内容。

欢迎读者和专家批评指正。

编 者

此为试读，需要完整PDF请访问：www.ertongbook.com

目　录

第 1 章　概论

随着计算机硬件的不断发展，计算机语言也在不断地更新换代，同时，人们在实践中也不断地总结出大量算法。本章对计算机系统的组成以及计算机软件技术进行了简要的介绍。

本章主要内容：

- 计算机系统的发展和软件的分类
- 计算机语言处理程序
- 计算机软件技术
- 软件开发的环境和方法

1.1　计算机系统

计算机系统是计算机硬件系统和软件系统的总称。计算机硬件系统是指计算机的硬件设施，它由五大部分构成即运算器、控制器、存储器、输入设备和输出设备。运算器和控制器组成中央处理器 CPU。存储器主要是用来存放数据和程序，它是由内部存储器（主存）和外部存储器（辅存）组成。常用的外部存储器有硬盘、软盘、光盘、磁带等。

而计算机软件系统又分成应用软件和系统软件两大部分。应用软件是根据计算机用户需要，专门用于解决某个或某类特定问题而编写的软件。它具有很强的针对性和实用性。随着计算机技术的发展和推广，各种应用软件在各自的行业应用领域发挥着越来越重要的作用。系统软件是主计算机系统的必备软件，它主要是用来管理、监控和维护计算机的软件及硬件资源。常用的系统软件有计算机操作系统、语言处理程序、数据库管理系统等。其中操作系统是系统软件中最为重要的。它是计算机硬件的第一级扩充，直接对计算机的硬件资源进行管理，其他的系统软件和所有应用软件都是建立在操作系统的基础之上的。它是用户和计算机之间的接口。目前使用最广泛的操作系统是 Windows 操作系统和 Unix 操作系统等。

1.1.1　计算机系统的发展

自从 1946 年 2 月世界上第一台计算机（ENIAC）在美国宾夕法尼亚大学问世以来，在短短的几十年里，计算机系统经历了巨大的变革。习惯上根据计算机系统所采用的硬件技术来划分计算机技术的发展阶段。

从 1946 年到 20 世纪 50 年代后期（1946～1957），计算机系统的主要元器件采用电子管，因此称为电子管计算机时期。其特点是体积大、功耗高、运算速度低。如 ENIAC 占地 170m^2，重达 30t，功耗为 140kW，有 18000 多个电子管，每秒钟能运行 5000 次加法运算。这一阶段，计算机主要用于军事、国防等尖端技术领域。在此期间，冯·诺依曼等人提出了存储程序的概念，为以后的计算机发展奠定了基石。IBM 公司 1954 年推出的 IBM650

是第一代计算机的代表。

从20世纪50年代后期到60年代中期（1957～1964），计算机系统所采用的主要元器件是晶体管，因此称为晶体管计算机时期。自从1947年晶体管在贝尔实验室诞生以后，引发了一场影响深远的电子革命。体积小、功耗低、价格便宜的晶体管代替了电子管，这不仅大大提高了计算机系统的性能，同时使计算机技术在科研、商业领域内广泛地应用。第二代计算机除主要元器件采用晶体管以外，其存储技术、操作系统及高级语言也随之出现，大大拓展了计算机的应用领域。这一时期计算机的主要代表有DEC公司1957年推出的PDP-I、IBM公司1962年推出的7094以及CDC公司1964年研制成功的CDC6600。1969年CDC公司研制的CDC7600平均速度达到每秒钟千万次浮点运算。

60年代中期集成电路的出现，宣告了第三代计算机时期的来临。因为第一代、第二代计算机均采用分离器件组成，成本高、体积大是不可避免的。采用集成电路之后，使得计算机的制造成本迅速下降，同时由于逻辑电路和存储器件的集成化，大大地提高了计算机的运算速度，功耗也随之下降。这一时期的计算机代表是IBM公司的system/360及DEC公司的PDP-8。这一时期为1965～1971年。

