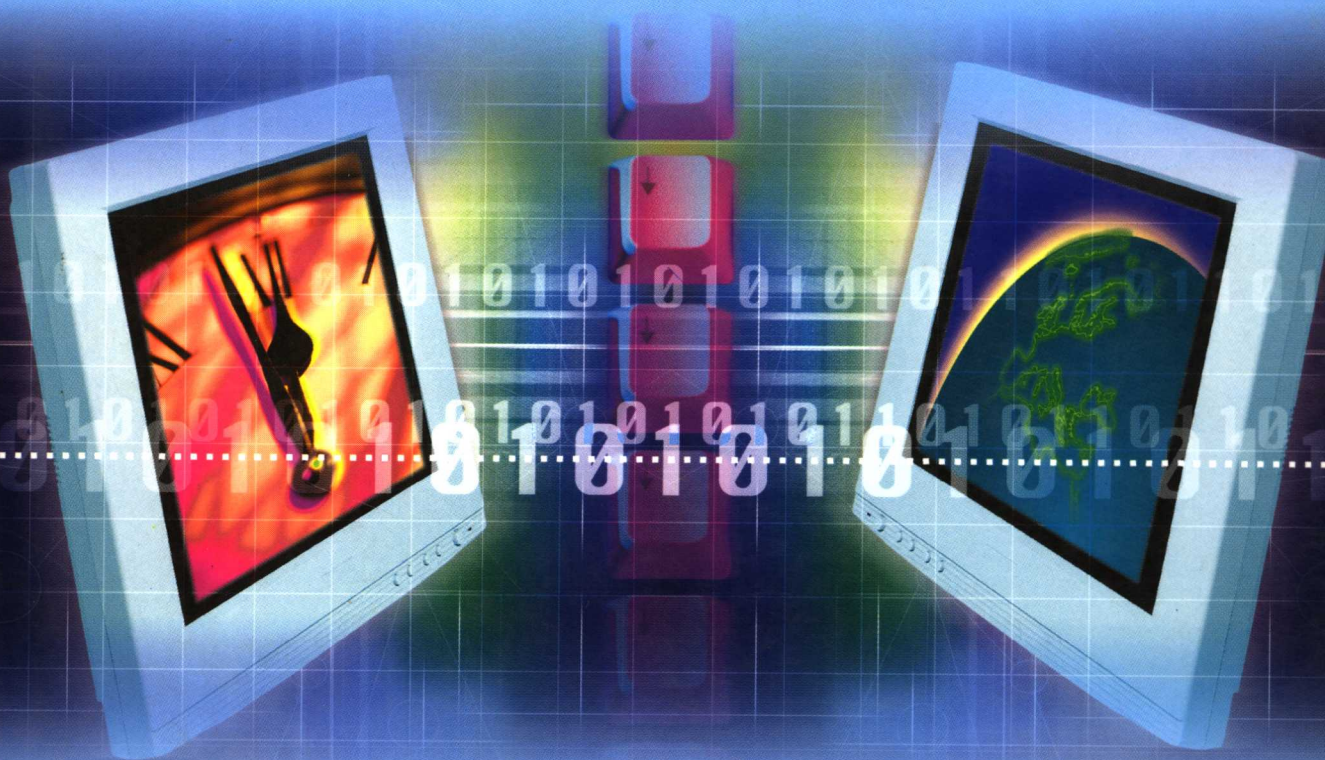


高等学校教材

计算机软件 技术基础

主编 陈建铎

副主编 王忠义 谢尊贤



高等教育出版社

高等学校教材

计算机软件 技术基础

第二版

清华大学出版社



清华大学出版社

高等学校教材

计算机软件技术基础

主 编 陈建铎
副主编 王忠义 谢尊贤

高等教育出版社

内 容 提 要

本书按照教育部提出的非计算机专业计算机基础课程多层次教学体系中软件技术基础课程的要求,讲授进行应用软件开发需要了解和掌握的一些基础知识和基本技能。

全书共分 12 章,主要包括:计算机软件概述,程序设计语言与编译,算法的概念、表示、复杂度以及高级语言程序设计中的常用算法,结构化程序设计,面向对象程序设计,数据结构,操作系统,数据库系统基础,软件工程,Visual C++ 可视化程序设计,多媒体技术与应用,网络编程等。

