

计 算 机 基 础 教 育 丛 书



谭浩强 主编

大学计算机软件 技术基础教程

(理工类)

高等院校非计算机专业教学用书

王启智 佟建新 毛汉书
雷光复 王 辉 王 利 编著

科 学 出 版 社
龍 門 書 局



计 算 机 基 础 教 育 丛 书



谭浩强 主编

大学计算机软件 技术基础教程

(理工类)

高等院校非计算机专业教学用书

王启智 佟建新 毛汉书 编著
雷光复 王 辉 王 利



科 学 出 版 社
龍 門 書 局

内 容 简 介

本书是基于高等院校计算机基础课程 21 世纪教育改革方案的思想而编写的。该方案将高校计算机基础教育分为三个层次：公共基础层（面向各专业）、专业技术基础层和结合专业的专业技术层。在第二层中分硬件技术基础和软件技术基础两门课程。本书即为第二层软件技术基础课程教材。

本书包括 6 章。第一章介绍计算机软件系统的基本组成和基本内容；第二章介绍算法和数据结构；第三章介绍操作系统与操作环境；第四章介绍软件开发技术，着重讲述面向对象软件开发技术和网络环境下的软件开发；第五章介绍软件工程基础；第六章介绍数据库系统。

本书是高等院校计算机系教授结合长期的教学经验编写而成，内容全面，叙述清楚，既满足高校教学大纲的要求，又介绍了前沿的新技术以帮助读者赶上软件技术发展的步伐。本书采用大量的图例和代码实例进行说明，易于理解，可操作性强，便于读者动手实践。

本书可作为高等院校非计算机专业计算机软件技术基础课程教材，也可作为计算机软件技术基础知识培训的通用教材。本书可供高校非计算机专业学生、教师使用。

需要本书或需要得到技术支持的读者，请与北京海淀 083 信箱北京希望电子出版社（邮编 100080）联系。
网址：www.bhp.com.cn, E-mail: qrh@hope.com.cn. 电话：010-62562329, 62541992, 62637102, 010-62633308, 62633309（发行、技术支持）；010-62613322-215（门市）；010-62531267（编辑部）。传真：010-62579874。

计算机基础教育丛书 大学计算机软件技术基础教程

（理工类）

谭浩强 主编

王启智 佟建新 毛汉书 雷光复 王辉 王利 编著

责任编辑：陈河南

北京东黄城根北街 16 号

科学出版社出版
龙 门 书 局

邮政编码 100717

北京媛明印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

2000 年 12 月北京第 1 版 开本：787×1092 1/16

2000 年 12 月第 1 次印刷 印张：19 1/8

印数：1~5000 册 字数：441,636

ISBN7-03-006304-X/TP·858

定价：26.00 元

前 言

《计算机软件技术基础》是国家教育部高教司和全国高等院校计算机基础教育研究会（以下简称研究会）关于高等院校非计算机专业计算机基础教育第二层次——计算机技术基础教育中的一门重要课程。《大学计算机软件技术基础教程》就是为此课程而编写的。

《大学计算机软件技术基础教程》是根据研究会课程建设委员会提出的适用于理工类的计算机基础教育改革方案所组织的一套教材中的最后一本（该套教材一共四本，其中《大学计算机公共基础实用教程》、《大学计算机公共基础实验教程》和《大学计算机硬件基础技术教程》已先后于1999年10月、2000年2月和3月出版）。

《大学计算机软件技术基础教程》一书定位在非计算机专业理工科专业大学普通本科生和高职相应课程。目标是使学生在了解软件技术基础中基本理论和基本技术的基础上，初步掌握一般应用软件的开发知识和开发能力。

当前，在有些高等院校的学生中，有一种倾向：摸得着、看得见的计算机硬件，学生不愿意学；而摸不着、看不见的计算机软件，学生又没信心学。计算机教育的要求是“软硬兼备”，但结果却是“软不成，硬不就”！软件技术基础课的确是一门讲究抽象思维的课，相对来说要稍微难一些，但只要破除神秘感，软件技术不是不可以掌握的。

《大学计算机软件技术基础教程》一书所介绍的内容都是理工科本科生必备的软件技术基础知识和基本能力。软件技术基础的重心是软件开发技术。对于大多数学生来说，要初步掌握先进的软件开发技术，就必须在掌握1到2种程序设计语言（如C、C++）的基础上，培养和建立较强的算法意识和算法思维能力；了解和掌握各种数据结构的特点，并能把它们较好地与程序设计紧密结合；能够选择或创建最佳软件开发环境和开发平台，运用先进的软件开发方法和工具，开发出一般应用软件和与其配套的数据库。再高一点的要求，即在初步了解和运用软件工程规范的基础上具有开发小规模应用软件系统的能力。

《大学计算机软件技术基础教程》用于大三相应课程，建议用72学时，课时、机时比为56:16。在16机时中，建议安排一个课程设计：在Windows X或Linux的开发平台上，用面向对象程序设计语言（C++）和面向对象软件开发技术，开发并测试一适当规模的应用软件。

本书在介绍软件技术基础中基本概念和基本技术的基础上，强调软件技术的实用性和实践性。故本书既可以用作理工科机电专业普通本科生和高职本科生相应课程的基本教材，也可以作为计算机工程技术人员的参考书和自选教材。