70年代初期到70年代后期（1972～1978），这一时期为大规模集成电路时期（LSI）。随着半导体存储技术的出现，迅速取代了磁心存储器，计算机的存储器向大容量、高速度的方向飞速发展。使得计算机集成电路的集成力度也越来越大，于是便出现了大规模集成电路。

随着就进入了超大规模集成电路的计算机时代。这一时期，软件和通信的重要性也逐渐上升，成为和硬件一样举足轻重的因素。同时计算机系统结构的不断改进对计算机系统的性能也产生了巨大的影响（中断系统、Cache存储器、流水线技术及RISC技术等等）。

计算机技术的发展日新月异，从单机系统到多处理器，从多处理器到分布式处理器。随着网络和通信技术的发展，计算机系统的应用领域也越来越广泛，目前，计算机已渗透到国防、教育、科研、商业、通信、金融、办公、自动化等领域。越来越成为与人们不可分割的辅助工具。

1.1.2 软件与软件的分类

1. 软件

计算机软件是指挥计算机工作的程序和程序运行时所需的数据，以及和这些程序和数据有关的文档。计算机软件是计算机系统的一个组成部分，早期的软件主要是指程序，程序的开发采用个体工作方式，效率低，规模小。随着计算机技术的飞速发展和应用领域的不断拓宽，软件在计算机系统中的地位越来越重要。软件所要求的功能及规模也越来越大。

2. 软件的分类

计算机软件根据其功能划分：

（1）系统软件：它主要是用来管理、监控和维护计算机的软件及硬件资源。主要有操作系统、数据管理软件及编译程序。

（2）支撑软件：是协助用户开发的工具软件。

（3）应用软件：是在特定领域里开发的，为特定目标服务的一类软件。现在，几乎所有领域都在使用计算机，因而为这些领域服务的软件种类繁多。

按软件的工作方式划分：

（1）实时处理软件：时实处理软件是在事件或数据产生时立即予以处理，并及时反馈信息，控制和监测处理及时的软件。一般包括采集、分析和输出三个部分。

（2）分时处理软件：分时处理软件是允许多个用户联机使用计算机，系统把处理机时间轮流地分配给各联机用户。

（3）交互式软件：交互式软件是指能实现人机通信的软件。这类软件能实现接收用户输入的信息，并将处理结果输出给用户。

（4）批处理软件：批处理软件是把一组输入的作业或数据成批地进行处理地方式一次运行，按顺序一个个地处理完这些作业。

按其规模划分：

（1）微型

（2）小型

（3）中型

（4）大型

（5）甚大型

（6）极大型

1.2　计算机语言和语言处理程序

程序设计语言分为两大类：低级语言和高级语言。

低级语言是每类计算机所固有的计算机能直接识别的语言。用机器语言编写程序需要对机器的结构有较多的了解，且编写的程序可读性差、不易理解和维护修改。为了提高编程效率，人们想到了用助记符来代替机器指令，于是就出现了汇编语言。然而汇编语言仍然是和机器语言十分接近的语言，离人们的惯常思维还是相差甚远。于是就出现了高级语言。而高级语言并非计算机能直接识别的。所以它需要某种能将高级语言所编写的程序翻译给计算机的中间处理程序。这就是所说的“语言处理程序”。目前的语言处理程序主要分为两类：编译程序和解释程序。

1.2.1　计算机语言概述

计算机语言分为低级语言和高级语言两大类。

低级语言又称面向机器的语言，是特定的计算机所固有的语言，用机器语言进行程序设计需要对机器有较多的了解。用机器语言写出的程序可读性差、程序难以修改和维护。为了提高程序设计的效率，便出现了汇编语言。虽然用汇编语言来编写程序，程序的编写效率和可读性都有所提高，但它仍然是一种和计算机机器语言十分接近的语言，它的书写格式在很大程度上取决于计算机的指令格式，它是一种低级语言。由于汇编语言仍未摆脱机器指令的束缚，对于人们的抽象思维和交流十分不便。于是高级语言产生了。