本书在内容组织上由浅入深,循序渐进,语言通俗流畅,以利于教学和自学。按教学需要和要求,本书基本上可按两种模式组织教学,一种是多学时,可修完全部内容;另一种是少学时,可选修前 9 章。本书还配有电子教案。

本书可作为高等院校非计算机专业学生学习“计算机软件技术基础”课的教材,也可供其他软件开发人员自学参考。

图书在版编目(CIP)数据

计算机软件技术基础/陈建铎主编. —北京:高等教育出版社,2003.8

ISBN 7-04-012667-2

I. 计... II. 陈... III. 软件-高等学校-教材
IV. TP31

中国版本图书馆 CIP 数据核字(2003)第 045326 号

出版发行 高等教育出版社

购书热线 010-64054588

社 址 北京市西城区德外大街 4 号

免费咨询 800-810-0598

邮政编码 100011

网 址 <http://www.hep.edu.cn>

总 机 010-82028899

<http://www.hep.com.cn>

经 销 新华书店北京发行所

印 刷 北京印刷三厂

开 本 787×1092 1/16

版 次 2003 年 8 月第 1 版

印 张 21.75

印 次 2003 年 8 月第 1 次印刷

字 数 530 000

定 价 27.20 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

前 言

随着计算机应用技术的普及与发展,计算机已经在科学技术乃至人类社会的各个领域都显示出无比巨大的作用。现今,充分发挥计算机的潜能,使用这一现代化的工具发展科学技术,推动人类社会的进步,不仅要依靠计算机专业人员,更要依靠广大非计算机专业的人才,由他们在各个领域从事更深层次的开发与研究。

在高等院校,为非计算机专业学生开设“高级语言程序设计”这门课,就是为了提高学生使用计算机解决本专业实际问题的能力。但是,计算机应用作为一门科学与技术,不是仅仅学习一门高级语言就能掌握好的。它涉及到计算机内部的体系结构、数据的组织形式、软件环境、多媒体网络技术以及许多新的语言,而且在应用开发中还存在有许多的方法和技巧。因此按照教育部提出的非计算机专业计算机基础课程三层次教学体系的要求,为非计算机专业学生开设“软件技术基础”课程,就成为必然的选择。

近年来,在理工科非计算机专业开设“软件技术基础”课的工作已经十分普及,许多出版社出版了相应的教材。但是,计算机应用技术发展很快,其教材需要不断地补充和更新,摒弃落后陈旧的内容,增添新的理论、技术、方法和技巧。为此,我们组织长期从事计算机软件教学的专家、教授,在深入研究现有教材的基础上,结合国内外计算机的最新技术,重新编写了“软件技术基础”,以满足广大非计算机专业学生对“高级语言程序设计”中更深层次的理论、方法和技巧的需求。本教材明显的特点是增添了 Visual C++ 可视化程序设计、多媒体技术与应用和网络编程,在软件工程一章中突出了 UML 的思想和方法。在章节编排和文字处理方面,我们尽量做到由浅入深,循序渐进,语言流畅,以利于教学和自学参考。同时,还配以电子教案。