本套丛书由研究会理事长谭浩强教授担任主编，并以“浩强创作室”的名义出版。

北京联合大学副校长、研究会副理事长高林教授担任本书第一主审（第一、四、五章），王启智教授策划、组织并担任第二主审（第二、三、六章及全书的统稿和校订）。

第一章由王启智教授编写，第二章由佟建新副教授编写，第三张由毛汉书教授编写，第四章由雷光复副教授编写，第五章由王辉老师编写，第六章由王利教授编写。

本书在编写和出版过程中得到许多老师的热情支持和帮助，特别是得到北京希望电子出版社刘晓融副社长给予的大力协助，在此对他们一并表示诚挚的谢意！

由于时间比较仓促，无论在书的选材、定题上，还是在概念、理论和技术的叙述上均有可能出现纰漏，甚至谬误，敬请读者不吝赐教。

编著者谨启 2000.5.1

目 录

第1章 计算机软件技术基础要论..... 1	4 软件开发技术..... 115
1.1 计算机系统和计算机软件系统..... 1	4.1 传统的软件开发技术..... 115
1.2 计算机软件技术基础的基本要求 和内容..... 2	4.2 面向对象软件开发技术..... 121
习题和思考题..... 14	4.3 网络环境下的软件开发..... 155
参考文献..... 14	4.4 可视化技术及其应用..... 176
2 算法与数据结构..... 15	习题和思考题..... 190
2.1 算法与算法的设计方法..... 15	参考文献..... 190
2.2 数据结构的基本概念..... 20	5 软件工程基础..... 191
2.3 线性数据结构及应用..... 22	5.1 软件工程的产生..... 191
2.4 树..... 37	5.2 系统分析与可行性研究..... 199
2.5 图..... 44	5.3 软件需求分析..... 201
2.6 数据集合及查找方法..... 50	5.4 系统介绍页面..... 215
2.7 简单排序..... 57	5.5 软件开发方法..... 216
习题和思考题..... 61	5.6 软件测试与维护..... 222
参考文献..... 62	5.7 软件工程新技术..... 225
3 操作环境与操作系统..... 63	习题和思考题..... 231
3.1 操作系统的运行环境..... 63	参考文献..... 232
3.2 操作系统基本概念..... 68	第6章 数据库系统..... 233
3.3 进程管理..... 70	6.1 数据库系统基础知识..... 233
3.4 存储管理..... 80	6.2 数据管理与数据模型..... 237
3.5 文件系统的组织、管理与使用..... 83	6.3 关系数据库..... 241
3.6 设备管理的基本功能..... 88	6.4 结构化查询语言——SQL..... 248
3.7 网络..... 91	6.5 关系规范化基础..... 264
3.8 操作系统安全..... 95	6.6 数据库设计..... 279
3.9 Linux 程序设计..... 100	6.7 面向对象数据库..... 288
习题和思考题..... 114	习题和思考题..... 297
参考文献..... 114	参考文献..... 297

信息技术的迅速发展使人类开始步入信息社会和知识经济时代，它的特征是以科学技术的研究开发为发展基础、以知识的传播和处理为中心任务、以智力资源和无形资产为资源配置的首要因素。在信息技术中，微电子技术是基础，计算机和通信设施是载体，软件技术是核心。没有软件，就没有信息化社会。软件是知识的集成，是知识的固化，是知识的结晶。软件的开发离不开知识，反之，知识的处理、传播也离不开软件。

半个世纪以来，计算机技术的发展总是依赖于计算机硬件技术和计算机软件技术的发展而发展，计算机软件是计算机应用的灵魂。特别是近 20 年来，计算机技术的日新月异更依赖于计算机软件技术的长足进展和飞速更新，在可预见的未来，这种发展趋势将更加光灿夺目。

从国际软件技术的发展趋势看，软件工程技术日益普及，软件开发环境、开发平台和开发工具将被普遍采用。软件复用中的软件构件技术受到广泛关注。网络、分布计算、Internet 技术等将成为软件技术中的主要技术热点。过程改善+构件技术+CASE（计算机辅助软件工程）环境正成为软件工程的主流发展趋势。

计算机软件产业是高新技术的重要组成部分，没有先进的软件产业，就不可能有先进的信息技术产业。自 1995 年以来，软件产业正逐步进入以软件过程、面向对象和构件重用等三项技术为基础的软件工业化生产时代。

我国的软件产业，除汉字激光照排、财务、教育和中文字处理等软件外，其他几乎都是境外软件占据着我国大部分市场。尽管我国的教育还不发达，但中国每年培养的软件人才的绝对数在世界是名列前茅的。目前，我国软件产业中的专业开发人员大约有 10 余万人，而各行各业中计算机应用部门从事软件开发的人员大约有 40 多万人，比例为 1:4，这个比例还要扩大，因为，在各行各业从事计算机应用软件开发的人员将越来越多。我们是从事非计算机专业计算机基础教育的学校，从这里毕业的学生将面临着各种实际应用，他们在学校学的计算机软件技术基础知识和技术将大有用武之地。

目前世界最大的软件公司几乎都在美国，这些公司的产品几乎占据了世界软件市场的半壁江山。在这些大公司中，聚集着相当多的中国人，几乎每个软件产品都包含着中国人的智慧。有人作过调查，在世界上无论是由中国人还是由外国人投资的软件企业里，中国工程师都占有相当的比例，在美国硅谷的企业里如果没有华人，那简直是不可想象的。这说明中国具有大批软件人才，中国人适合生产软件产品。

1.1 计算机系统和计算机软件系统

(1) 计算机系统的基本组成。

计算机系统（这里主要介绍 PC 机系统）包括计算机软件系统和计算机硬件系统。在《大学计算机硬件技术基础教程》一书中，我们对计算机的硬件系统已经作了比较详细的介绍，这里简单介绍一下计算机软件系统的基本组成，请看图 1.1。