高级语言是以人们的思维方式作为背景，和人们的自然语言比较接近。用高级语言编写的程序可读性好，便于结构化开发，便于修改和维护，大大地提高了人们编程的效率。但今天的计算机仍然只能理解计算机机器语言，所以用高级语言编写的程序，只有翻译成机器语言才能被机器执行。完成这项任务的是编译程序和解释程序。

20 世纪 50 年代 FORTRAN 语言的推出，标志着高级语言的诞生。FORTRAN 语言作为国际上公认的第一种高级语言，确立了高级语言在计算机科学中的地位。随后出现的 ALGOL60 首次引进了许多现代程序设计语言的新概念。书写上的自由格式、系统预定义保留字、循环结构、分支结构、作用域规则、递归过程、值参数、换名参数及动态数组等具有严格的文法规则，首次引入分程序的概念。这些标志着高级语言已形成一门学科。

20 世纪 60 年代初，高级语言进入了突飞猛进的发展阶段。这一时期 FORTRAN 语言在数值计算领域，COBOL 语言在事务处理方面都确立了主导地位。同时也出现了继承和发扬 ALGOL60 风格的典型语言，ALGOL_W、Euler、SIMULA、PL/1 及 PASCAL 语言。ALGOL_W、Euler 引入了 CASE 语句、记录和指针等新概念。SIMULA 是第一代用于模拟领域的高级语言，它引入了 CLASS 的概念，是数据抽象的先驱。PL/1 引入了异常处理和多任务的概念。PASCAL 语言具有相当强的表达功能，其数据结构功能、尤其是用户自定义数据类型及该语言的公理化定义，使它在高级语言的领域具有相当大的吸引力。

20 世纪 70 年代，高级语言开始向纵深发展。在此期间，由于软件危机的出现，高级语言进入服务、适应“软件工程”并逐步走向成熟时期。其中，C 语言就是这个时期发展起来的一种通用程序设计语言。C 语言在很大程度上受 BCPL 和 B 语言的影响。C 语言以其简洁、灵活、统一风格及丰富的数据类型、指针及地址等优点很快地被广泛使用。著名的 UNIX 操作系统就是 C 语言的杰作。

20 世纪 80 年代以 ADA 为代表的强制式语言走向成熟。ADA 作为第一种军事程序设计语言，它把“软件工程”规范化贯穿始终。它不仅功能完善、应用面广而且在程序中处处强调可靠性、可维护性及可移植性。另一方面，面向知识处理的作用式语言得到了深入探讨和发展。逻辑语言（PROGOL）、函数语言（LISP、PF、ML）和数据流处理语言（VAL）就是作用式语言发展的结果。在此期间，随着面向对象的技术的发展，出现了面向对象的语言，典型代表有 SMALLTALK80 及后来的 C++。面向对象的程序设计的主要思想是：知识是对象的集合，对象是由数据及施加于其上的操作组成。对象间的通信采用消息传递机制，每个对象的具体实现相互独立，以便于模块化及信息隐蔽。对象的继承关系有利于知识抽取和重新利用。

随着程序设计语言的蓬勃发展，出现了许多新思想、新概念，某些研究方向代表了今后一段时期程序的设计语言的发展趋势。这一时期程序设计语言发展有三大特点：面向对象、网络化及可视化。可视化语言可谓独树一帜，其代表有微软公司的 Viusal Basic、Viusal C 及 Borland 公司的 Delphi。可视化程序设计是在综合了近年来软件工程及其相关技术的基础上发展起来的。如构件、控件技术代码自动实现等，都是软件复用技术及面向对象技术的体现。

1.2.2 计算机语言处理程序概述

计算机语言处理程序是将高级语言编写的程序翻译成计算机能执行的机器语言，主要包括汇编程序、编译程序和解释程序。

此为试读，需要完整PDF请访问：www.ertongbook.com

1. 机器语言

每种型号的计算机都有自己的指令集。集合中的每一条指令完成某一具体功能，这种机器指令集合就称为机器语言。早期的计算机是以“裸机”的面目出现在用户面前的。当要完成某种操作时，用户就用机器指令来编写指令序列。这就是机器语言程序。