本书共分 12 章。其中第一章概述,主要介绍软件的特征、类型和发展;第二程序设计与编译,主要介绍程序设计语言的类型、基本特征、程序的结构化控制以及高级语言的编译过程;第三章算法,主要介绍算法的概念、表示、复杂度以及高级语言程序设计中的常用算法;第四章结构化程序设计,主要介绍结构化程序设计的基本概念、基本结构及其设计方法;第五章面向对象程序设计,主要介绍面向对象的基本思想、面向对象程序设计语言的种类、特征、基本要素的定义、创建、操作以及面向对象程序设计的方法;第六章数据结构,主要介绍数据结构的概念、基本类型(线性表、栈、队列、串、数组、二叉树、图)以及查找和排序等数据处理的方法;第七章操作系统,主要介绍操作系统的基本概念、结构模式、资源管理、操作系统的服务与用户接口以及 UNIX、Windows NT、Linux、Windows XP 等常用操作系统的特征;第八章数据库系统基础,主要介绍数据库的基本概念、数据模型、数据库系统结构、关系模型的数学定义、关系型数据库的相关理论、标准查询语言 SQL 以及关系型数据库的安全性、完整性;第九章软件工程,主要介绍软件工程的基本概念、软件开发的方法、软件过程、建模语言 UML 以及软件开发的过程;第十章 Visual C++ 可视化程序设计,主要介绍 Visual C++ 可视化程序设计的基本方法、MFC 编程工具、Windows 编程思想、MFC 程序结构、MFC 类库编程以及 MFC 程序的设计与调试等;第十一章多媒体技术与应用,主要介绍多媒体的基

本概念、多媒体计算机的组成与信息处理技术、多媒体应用软件的开发过程以及常用创作工具等;第十二章网络编程,主要介绍网络编程的协议基础、网络工作模式、Socket 程序设计、动态网页设计以及 Java 语言在网络编程中的应用等。

全书按照 72 学时编写,以供不同层次、不同专业学生的教学需求。基本上可按两种模式组织教学,一种是 72 学时,修完全部内容;另一种是 60 学时,选修前 9 章。

本书第一、二、十一章由陈建铎编写,第三章由杨卫社编写,第四章由段宏斌编写,第五章由张莉编写,第六章由谢尊贤编写,第七章由廉师友编写,第八章由王忠义编写,第九章由卫红春编写,第十章由郑凯东编写,第十二章由任多礼编写。陈建铎、王忠义统稿、编校。编写完后,教育部计算机基础课程教学指导委员会主任、西安交通大学博士生导师冯博琴教授对全稿进行了详细的审阅,提出了许多具体的建议和修改意见,在此表示诚挚的感谢。但是,由于作者水平有限,难免存在不足或谬误之处,诚请同行专家和广大读者批评指正。

编 者

2003 年 6 月

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人给予严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话：(010) 82028899 转 6897 (010)82086060