计算机软件系统包括系统软件和应用软件，系统软件包括操作系统、程序设计语言处理程序等，应用软件包括各种专用软件。

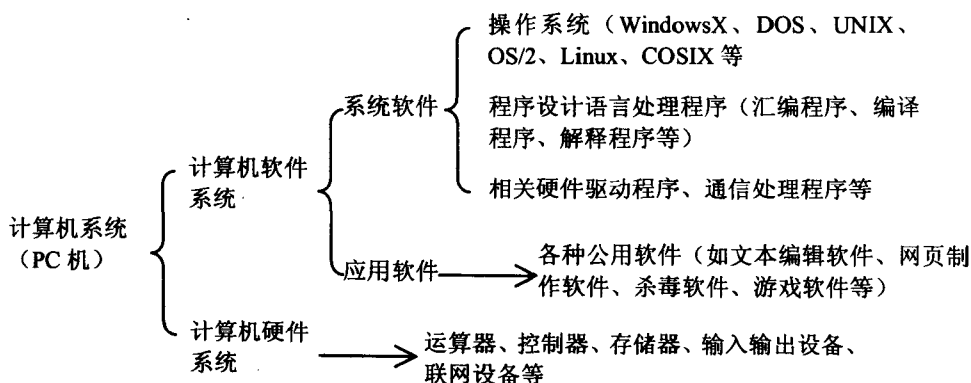


图 1.1 计算机系统的基本组成

(2) 计算机软件系统与计算机硬件系统、用户之间的层次关系如图 1.2 所示。

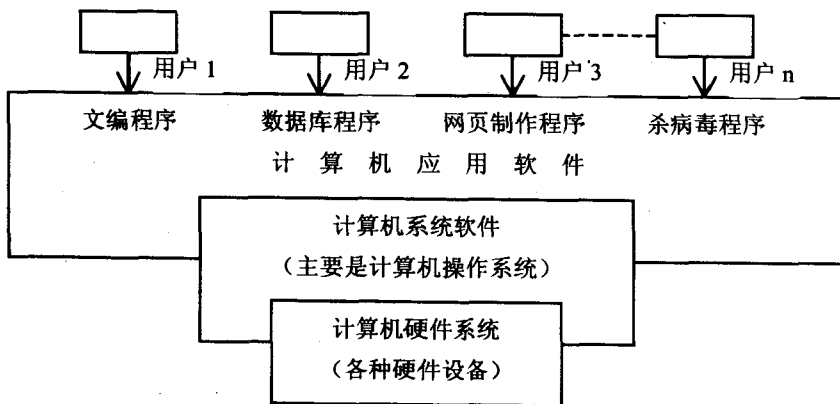


图 1.2 计算机软件系统与用户、硬件系统之间的关系

本书的目的在于使理工类大学机电专业本科生能在特定的软件开发环境中，在某种操作平台上，利用现成的软件开发工具和应用程序，按照软件开发的规范和软件工程要求，初步掌握开发适合本专业和实际工作需求的软件的能力。

1.2 计算机软件技术基础的基本要求和内容

作为理工类机电专业大学本科生，必须具备根据专业的需要，开发应用软件的能力。为此，在掌握并能用 1 至 2 种程序设计语言编写数百条语句程序的基础上，对算法和数据结构、软件开发环境、软件开发平台、软件开发技术、软件工程、数据库等软件技术基础知识有比较深入的了解，并初步掌握一般小规模应用软件的开发、测试。对于少数基础好、算法意识强、编程能力强的学生，要初步了解和掌握一些与具体应用无关的软件设计技术——软件工程方法学，即为了保持和延长“软件生命周期”所要完成的软件工程中计划阶段、开发阶段和维护阶段的例行任务。

1.2.1 程序设计语言与语言实现

1. 程序设计语言 (programming language)

程序设计语言从问世到现在已经有近半个世纪的历史。它经历了由机器语言 (machine language) 发展到符号化了的机器语言, 即用助记符 (mnemonic) 表示的汇编语言 (assembler language), 它们都是面向机器的语言 (machine-oriented language)。后来又发展成面向过程的语言 (process-oriented language), 进而从模块化、结构化发展成面向对象的语言 (object-oriented language)。这些变化反映了语言由低级向高级、由过程性向非过程性的发展。语言的应用范围也从单处理机发展到异构的、物理上分布式的计算机网络, 从单纯的科学计算发展到包括过程控制、信息处理、事务处理等各个应用领域。语言在软件开发过程中的作用也从单纯的程序设计语言发展到可用于软件开发过程中的各个阶段、各个方面。因此可以说, 当今世界上计算机无处不在, 计算机程序设计语言也无处不在、无时不在、无事不在。

至今为止 PC 机的历史差不多二十年, 现在不少程序设计语言已移植到 PC 机上。很多常用的程序设计语言相继标准化, 如 Ada 95、C++ 和 Fortran 90/95。在科学计算方面, Fortran 语言仍是主流语言, 该语言结构简单高效, 它的 Fortran 90 版增加了大量的现代数据和控制特点, 具有 Ada、C 和 Pascal 等语言的能力, 其修订版是 Fortran 95。

对于许多要求高可靠性和高安全性的应用, Ada 仍是被选用的一种语言。C++ 的使用迅速扩展和流行。但最引人注目的是 Java 语言, 它已从一种编程语言发展成为网络时代的一种通用计算机平台。1998 年底, Sun 公司继推出 Java 2 (包括适用于不同对象的速度已接近 C++ 的三个版本) 之后, 又于 1999 年初推出了革命性的新语言 Jini。

2. 语言实现 (language realization)

语言实现实际上研究的是编译程序的构造原理和技术。图 1.3 就是一个典型的语言实现结构。

Java 语言是面向 21 世纪的新技术, 它将改变传统的计算模式。Java 语言实现结构的编译系统经历了三代的改进, 由第一代的传统编译发展到第二代的及时编译, 再到第三代的动态编译, 即将及时编译与解释相结合。第一代 Java 的编译系统如图 1.4 所示。

编译型语言有 C、C++、Fortran、Pascal 和 Ada 等, 用此类语言编写的源程序在执行前需要经过编译程序 (compiler) 直接转变为目标机器语言。

解释型语言有 BASIC、LISP、ML、Prolog 和 Smalltalk 等, 用此类语言编写的源程序在执行前需要先用汇编程序 (assembler) 转变为易于执行的中间代码, 然后再由解释程序 (interpreter) 把中间代码转变为目标机器代码。