2. 汇编程序

为了克服机器语言程序编写和调试上的困难，首先想到用助记符来代替指令编写程序，从而引入汇编的概念。

3. 编译程序

编译程序的职能是将用某些程序设计语言书写的程序翻译成与之等价的机器语言或汇编语言（目标程序），这里所说的等价是指目标程序执行原程序预定任务。编译程序的工作是相当复杂的。一般把编译程序分成词法分析、句法分析和语义分析、代码优化、代码生成、符号表管理等几个部分。

4. 解释程序

解释程序是语言处理程序的一种，它直接执行源程序或源程序的内部形式，因此，并不产生目标程序，这一点也是它和编译程序的主要区别。

1.3　计算机软件技术概述

随着计算机技术的飞速发展，特别是计算机硬件的迅猛发展，使得计算机软件日益落后。为了适应计算机硬件的发展和满足人们对软件的需求，在近几十年里，软件技术也得到了快速的发展。其主要体现在算法和数据结构上。

1.3.1　数据结构、算法和程序设计

1. 算法

算法是解决问题的精确描述，但并不是所有问题都有算法。

算法具有以下性质：

（1）由有限条可以机械执行的指令组成。

（2）有零个或多个输入。

（3）有一个或多个输出。

（4）解决问题的每一步都具有可描述性。

（5）解决问题的步骤是有限的。

（6）算法通常分为数值和非数值两大类。

2. 数据结构

数据结构是数据对象及其相互关系和构造方法。一个数据结构 B 可以形象地用一个二元组来表示：B=（A，R），其中 A 是数据结构中的数据（称为结点），R 是定义在 A 上的非空有限集合。结构是指结点之间的关系。所以数据结构也可以这样理解，即结点的有限集合和关系的有限集合。

数据结构按逻辑关系的不同分为线性数据结构和非线性数据结构。其中非线性数据结构又分为树形结构和图形结构。

算法和数据结构之间存在密切的关系。算法是建立在数据结构基础之上的。而数据结构构造方法也直接影响算法的好坏，二者相互依赖。

3. 程序

程序是对所要解决问题的各种对象和处理规则的描述，或者说是数据结构和算法的描述，因此，有人称：

数据结构+算法=程序。

4. 程序设计

程序设计就是设计、编制和调试程序的过程。也就是说，当问题分析之后，解决问题的全过程。

5. 结构化程序设计

结构化程序设计是利用逐步精化的方法，按一套 程式化的设计准则进行程序的设计。由这种方法产生程序结构良好，所谓“结构良好”是指：

（1）易于保证和验证其正确性。

（2）易于阅读、易于理解和易于维护。

1.3.2 操作系统和计算机网络

1. 操作系统

现代计算机系统由计算机硬件和软件两部分组成。硬件是构成计算机系统的物理设备。而软件则是指挥计算机按照某种流程来执行的一组指令。计算机软件可分为：系统软件和应用软件。没有安装任何软件的计算机称为“裸机”，让用户用“裸机”工作是很困难的。操作系统是硬件上建立的一个服务系统，是计算机硬件上的第一级扩充。

操作系统是最基本的系统软件，用户可以直接通过操作系统使用计算机。由此可以看出：

操作系统是一个大型的系统软件，是为了提高计算机系统资源的利用效率并方便用户使用计算机的一组程序，这些程序可以用软件实现也可以用硬件实现。

从用户的角度看，操作系统是用户与计算机之间的接口。

从资源管理的观点看，这里所说的资源是计算机系统进行数值运算所需要的硬件资源和软件资源。从这个角度上看，操作系统是管理和控制计算机资源的程序。

从进程的角度看，操作系统可以看成由若干个可以独立运行的程序组成。

从分层的观点看，操作系统通常分为：批处理操作系统，分时操作系统，时实操作系统。

随着网络技术的发展与普遍使用，计算机网络资源的共享使得网络环境下的操作系统也流行起来。网络环境下的操作系统分为：网络操作系统和分布式操作系统。网络操作系统是在各个单机操作系统的基础上，按照网络体系结构上的协议、标准进行开发，形成网络软件。它包括网络管理、通信、资源共享、系统安全系统等多种网络应用。而分布式操作系统是以安全系统的资源分配、调度为主为用户提供一个操作界面，用户据此使用系统资源，完成所有任务的一种操作系统。