传真：(010) 82086060

E-mail: dd@hep.com.cn

通信地址：北京市西城区德外大街4号

高等教育出版社法律事务部

邮编：100011

购书请拨打读者服务部电话：(010)64054588

策划编辑	何新权
责任编辑	何新权
封面设计	王凌波
责任印制	孔源



目 录

第一章 概述	(1)	3.2.1 用自然语言表示算法	(48)
1.1 程序与软件	(1)	3.2.2 用流程图表示算法	(48)
1.1.1 程序	(1)	3.2.3 用伪代码表示算法	(49)
1.1.2 软件	(2)	3.3 算法的复杂度	(50)
1.2 软件的特征与类型	(2)	3.3.1 算法设计的目标	(50)
1.2.1 软件的特征	(2)	3.3.2 算法的时间复杂度	(51)
1.2.2 软件的类型	(4)	3.3.3 算法的空间复杂度	(52)
1.3 软件的产生、发展与软件危机	(5)	3.4 常用算法	(52)
1.3.1 软件的产生与发展	(5)	3.4.1 枚举法	(52)
1.3.2 软件危机	(6)	3.4.2 迭代法	(53)
1.4 常用软件	(7)	3.4.3 递归法	(55)
习题与思考题	(9)	3.4.4 递推法	(55)
第二章 程序设计语言与编译	(10)	3.4.5 分治法	(56)
2.1 程序设计语言	(10)	3.4.6 回溯法	(57)
2.1.1 机器语言	(10)	3.5 数值计算	(58)
2.1.2 汇编语言	(10)	3.5.1 数值计算基本过程	(59)
2.1.3 高级语言	(11)	3.5.2 误差	(59)
2.2 高级语言基本特征	(12)	3.5.3 函数的插值与逼近	(59)
2.2.1 高级语言词法标记	(12)	3.5.4 代数方程求解	(62)
2.2.2 基本数据类型	(13)	3.5.5 数值积分	(63)
2.2.3 常量、变量与表达式	(15)	习题与思考题	(64)
2.2.4 自定义数据类型	(20)	第四章 结构化程序设计	(66)
2.3 程序的结构与控制	(28)	4.1 结构化程序设计及其基本结构	(66)
2.3.1 赋值语句与顺序程序结构	(29)	4.1.1 程序设计的发展过程	(66)
2.3.2 控制语句与程序结构的控制	(29)	4.1.2 结构化程序设计及其基本 结构	(66)
2.3.3 函数与函数的参数传递	(33)	4.2 结构化程序设计方法	(69)
2.3.4 变量的作用域与存储类型	(36)	4.2.1 逐步求精法	(69)
2.3.5 输入/输出	(39)	4.2.2 模块化程序设计法	(71)
2.4 高级语言程序的编译与执行	(40)	4.3 结构化程序设计举例	(73)
2.4.1 编译执行	(40)	习题与思考题	(77)
2.4.2 解释执行	(41)	第五章 面向对象程序设计	(78)
习题与思考题	(43)	5.1 面向对象程序设计的思想和方法	(78)
第三章 算法	(45)	5.2 面向对象程序设计语言	(79)
3.1 算法的概念	(45)	5.3 类、对象与封装	(79)
3.1.1 算法的两要素	(45)	5.3.1 对象	(80)
3.1.2 算法的特征	(47)	5.3.2 类	(80)
3.2 算法的表示	(48)		

5.3.3 封装.....	(80)	6.6.2 二叉树的定义及性质	(129)
5.4 函数重载	(81)	6.6.3 二叉树的存储结构	(130)
5.5 类的定义和对对象的说明	(82)	6.6.4 二叉树的遍历	(133)
5.5.1 类的定义.....	(82)	6.6.5 树、森林到二叉树的转换.....	(134)
5.5.2 定义成员函数.....	(83)	6.7 图	(135)
5.5.3 创建对象.....	(84)	6.7.1 图的基本概念与术语	(135)
5.5.4 构造函数和析构函数.....	(85)	6.7.2 图的存储	(136)
5.6 继承和派生类	(86)	6.7.3 邻接表图的遍历	(140)
5.6.1 继承.....	(86)	6.8 查找	(142)
5.6.2 派生类.....	(87)	6.8.1 基本概念	(142)
5.6.3 派生类的构造函数和析构 函数.....	(90)	6.8.2 线性表的查找	(143)
5.7 运算符重载	(92)	6.8.3 树表的查找	(146)
5.7.1 运算符重载.....	(92)	6.8.4 哈希表的查找	(148)
5.7.2 实际举例.....	(93)	6.9 排序	(150)
5.8 类模板和函数模板	(95)	6.9.1 基本概念	(150)
5.8.1 类模板.....	(95)	6.9.2 插入排序	(151)
5.8.2 函数模板.....	(98)	6.9.3 交换排序	(152)
5.9 面向对象软件开发方法	(100)	6.9.4 选择排序	(154)
5.9.1 面向对象的分析	(101)	6.9.5 归并排序	(155)
5.9.2 面向对象的设计	(103)	习题与思考题	(156)
习题与思考题	(103)	第七章 操作系统	(158)
第六章 数据结构	(106)	7.1 操作系统概述	(158)
6.1 数据结构概述	(106)	7.1.1 操作系统的概念	(158)
6.1.1 数据结构的概念	(106)	7.1.2 操作系统的类型	(159)
6.1.2 抽象数据类型	(107)	7.2 操作系统的资源管理	(161)
6.1.3 数据结构应用示例	(108)	7.2.1 进程管理	(161)
6.2 线性表	(109)	7.2.2 存储管理	(169)
6.2.1 线性表的逻辑结构及其操作 ..	(109)	7.2.3 文件管理	(173)
6.2.2 线性表的顺序存储结构	(109)	7.2.4 设备管理	(177)
6.2.3 线性表的链式存储结构	(111)	7.3 操作系统的服务与用户接口	(178)
6.3 栈和队列	(115)	7.3.1 操作系统为用户提供的公共 服务	(178)
6.3.1 栈	(115)	7.3.2 系统调用	(179)
6.3.2 队列	(118)	7.3.3 API 函数	(179)
6.4 串	(122)	7.3.4 操作系统的用户接口	(180)
6.4.1 串及其运算	(122)	7.4 操作系统的结构	(181)
6.4.2 串的存储结构	(123)	7.4.1 传统操作系统的结构模式	(181)
6.5 数组	(124)	7.4.2 现代操作系统的内核	(182)
6.5.1 数组定义和运算	(125)	7.4.3 现代操作系统的结构模式	(183)
6.5.2 数组的顺序存储结构	(125)	7.5 操作系统实例介绍	(184)
6.5.3 矩阵的压缩存储结构	(126)	7.5.1 UNIX	(184)
6.6 树和二叉树	(128)	7.5.2 Windows NT	(185)
6.6.1 树的定义和基本操作	(128)	7.5.3 Linux	(187)