选择什么语言主要取决于待解决问题的应用领域。表 1.1 为各应用领域使用的语言。

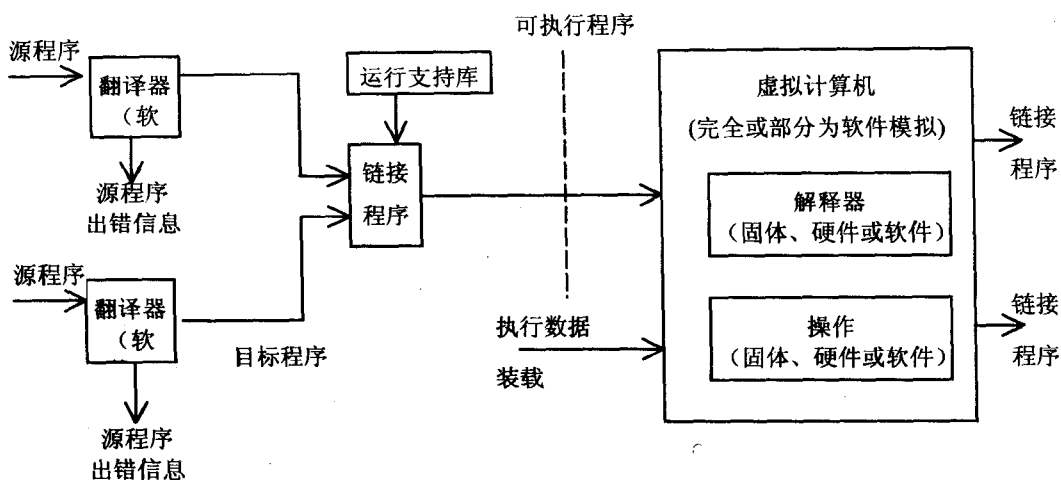


图 1.3 一个典型的语言实现结构

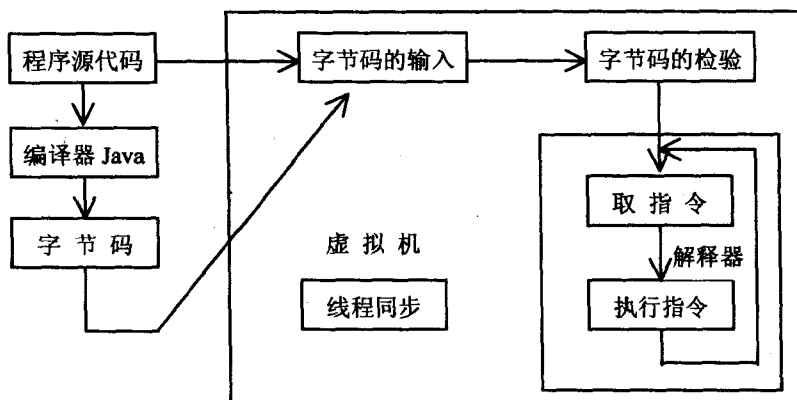


图 1.4 第一代 Java 编译系统

表 1.1 各应用领域的选用语言

年代	应用领域	主要语言	其它语言
六十年代	商业	COBOL	Assembler
	科学	FORTRAN	ALGOL,BASIC,APL
	系统	Assembler	JOVIAL,Forth
	人工智能	LISP	SNOBOL
今日	商业	COBOL,Spreadsheet	C,PL/I,4GLs
	科学	FORTRAN,C,C++	BASIC,Pascal
	系统	C,C++	Pascal,Ada,BASIC,Modula
	人工智能	LISP,Rrolog	
	出版业	Tex,Postscript,word processing	Rocessing
	处理	UNIX shell,TCL,PERL	Marvel
	新范例	ML,Smalltalk	Eiffel

1.2.2 算法与数据结构是软件开发技术中的核心

1. 算法 (algorithm) 的重要性

电脑发展到今天,能有如此神通广大的功能,除了集成电路芯片制造工艺的飞速提高及芯片内体系结构的不断改进外,主要靠软件的进步,而软件的核心是算法,算法中蕴涵的设计思想是灵魂。无论是制造芯片的 EDA (Electronic Design Automation, 电子设计自动化) 软件、多媒体信息的压缩技术 MPEG (Motion Picture Experts Group, 运动图象专家组), 还是“深蓝”(deep blue) 电脑赢了卡斯帕罗夫,亦或是多国部队在海湾“沙漠风暴”中所进行的一场胜者流血最少的战争,算法都起了关键作用。北大方正“华光系统”的核心是王选院士在 70 年代中期创立的算法,方正开发的彩排系统关键也在算法。

国家 973 信息技术与高性能软件工程项目首席科学家顾钧教授认为:未来计算机的竞争将从芯片性能的竞争转向算法的竞争。一个好的计算方法和程序设计有时会取代硬件的升级,一个好的算法可以带动甚至创造一个产业。

算法是解决某一特定类型问题的有限运算序列,描述一个算法可以采用某一种计算机程序设计语言,也可以采用程序设计中的流程图,算法是一种思想,而不是编程的技巧。

算法所具有的 5 个基本特性是有穷性、确定性、可行性、输入和输出。评价一个算法是否优劣主要看计算是否正确和计算复杂性 (computing complexity), 通俗地说就是计算机求解问题的难易程度。衡量复杂性的标准,一是计算所需的步骤或指令条数,即运行的时间,称时间复杂度;二是计算所需的存储单元数量,即占用的存储空间,称空间复杂度。另外,还要看算法是否简单和是否易读。

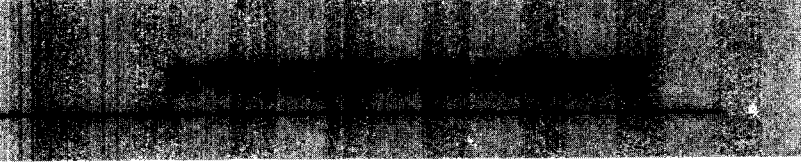
一个好的编程人员,主要不是看他是不是一个“编程快手和能手”,主要看他是否有强烈的算法意识、细腻而又科学的算法思维、敏捷而又扎实的程序设计思想。显然,能否熟练而又有效地掌握软件开发技术,很重要的一条就是培养算法意识,使算法所具有的 5 个特性和 4 个评价标准都最大限度地体现在软件的开发和运行当中。软件开发并不只是编写程序,编程不过是软件最后的实现形式,软件的核心是解决算法问题。