2. 计算机网络

计算机网络是采用通信手段，将地理位置分散的、各自具备主动功能的若干台计算机有机地联接起来组成的一个复合系统，这个复合系统可以用来实现通信交往、资源共享和协同工作等目标，称这个系统为计算机网络。

（1）计算机网络的分类：

按地理位置划分为：局域网，城域网，广域网和互联网（Internet）。

按信息传输技术分为：广播式网络，点到点网络。

（2）计算机网络的基本组成：任何计算机网络都由计算机硬件、软件、通信设备和通信线路组成。

具体的看一个 LAN 大体上由三部分组成：

①计算机硬件：通常包括作为 LAN 站点的各台机器及其配置的各种外围设备。位于计算机内的网卡，可选的网络专用设备及传输介质。

②计算机软件：通常是指实现 LAN 功能用的“服务器软件”和“客户机软件”。当然也包括网络管理软件、通信协议等。

③网络资源与网络应用程序。

1.3.3 数据库管理系统

数据库管理系统 DBMS 是数据库系统的关键组成部分。任何数据操作，包括数据库定义、数据查询、数据维护、数据库运行控制等都是在 DBMS 的管理下进行的。DBMS 是用户与数据库的接口，应用程序只有通过 DBMS 才能和数据库打交道。

DBMS 具有相当强大的数据管理功能，主要体现在以下几个方面：

1. 数据库定义功能

DBMS 提供了数据描述语言 DDL（Data Description Language），来定义模式、外模式和内模式。并将名种模式翻译成相应的目标代码。这些目标代码并不是数据库中的数据，而是数据库的结构。翻译后的目标模式保存在系统的数据字典中。供 DBMS 进行数据管理时参照使用。

2. 映射功能

实现外模式/模式和模式/内模式的转换。

3. 数据库操作功能

DBMS 提供了数据库操纵语言 DML（Data Manipulation Language）实现对数据库的操作。DBMS 提供了四种操作命令：检索、插入、删除和修改。

4. 程序设计语言

任何 DBMS 均支持某种程序设计语言。有两种类型的程序设计语言：“宿主语言”和“自主语言”。

（1）宿主语言（Host Language）：用一般的程序设计语言（称为主语言，如 C、FORTRAN 等）编程，而把 DML 语言称为子语言，作为主语言的一种扩充嵌入到主语言中。

（2）自主语言（Self Contained Language）：DBMS 自含的程序设计语言，可以与 DML

有机的结合，也可以独立使用。Fox pro 就是这种语言。

5. 数据库运行控制功能

DBMS 对数据库的运行控制主要是通过数据的安全性、完整性、故障恢复和并发操作四个方面实现的。

（1）数据安全性控制：防止未被授权者非法存取数据库。采取的措施有鉴定用户身份、设置口令、控制用户存取权限及数据加密等。

（2）数据完整性控制：数据的完整性是指数据的正确性和相容性。DBMS 在建库时，把完整性作为模式的组成部分存入数据字典。数据字典 DD（Data Dictionary）中存放着数据库三级结构的描述以及各数据项的类型、值域和关键字等。从结构上对数据的定义和取值范围加以约束。

（3）故障恢复

数据库系统一旦投入运行，数据库中的数据时刻在发生着变化。然而许多因素可能使数据库遭到破坏。如磁盘损坏、病毒、操作失误等。作为 DBMS 必须有一定的数据恢复机制，把数据库从破坏的状态恢复到破坏前的状态。通常可以根据系统工作日志中记载的数据操作命令，逐步回退加以恢复。作为用户，则应注意随时转储数据库，时刻留有一份副本在手中。