7.5.4 Windows XP	(188)	9.1.1 软件工程的形成与发展	(233)
习题与思考题	(190)	9.1.2 软件工程的概 念	(233)
第八章 数据库系统基础	(192)	9.1.3 软件工程学的基本内容	(233)
8.1 数据库概述	(192)	9.1.4 软件工程学的特点及与 其他学科的关系	(235)
8.1.1 数据管理技术的发展	(192)	9.2 软件开发方法	(235)
8.1.2 数据库的体系结构	(193)	9.2.1 软件开发方法的分类	(235)
8.1.3 数据库管理系统(DBMS)	(194)	9.2.2 结构化方法	(236)
8.2 数据模型	(195)	9.2.3 JSD方法	(236)
8.2.1 E-R模型	(195)	9.2.4 原型法	(237)
8.2.2 层次模型与网状模型	(196)	9.2.5 面向对象法	(237)
8.2.3 关系模型	(198)	9.3 软件过程	(237)
8.3 数据库系统结构	(199)	9.3.1 软件的生命周期	(237)
8.3.1 单用户数据库系统	(199)	9.3.2 软件过程	(237)
8.3.2 集中式数据库系统	(199)	9.3.3 统一软件开发过程	(239)
8.3.3 分布式数据库系统	(200)	9.4 软件建模语言	(242)
8.3.4 客户机/服务器数据库系统	(200)	9.4.1 软件语言与软件建模语言	(242)
8.3.5 Internet 数据库系统	(200)	9.4.2 UML概述	(242)
8.4 关系模型的数学定义	(201)	9.4.3 UML的基本内容	(243)
8.4.1 n 元关系	(201)	9.5 软件开发	(246)
8.4.2 关系代数	(202)	9.5.1 需求	(246)
8.4.3 关系演算与域关系演算	(207)	9.5.2 分析	(249)
8.5 关系数据库理论	(208)	9.5.3 设计	(252)
8.5.1 异常问题	(208)	9.5.4 实现	(256)
8.5.2 函数依赖	(209)	9.5.5 测试	(258)
8.5.3 关系模式的分解	(211)	习题与思考题	(259)
8.5.4 关系模式的范式	(212)	第十章 Visual C++ 可视化程序设计	(261)
8.6 关系数据库标准查询语言	(213)	10.1 Visual C++ 可视化编程概述	(261)
8.6.1 SQL概述	(213)	10.1.1 Visual C++ 与 MFC 编程 工具	(261)
8.6.2 SQL的数据定义	(214)	10.1.2 Windows 与 MFC 编程特点	(264)
8.6.3 SQL的数据查询	(216)	10.1.3 MFC 应用程序开发过程	(265)
8.6.4 SQL的数据更新	(218)	10.2 Windows 编程思想	(266)
8.6.5 嵌入式 SQL	(219)	10.2.1 窗口、句柄、事件和消息	(266)
8.7 关系数据库的安全与完整性	(220)	10.2.2 常用消息	(267)
8.7.1 数据库的安全性	(220)	10.2.3 事件驱动与消息传递	(268)
8.7.2 数据库的完整性	(223)	10.3 MFC 程序结构	(269)
8.8 新型数据库简介	(225)	10.3.1 MFC 类库层次结构	(269)
8.8.1 面向对象数据库	(225)	10.3.2 MFC 应用程序框架	(270)
8.8.2 分布式数据库	(228)	10.3.3 MFC 程序结构分析	(271)
8.8.3 并行数据库	(229)	10.4 使用 MFC 类库编程	(274)
8.8.4 多媒体数据库	(229)	10.4.1 输入接口设计	(274)
习题与思考题	(230)	10.4.2 屏幕输出方法	(278)
第九章 软件工程	(233)		
9.1 软件工程概述	(233)		