2. 数据结构 (data structure) 在程序设计中的作用

数据是信息的载体,是所有输入到计算机并由计算机程序进行处理的符号的总称。计算机的应用已不限于科学计算,更多的是用于控制、管理和数据处理。计算机加工处理的数据已不限于纯粹的数值,更多的是符号、表格、图形和图像。由于信息的类型不一样,信息载体的数据结构也不一样。数据结构就是研究数据组织、存储和运算的一般方法的一门学问。

数据结构是指同一数据元素类中各数据元素之间存在的结构关系,包括它们之间的逻辑结构关系、存储结构关系和运算关系。数据的逻辑结构是从逻辑上观察数据并以此来描述数据之间的关系;数据的存储结构是数据的逻辑结构在计算机存储器中的实现。逻辑结构与数据的存储无关,是独立于计算机的,而存储结构是依附于计算机的。数据的运算是在数据的逻辑结构上定义的操作算法。

1976 年,沃思 (worth) 提出了“算法 + 数据结构 = 程序设计”的论断,阐明算法与数据结构在程序设计中的重要作用。从该论断可进一步看出程序设计的实质:程序设计



实际上就是构思和设计解题的思路和步骤，对给定的包括各种结构的数据进行有效的处理。构思和设计解题思路、步骤是算法所研究的内容。对各种结构的数据进行合理的组织、存储和运算是数据结构研究的内容。在程序设计中选择合适的数据结构，可以起到简化程序的算法，节约占用存储空间和提高操作效率的作用。

1.2.3 软件开发环境与开发平台的走向开放

搞好软件的开发，除了要有好的算法、构思和设计解题的思路和步骤、对给定的各种结构的数据进行有效的处理外，还要有良好的软件开发环境。良好的软件开发环境主要包括对开发者友好的“用户界面”(user interface)和功能强大的为开发者提供各种有用开发工具的“操作平台”(operating platform)。

1. 友好的用户界面

在软件开发环境中，用户界面占有重要的地位。在计算机应用中，用户与计算机都要通过人-机界面把信息传递给对方，即所谓人机交互(human_machine interaction)。所谓界面友好，从广义上说，主要指计算机的显示器、键盘、鼠标等输入/输出设备和计算机的系统软件、装载的应用软件是否对“用户友好”(user friendly)。就是说这些硬、软件的外观上是否美观清晰，感觉上是否舒适愉快，使用上是否方便简单，性能上是否可靠稳定。从狭义上说，人们往往把软件在显示器屏幕上体现的用户界面是否灵活方便、鲜明简捷、赏心悦目、兴味索然，这就叫作界面友好。

随着计算机技术的发展和用户对计算机要求的提高，用户界面也在不断发展。80年代以前，用户界面主要是基于字符(或行)的界面，即字符用户界面 CUI (Character User Interface)。80年代出现了图形用户界面，即 GUI (Graphic User Interface)。进入90年代，图形用户界面得到长足进展和广泛推广，特别是随着多媒体技术的发展，图文并茂、声形兼备的视窗用户界面精彩纷呈，用户既可以使用传统的字符、菜单命令、联机帮助，也可以使用多窗口的图形、图像、声音等媒体与计算机交互，使计算机的操作和应用更加丰富多彩、方便怡人。

Internet 时代的到来，使用户界面又将发生变化，图形用户界面(GUI)将可能发展到浏览器用户界面 BUI (Browser User Interface)。

2. 操作系统(operating system)走向开放性

任何类型的计算机都必须有以操作系统为主的系统软件。操作系统是软件开发人员进行软件开发的一个平台，本书主要介绍 PC 机的软件开发平台。

当前国际上的主流操作系统有三大类：一是微软的操作系统(主要是 Windows 9X)系列；二是一些 IT 产业为自己硬件产品提供的配套专用操作系统，如 DEC 公司的 VMS、IBM 公司的 OS/2、Apple 公司的 MacOS 等；三是对外开放的操作系统，如 UNIX、Linux 等。

尽管微软的 Windows 9X 系列仍主导 OS 市场，保持常盛不衰，操作系统的功能，特别是他们的网络功能(如 Windows 2000)日益强大，但仍然存在与以网络为中心的环境不相适应、OS 内部通信、硬件和软件都抢着控制 CPU 的时间和资源的分配等问题。这

些问题都在不同程度上导致系统的异常和出现错误。21 世纪的网络用户需要一个更加智能、快速和多任务的操作系统，使它能稳定可靠地传输异步、非预定制和实时数据流。

从下面表 1.2 各主要操作系统三年来的销售量比较可看出桌面操作系统的兴衰。

表 1.2 操作系统销量比较

单位: 1000 套

操作系统	1997	1998	1999	2000
DOS (Without Windows)	1478.9	872.6	514.8	303.7
UNIX	1496.0	1744.4	2151.8	2575.1
Windows 3.1(with DOS)	7849.9	1484.2	536.1	0.0
Mac OS	2988.0	2932.7	2892.4	2903.5
Windows 95	31321.2	0.0	0.0	0.0
OS/2 Warp	1256.5	759.8	480.1	60.2
Windows NT	72041	13954.4	25846.0	43243.2
Windows 98	0.0	56683.8	95539.4	99319.8
Proprietary and others	369.1	343.9	319.1	294.9
总量	93663.6	110601.1	128870.3	149375.5

尽管该表的数据并不充分,缺少 2000 年 Windows2000、Linux 等产品销量记录,但仍可看出:绝大多数用户 2000 年还将把微软各种版本的 Windows 作为他们首选的桌面操作系统。

1990 年,芬兰赫尔辛基大学的一个学生 Linus Torvalds 在学习 UNIX 课程时,上机要排队, Linus 就自己买了一台 PC 机,安装了 Andy 教授自己利用业余时间开发的用于教学的操作系统 Minix,它是一个基于微内核技术的类似于 Unix 的操作系统。在使用过程中, Linus 受 Minix 的启发,用 C 语言编写代码。