（4）并发控制：在网络环境下，数据库一般由多用户共享。当碰巧多个用户同时操作同一数据时，即使其他方面没有任何问题，也有可能导致数据的正确性出错，这通常是由于两个进程之间不合理的时差造成的。例如，两个民航同时办理同一日期、同一航班、同一座位的机票，恰巧同时检索数据发现有票，于是就造成了把同一张机票就卖给了两个不同乘客的错误。DBMS 有这种并发的控制功能。例如通过“加锁”和“解锁”控制并发作业的进程，以确保数据的正确性。

6. 数据库维护功能

这部分包括数据库数据初始装入、数据库转储、数据库重组及记载系统工作日志等功能。这些功能大多由相应的实用程序来完成。

1.3.4 软件工程

软件工程学科是一门指导计算机软件开发和维护的工程学科。软件工程是一类求解软件的工程。它应用计算机科学、数学及管理科学等原理，借鉴传统工程的原则、方法，创建软件以达到提高质量，降低成本的目的。其中，计算机科学、数学用于构造模型与算法，工程科学用于制定规范设计范型、评估成本及确定权衡，管理科学用于计划、资源、质量、成本等管理。

软件工程的方法、工具、过程构成了软件工程的三要素。

软件工程的目标可概括为：在给定成本、进度的前提下，开发出具有可修改性、有效性、可靠性、可理解性、可维护性、可重用性、可适应性、可移植性、可追踪性和可互操作性并满足用户需要的软件产品。

软件生命周期表明软件从功能确定、设计，到开发成功投入使用，并在使用中不断地修改、增补和完善，直至被新的需要所替代而停止该软件的使用的全过程。

软件开发模型是从软件项目需求定义直至软件经使用后废弃为止，跨越整个生存期的系统开发、运作和维护所实施的全部过程、活动和任务的结构框架。

1.3.5　面向对象方法学

面向对象技术是软件技术的一次革命，在软件开发史上具有里程碑的意义。

随着 OOP（面向对象编程）向 OOD（面向对象设计）和 OOA（面向对象分析）的发展，最终形成面向对象的软件开发方法 OMT（Object Modelling Technique）。这是一种自底向上和自顶向下相结合的方法，而且它以对象建模为基础，从而不仅考虑了输入、输出数据结构，实际上也包含了所有对象的数据结构。所以 OMT 彻底实现了 PAM 没有完全实现的目标。不仅如此，OO 技术在需求分析、可维护性和可靠性这三个软件开发的关键环节和质量指标上有了实质性的突破，彻底地解决了在这些方面存在的严重问题，从而宣告了软件危机末日的来临。

1. 自底向上的归纳

OMT 的第一步是从问题的陈述入手，构造系统模型。从真实系统导出类的体系，即对象模型包括类的属性，与子类、父类的继承关系，以及类之间的关联。类是具有相似属性和行为的一组具体实例（客观对象）的抽象，父类是若干子类的归纳。因此这是一种自底向上的归纳过程。在自底向上的归纳过程中，为使子类能更合理地继承父类的属性和行为，可能需要自顶向下的修改，从而使整个类体系更加合理。由于这种类体系的构造是从具体到抽象，再从抽象到具体，符合人类的思维规律，因此能更快、更方便地完成任务。这与自顶向下的 Yourdon 方法构成鲜明的对照。在 Yourdon 方法中构造系统模型是最困难的一步，因为自顶向下的“顶”是一个空中楼阁，缺乏坚实的基础，而且功能分解有相当大的任意性，因此需要开发人员有丰富的软件开发经验。而在 OTM 中这一工作可由一般开发人员较快地完成。在对象模型建立后，很容易在这一基础上再导出动态模型和功能模型。这三个模型一起构成要求解的系统模型。