10.4.3 文件处理.....	(280)	11.4.2 Director 的特点与使用	(308)
10.5 MFC 程序编写与调试	(281)	11.4.3 Flash 的特点与使用	(310)
10.5.1 创建工作区与工程.....	(281)	习题与思考题	(312)
10.5.2 源代码编辑.....	(281)	第十二章 网络编程	(314)
10.5.3 编译与调试技术简介.....	(284)	12.1 网络编程的协议基础和网络工作	
习题与思考题	(286)	模式.....	(314)
第十一章 多媒体技术与应用	(287)	12.1.1 TCP/IP 协议	(314)
11.1 多媒体技术概述.....	(287)	12.1.2 客户机/服务器工作模式	(317)
11.2 多媒体计算机的组成与信息处理		12.2 Socket 程序设计	(318)
技术.....	(288)	12.2.1 UNIX Socket 程序设计	(318)
11.2.1 多媒体计算机基本配置.....	(288)	12.2.2 Windows Sockets 程序设计	(320)
11.2.2 音频处理技术.....	(290)	12.3 动态网页设计.....	(324)
11.2.3 视频处理技术.....	(292)	12.3.1 CGI 程序设计	(324)
11.2.4 光盘 CD-ROM	(296)	12.3.2 ASP 网页设计	(328)
11.2.5 Windows 的多媒体功能	(297)	12.3.3 网页与数据库的连接.....	(332)
11.3 多媒体应用技术.....	(299)	12.4 Java 语言的网络编程	(334)
11.3.1 多媒体应用软件的开发		12.4.1 Java 语言简介	(334)
过程.....	(299)	12.4.2 Java Applet	(335)
11.3.2 多媒体素材设计.....	(300)	12.4.3 Java 开发环境	(335)
11.3.3 多媒体开发工具与平台.....	(301)	12.4.4 Java 语言编程实例	(336)
11.3.4 计算机动画.....	(302)	习题与思考题	(337)
11.4 多媒体创作工具简介.....	(305)	参考文献	(339)
11.4.1 Authorware 的特点与使用	(305)		

第一章 概述

随着计算机的发展,软件技术逐步成为一门相对独立的学科。它研究程序设计的理论和方法,创建各种应用环境,以便使计算机具有更为强大的功能。本章的目的在于使读者在学习软件技术基础之前,对程序和软件的定义、特征、类型以及软件的产生与发展过程有一个概括的了解。