Linux 为何倍受青睐?一是因为它免费提供源代码;二是因为爱好者可以按照自己的需要自由修改、复制和发布程序的源代码,并公布在 Internet 上。于是,世界各地的许多操作系统高手为 Linux 编写了大量的驱动程序和应用软件,使 Linux 成为一种包括操作系统内核、系统管理工具、完整的开发环境和开发工具、应用软件在内的用户很容易得到的操作系统。

Linux 可以说是一个运行在 PC 机上的 UNIX 系统,所以,UNIX 上的应用可以很方便地移植到 Linux 平台,使 UNIX 用户很容易掌握 Linux,特别是由于 Linux 的低廉价格而被中小企业广泛应用。

Linux 还支持一系列 UNIX 开发工具, C、C++、FORTRAN、PASCAL 等程序设计语言都移植到 Linux 平台上。

越来越多的软、硬件厂商,如 Intel、IBM、Compaq、Dell、HP 等都纷纷宣布对 Linux 表示支持。IBM、Oracle、Informix、Sybase 相继推出了数据库的 Linux 版本。我国非常重视 Linux 技术的研究,把它看成发展我国自主操作系统和民族软件产业的一个难得的契机。从事 Linux 开发的公司越来越多, Linux 中文平台如雨后春笋般地争先出台,如 Turbo Linux、Xteam Linux、Blue Point Linux、Redhat Linux、红旗 Linux、Lindows 等。

除了抓住 Linux 这个开放式操作系统的机遇外,我国还决定开发开放式操作系统

UNIX。中软公司自推出两代自主版权的 32 位的 UNIX 操作系统——COSIX 后，又于 1998 年 9 月与康柏公司签订了联合开发 64 位操作系统 COSIX64 的协议，协议规定将康柏 Tru 64 的源代码完全向中软开放，由中软加入中文、安全等增值工作，使原来基于 Tur64 的应用程序都可以移植到 COSIX64 上，从而到 1999 年 11 月，开发出了我国最先进的 64 位操作系统 COSIX64 V5.0。它的 CDE (Common Desk Environment, 公共台式环境) 图形环境，帮助信息全部采用中文，COSIX64 被认为是我国的 UNIX 用户的一个近乎完美的解决方案。除了界面的汉化外，中文的编辑、输入、输出、打印等方面都取得了很好的效果。

1.2.4 软件开发技术的发展

有了好的算法、软件开发环境还不能开发出好的软件，还必须要有好的软件开发技术。

软件开发技术随程序设计语言由模块化向结构化再向面向对象而相应发展。软件开发技术可分为流行于学术界的形式化方法和流行于工业界的非形式化方法。而工程的、非形式化的软件开发技术，即传统的称为瀑布模型的软件开发技术。

1. 传统的瀑布模型 (waterfall model) 方法

传统的软件开发模型是瀑布模型，它基本上反映了软件个体手工开发时所习惯用的开发过程和开发方法。典型的瀑布模型如下面图 1.5 所示。

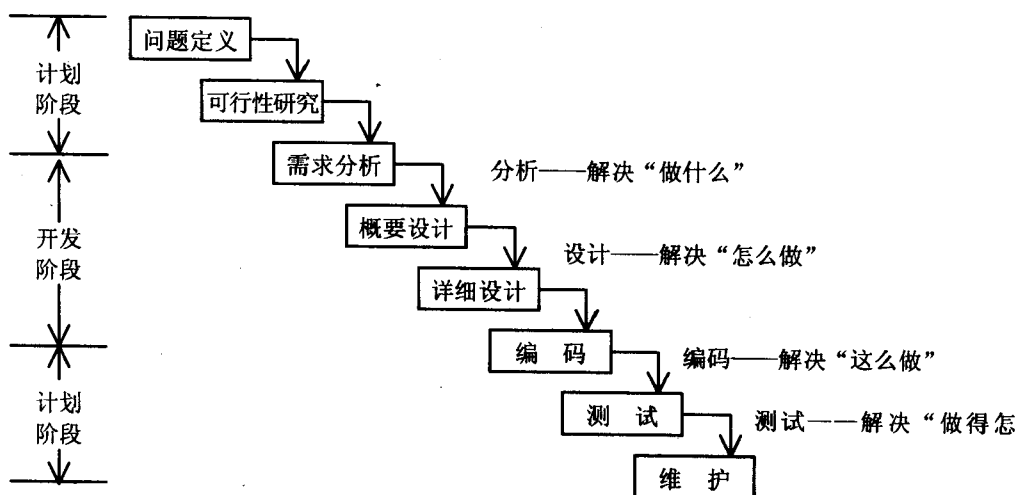


图 1.5 典型的瀑布模型

图 1.5 的瀑布模型把软件的开发阶段划分为四个主要步骤，即分析、设计、编码和测试，各个步骤自上而下紧密相连，如同一泻而下的“瀑布”。

在开发阶段的五个步骤相继完成“做什么”、“怎么做”、“这么做”和“做得怎样”四个任务。

随着软件开发技术的发展，又出现了一些新的如快速原型模型、操作模型、软件演进模型等。

2. 广泛使用的结构化设计 (structured design) 方法

在软件设计的长期实践中使用着两种最基本的设计策略,一种是模块化设计(modular design),另一种是由顶向下设计(top down design),后一种已成为现代软件开发中的主要设计策略。

在上面两种设计策略的基础上,出现了许多对软件进行设计的系统方法。其中结构化设计方法(又称SD法)是比较成熟和广泛使用的一种。

SD方法是一种面向数据流的设计方法,重点在于确定软件的结构,主要是用来指导瀑布模型中软件“概要设计”的。因此,它的目的主要是满足系统要求的最佳软件结构。

3. 先进的面向对象设计(object-oriented design)方法

软件开发中一直受两大难题所困扰:一个是如何跨越程序复杂性障碍;另一个是如何在计算机系统中自然地表示出客观世界,即对象模型(object model)。由C++语言编写的面向对象程序设计是解决上述两大难题的最有前途的方法。面向对象程序设计是软件开发方法的一场革命,它代表了新颖的程序设计思维方法,使计算机对问题的求解更接近人的逻辑思维方式。这种方法是通过增加软件可扩充性和可重用性,以提高程序员的编程能力,并能控制软件维护的复杂性和开销。