2. 自顶向下的分解

系统模型建立后的工作就是分解。与 Yourdon 方法按功能分解不同，在 OMT 中通常按服务（Service）来分解。服务是具有共同目标的相关功能的集合，如 I / O 处理、图形处理等。这一步的分解通常很明确，而这些子系统的进一步分解因有较具体的系统模型为依据，也相对容易。所以 OMT 也具有自顶向下方法的优点，即能有效地控制模块的复杂性，同时避免了 Yourdon 方法中功能分解的困难和不确定性。

3. OMT 的基础是对象模型

每个对象类由数据结构（属性）和操作（行为）组成，有关的所有数据结构（包括输入、输出数据结构）都成了软件开发的依据。因此 Jackson 方法和 PAM 中输入、输出数据结构与整个系统之间的鸿沟在 OMT 中不再存在。OMT 不仅具有 Jackson 方法和 PAM 的优点，而且可以应用于大型系统。更重要的是，在 Jackson 方法和 PAM 方法中，当它们的出发点——输入、输出数据结构（即系统的边界）发生变化时，整个软件必须推倒重来。但在 OMT 中系统边界的改变只是增加或减少一些对象而已，整个系统改动极小。

4. 需求分析彻底

需求分析不彻底是软件失败的主要原因之一。即使在目前，这一危险依然存在。传统的软件开发方法不允许在开发过程中用户的需求发生变化，从而导致种种问题。正是由于这一原因，人们提出了原型化方法，推出探索原型、实验原型和进化原型，积极鼓励用户改进需求。在每次改进需求后又形成新的进化原型供用户试用，直到用户基本满意，大大提高了软件的成功率。但是它要求软件开发人员能迅速生成这些原型，这就要求有自动生成代码的工具的支持。

OMT 彻底解决了这一问题。因为需求分析过程已与系统模型的形成过程一致，开发人员与用户的讨论是从用户熟悉的具体实例（实体）开始的。开发人员必须搞清现实系统才能导出系统模型，这就使用户与开发人员之间有了共同的语言，避免了传统需求分析中可能产生的种种问题。

5. 可维护性大大改善

在 OMT 之前的软件开发方法都是基于功能分解的。尽管软件工程学在可维护方面作出了极大的努力，使软件的可维护性有较大的改进。但从本质上讲，基于功能分解的软件是不易维护的。因为功能一旦有变化都会使开发的软件系统产生较大的变化，甚至推倒重来。更严重的是，在这种软件系统中，修改是困难的。由于种种原因，即使是微小的修改也可能引入新的错误。所以传统开发方法很可能会引起软件成本增长失控、软件质量得不到保证等一系列严重问题。正是 OMT 才使软件的可维护性有了质的改善。

OMT 的基础是目标系统的对象模型，而不是功能的分解。功能是对象的使用，它依赖于应用的细节，并在开发过程中不断变化。由于对象是客观存在的，因此当需求变化时对象的性质要比对象的使用更为稳定，从而使建立在对象结构上的软件系统也更为稳定。

更重要的是 OMT 彻底解决了软件的可维护性。在 OO 语言中，子类不仅可以继承父类的属性和行为，而且也可以重载父类的某个行为（虚函数）。利用这一特点，我们可以方便地进行功能修改：引入某类的一个子类，对要修改的一些行为（即虚函数或虚方法）进行重载，也就是对它们重新定义。由于不再在原来的程序模块中引入修改，所以彻底解决了软件的可修改性，从而也彻底解决了软件的可维护性。OO 技术还提高了软件的可靠性和健壮性。

1.4 软件开发环境

软件开发环境是支持软件产品开发的软件系统，它由软件工具集和环境集成机制构成。前者用以支持软件开发的相关过程、活动和任务。而后者为工具集成和软件开发。

1.4.1 软件开发方法

20 世纪 60 年代中期开始爆发了众所周知的软件危机。为了克服这一危机，在 1968、1969 年连续召开的两次著名的 NATO 会议，会议上提出了软件工程这一术语，并在以后不断发展、完善。与此同时，软件研究人员也在不断探索新的软件开发方法。至今已从结构化程序设计方法逐步走向面向对象的程序设计方法。

此为试读，需要完整PDF请访问：www.ertongbook.com