对象(object)通常作为计算机模拟思维,表示真实世界的抽象。一个对象如同一个软件构件(software component),它包含相应的数据结构和相关的操作。对象本身可为用户提供一系列服务,如改变对象状态、测试、传递消息等,用户无需知道服务的任何实现细节,操作完全是封闭的,即对象是一组数据和作用于这组数据的操作(也叫方法)的封装(package)。

面向对象程序设计中定义了一些概念,如抽象数据类型(abstract data type)、类(class)和类型层次(type hierarchy)、子类(subclass)、继承性(inheritance)、消息、事例等。

Ada语言在推广面向对象方法和技术上起到了不小的作用。C++是C语言基础上加以扩展并成为一种面向对象的程序设计语言。由于它是C语言的扩充,容易学习和掌握,很受用户欢迎。

面向对象软件开发(OOSD -- Object_oriented Software Development)的开发过程一般分为三个阶段:面向对象的分析(OOA -- Object_Oriented Analysis)、面向对象的设计(OOD -- Object_Oriented Design)和面向对象程序设计(OOP -- Object_Oriented Programming)。

从80年代到现在,对面向对象软件开发技术的研究发展很快,新的方法层出不穷,据不完全统计约有50余种。但不论哪种方法,都覆盖上面开发过程的三个阶段,而且都把OOA阶段作为重点。因为科学和正确的问题模型的建立是基于对问题域内问题所涉及的对象与对象之间关系的正确分析之上的。

面向对象的软件开发方法逐渐成熟之后的首要任务是使这种开发方法统一化和标准化。1997年1月由Rational软件公司推出的UML(Unified Modeling Language,统一建模语言),UML语言主要用来规范、表示和构造系统的模型,是一个通用的建模语言,适用于使用各种软件开发方法的用户。UML是一种标准的建模语言,而不是一个标准的开发过程,它完全独立于开发过程。已显示出强大生命力的UML建模语言,将是未来面向对

象软件开发方法的核心。

非面向对象的过程语言（如 C、PASCAL 等），其数据结构是问题解的核心。一个软件系统的结构是围绕一个或几个关键数据结构为核心而组成的。这就使软件的开发很难逾越程序复杂性的障碍和计算机系统中如何自然地表示客观世界（即对象模型）的困扰。

“算法 + 数据结构 = 程序设计”的原则在软件开发的过程中产生了积极和深远的影响。但随着软件系统的规模越来越大，复杂性越来越增长，人们不得不对“关键数据结构”重新评价。数据结构的主要缺点是应用范围受到局限，而且可视性差。因为在非面向对象的软件系统中，许多重要的过程和函数（子程序）的实现严格地取决于关键数据结构，如果这些关键数据结构中的一个或几个数据有所改变，将影响到整个软件系统，致使许多过程和函数必须重新编写，严重时会导致整个系统崩溃。

在面向对象的程序设计中，是把密切相关的数据与过程定义为一个整体（即对象），用以克服复杂性的障碍。面向对象软件系统的构造不取决于对象的内部结构，仅取决于定义能在对象内部数据上实行操作的方法。

1.2.5 软件工程与软件应用

由于软件成本的急剧增长，软件系统的规模越来越大，软件的质量越来越受到人们的关注，在软件开发中越来越遇到如何开发、如何维护应用需求日益增长的软件，于是出现了“软件危机”（software crisis）。为了克服“软件危机”，出现了“软件工程”（software engineering）技术。为了延长“软件生存周期”（software life cycle），软件工程的各个步骤，包括设计、分析、实现、测试和维护等都是必要的。

本书将从传统和现实出发，对软件的需求分析、软件设计、软件测试与维护，结合软件工程的规范和要求，进行比较系统的介绍，为非计算机专业理工类机电专业的大学本科初步掌握软件开发中的软件工程技术，打下一个起码的基础。

1. 计算机系统工程

计算机系统工程是计算机应用中对问题求解的活动，对系统所要求的功能进行分析并分配给系统的各个部分。计算机系统工程的概貌如图 1.6 所示。

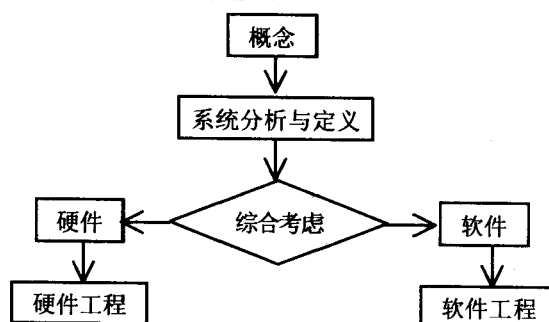


图 1.6 计算机系统工程

2. 软件工程的基本目标

对于不同应用领域的软件，分析问题和求解问题的方法与手段大不相同，但可以使用一些与具体应用无关的技术来研究如何开发软件，这些技术就构成了软件工程方法学（methodology）。

软件工程的基本目标为：

(1) 它是一种定义良好的方法学，该方法学是面向包括计划、开发和维护阶段的软件生命周期的；

(2) 它是一组确定的软件成分，它对软件生命周期的每一步都记录在文档资料上，而

且具有显示每一步轨迹的能力；

(3) 它是一组在软件周期中可以预测且可以每隔一定时间对其进行复审的技术目标。

软件生命周期包括从软件开始开发之前到软件交付使用之后的全部计划阶段 (planning stage)、开发阶段 (development stage) 和维护阶段 (maintenance stage)。

3. 软件的应用

决定一种软件应用的性质时，信息的“内容”和“确定性”是两个重要因素。内容 (content) 是指输入信息和输出信息的含义和形式。例如在很多商业应用中都使用高度结构化的输入数据 (即一个数据库)，并产生具有某种格式的“报告” (report)。而那些控制一个自动化机器的软件 (例如数值控制)，则使用具有限定结构的离散数据，并快速产生响应的控制“命令” (order)。“确定性” (determinacy) 是指信息的次序与时序的可预测性。一个工程分析程序接收具有预定次序的数据，连续执行分析算法并产生用表格或图形表示的结果数据，这就是具有“确定性”的信息。反之则是“不确定”的信息，例如，对于一个多用户操作系统，输入的信息内容是多种多样的，输入的次序和时序是任意和随机的，执行的算法可被外部条件中断，产生的输出结果数据将随着环境和时间而改变。

由于以上原因，软件的应用领域很宽，包括：

(1) 系统软件 (system software)，它是为其它程序运行服务而编写的一些程序的集合。有些系统软件 (例如编译程序、编辑程序等) 处理确定的信息结构，而有些系统软件 (例如操作系统、驱动程序等) 大多处理非确定的信息结构。不论哪一种系统软件，共同的特点是：与计算机硬件频繁地相互作用、多个用户频繁使用、需要进行调度、资源共享和进程管理、复杂的数据结构和多个外部接口。

(2) 实时软件 (realtime software)，它是测量 (数据采集部件)、分析 (数据转换部件) 和控制 (控制输出部件和监控部件) 现实世界中某些客观事件的软件。“实时”与“交互” (interaction)、“分时” (shared - time) 不同。一个实时系统必须严格地在规定的时间内作出响应 (response)。

(3) 商用软件 (commercial software)，它是专门处理商业信息的应用软件。如信息管理系统 (MIS) 软件、大型数据库软件、电子商务软件等。

(4) 工程和科学软件 (engineering and science software)，它是指那些对数字进行大量运算的工程与科学软件。其应用范围从天文学到核物理；从汽车的应力计算到宇宙飞船的轨道动力学；从分子生物学到大规模自动化生产。

4. 软件复用技术

(1) 软件复用技术的由来

在通常情况下，应用软件系统的开发过程一般包括需求分析、设计、编码、测试和维护几个阶段。如果每个应用软件的开发都“从零开始”，势必在开发过程中存在大量的重复劳动。为了避免软件开发中人力资源的浪费，提高应用软件开发的效率和质量，降低应用软件开发的成本，最大限度地避免重复开发，从 1968 年开始，人们经过研究并产生了应用软件开发中的复用技术。

软件复用 (software reuse) 是指重复使用“为了复用目的而设计的软件”的过程。反

过来说就是：可复用软件（reusable software）是指那些为了复用目的而设计的软件。实践证明，软件产业要发展并形成规模经济，像机械、建筑等传统行业一样，标准构件的生产和复用是关键因素。

(2) 软件复用的种类

根据复用的对象，软件复用可分为产品复用（products reuse）和过程复用（process reuse）。产品复用是复用已有的软件构件（software component），通过这些软件构件的集成（或组装）得到新的系统，产品复用是目前的现实、主流途径。过程复用是复用已有的软件开发过程，使用可复用的应用生成器（application generator）来自动或半自动地生成所需系统。过程复用取决于软件自动化的发展程度，目前只适用于一些特殊的应用领域。

根据对可复用信息进行复用的方式，可以把软件复用分为黑盒（black-box）复用和白盒（white-box）复用。前者是指对已有的构件不需要做任何修改，直接进行复用。后者是指对已有构件进行必要修改后才可使用。

(3) 实现软件复用的关键因素

实现软件复用的关键因素主要包括：软件构件技术（software component technology）、领域工程（domain engineering）、软件构架（software architecture）、软件再工程（software reengineering）、开放系统（open system）、软件过程（software process）、计算机辅助软件工程 CASE（Computer Aided Software Engineering）等技术因素和非技术因素，它们互相联系、互相影响。如图 1.7 所示。

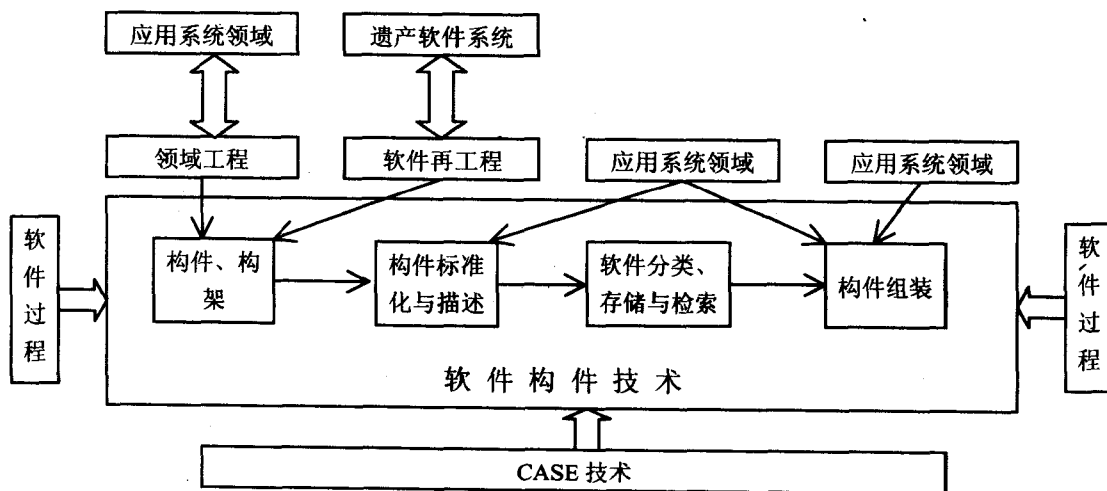


图 1.7 实现软件复用的关键因素

软件复用技术的应用，从狭义上说大大提高了软件开发的效率和质量，从广义上说大大促进了软件产业的变革，使软件产业真正走上工程化、工业化的发展道路。软件系统的开发将由软件系统集成商通过购买商用构件，然后再集成组装来完成。

1.2.6 数据库的发展与展望

在计算机技术和网络技术高速发展的带动下，数据库的发展极为迅速，应用范围极为广泛，在当今的信息社会中几乎无所不在。传统的层次型、网状型和关系型数据库技术在管理和商务等应用领域中取得了极大的成功。但随着 Internet 技术的普及，人们对数据