1.1 程序与软件

众所周知,计算机是由硬件和软件组成,称之为计算机系统。计算机硬件有 CPU、存储器、输入设备、输出设备以及外部存储器等。其中常用的输入/输出设备有键盘、鼠标、扫描仪、显示器、打印机和投影仪等;在构成计算机网络时,通信设备既是输入设备又是输出设备。这些设备之所以能够工作,除了硬件自身的功能之外还配有一定的软件。如果没有相应的软件支持,再好的设备也是不能工作的。

一般而言,计算机硬件的功能是有限的,只能进行数据传送和一些简单的算术/逻辑运算。但是,人们用它能进行复杂的数学计算,实施高难度的过程控制,实现奇妙的三维动画,构造虚拟世界,支持全球互联网的运行等等,这些都是通过软件来实现的。人们已经知道,对于相同的硬件,如果配置的软件越丰富,其功能也就越强。因此,在学习计算机应用技术时,除了掌握必要的硬件技术知识之外,还必须学习一定的软件技术知识。

1.1.1 程序

目前的主流计算机都是按照冯·诺依曼“存储程序”的思想设计的,因此,程序也是计算机的主要组成部分。通常把指示计算机进行某一工作的命令称为指令,而为完成某一任务的若干条指令的有序集合称为程序。程序由指令组成,具有以下属性:

1. 能完成某一确定的任务;
2. 由一种计算机语言来描述;
3. 能在一定的计算机系统环境下运行。

由于程序反映的是人们进行数学计算或求解某一问题的思想,因此它一方面能输入计算机执行,另一方面可供人们阅读和交流。在任何一台计算机中,无论是数学计算、图形图像处理、过程控制还是上网通信,都是依靠程序进行的。因此可以说,计算机的全部工作就是执行程序的过程。

为了设计出高水平的程序,一般要求程序设计人员要有清楚的概念、正确的算法和熟悉程序设计的语言。设计出的程序应当编排合理,结构紧凑,功能完整,易于执行,易于阅读和理解。

1.1.2 软件

通常,人们把程序及其相关的数据统称为软件。而严格地讲,软件应当是计算机系统中程序、数据及其相关文档的总称。其中程序是软件的主要组成部分和表现形式;数据是在软件处理过程中用来描述事物的静态特征,是程序处理的对象;而文档是与程序设计、维护及使用有关的图文资料,是对软件开发和维护过程的描述与记录。它包括软件需求调研报告、立项与组织文档、设计与调试文档、运行与维护文档等。

一般软件具有两方面的含义。其一是个体含义,表示计算机系统中具体的程序、数据和有关文档,例如软件“Windows XP”,是从个体含义上讲的;其二是整体含义,它相对于硬件而言,是对计算机系统中所有程序、数据及相关文档的概括。

软件有两种属性。其一是静态属性,它由程序、数据及相关文档组成,可以存储,也可供人们阅读和交流;其二是动态属性,它蕴涵着一定的思想,即计算机的操作步骤,由计算机执行而产生特定的结果或动态效应。

在实际使用中,人们总希望计算机的性能非常良好,这无疑涉及到硬件和软件两个方面的性能。为了使一台计算机能够发挥良好的效果,就必须配以良好的程序。

如今,为计算机进行软件设计,使其发挥常人难以想象的神奇功效,已经成为软件人员的使命,也是广大软件人员的追求。

1.2 软件的特征与类型

1.2.1 软件的特征

在计算机中,软件和硬件是两种截然不同的概念和产品。硬件是构成计算机的物理实体,而软件则是一种无形的产品,是人们的理念、逻辑思维与具体算法的集中体现。一个好的软件,往往是众多软件设计人员集体智慧的结晶。

从软件的属性来看,它是一种特殊的事物,具有自身的特性,可概括如下:

1. 智能性

软件是人们智慧的结晶、智力劳动的成果。软件开发的过程是人们高度智力活动的过程,软件中的程序、流程、算法、数据结构是通过人们的思维活动进行设计、编排和组织的。当然,在其他行业的产品中也蕴涵着人们的智慧;但是比较起来,软件中的智力特征更集中,更明显。

一旦一种软件设计成功,可替代人们的智力劳动,也是人们智能的延伸。

2. 无形性

软件是人们的理念、逻辑思维与具体算法的集中体现。它没有明显的物理形体可被人们直接地观察或者欣赏。软件的表示与存储方式也决定了它的无形性,它以二进制编码表示,以电和磁方式存储,这些都处于人们直接感官之外,是人们无法直接感觉到的。至于光盘,它以分子的排列或形变表示二进制信息,也是人们无法直接感知的。这些,都决定了软

件的无形性。

3. 抽象性

软件的智能性与无形性有时令人难以理解,因而导致了认识上的抽象性。实际上,在软件的研制过程中需要调研和分析,然后进行逻辑设计,这往往需要抽象的思维与方法,因而孕育了软件在设计中的抽象性。软件运行时,人们看到的是窗口界面,是其功能和作用的外部表现形式,而丰富的内涵则蕴涵在软件的内部,这体现了软件在实现中的抽象性。这种抽象性增加了人们对软件认识 and 理解的难度。

4. 系统性

任何一个软件都是由多种要素组成的有机整体,是为某一目标而设计,按照一定结构形式组织起来,因而具有明显的系统性和完整性。另外,软件服务于某一领域或在特定的环境下运行,因而从属于某一系统或环境。因此,软件的开发要按一定的系统方法,在一定的环境下进行。

5. 泛域性

泛域性也称为多域性,是指一个软件可服务于多种行业或领域。就目前来看,对于人类智力所涉足的领域,软件都可以发挥作用。如今,没有哪一个行业或领域不在使用计算机,因此软件服务面向于社会的各个方面和人类思维的各个领域。

另外,软件的功能还体现在与服务领域中知识的结合上。它与服务领域中的数据信息、过程参数、处理思想融合到一起。这种关系决定了软件组成的复杂性和设计的困难性。因此在进行软件开发时,不仅要考虑软件本身的需求,更多地要考虑服务领域中的相关信息。

6. 依附性

由于软件的开发是按一定的系统方法,在一定的环境下进行的,因此,它不像其他产品或设备那样独立地存在与工作,而要依附于一定的环境。这种环境由计算机的硬件、软件、网络环境等要素组成。这些环境也就决定了软件开发和运行时的依附性。

7. 非损性

软件在使用过程中不像其他产品那样存在损耗和物理老化现象,只要硬软件环境不变,就可以永远使用。

8. 复制性

由于软件是以电、磁、光等物质形式存储和传输的,这就决定了软件可以以同样的物质形式复制。软件的可复制性决定了软件开发的成本主要体现在首次开发过程中,一旦开发成功,复制和传输的费用一般很低。

9. 演化性

软件投入使用后,其功能、运行环境甚至界面往往处于不断的变化之中,这一特性称为演化性。由于计算机发展很快,任何一种软件的生存都有一定的时间范围,称之为生命周期。为了延长软件的生命周期,软件开发人员将不断地修改、完善自己的软件,使其适应不断变化的环境,即所谓的版本升级。

1.2.2 软件的类型

目前,世界上的软件很多,可以说是五花八门,因而分类的方式也很多。站在不同的角度,可以得到不同的类型,比如按功能、规模、工作方式、应用领域或服务对象进行划分等。下面仅以软件的功能和规模说明其类型。

1. 按功能分类

若按照软件的功能分类,在一个计算机系统中,软件可分为系统软件、支撑软件和应用软件等三种类型。

(1) 系统软件

系统软件是用来管理计算机的各种资源和程序流程的软件,以保证计算机系统正常运行,为用户提供基本的支持与服务。它靠近计算机的硬件系统,与计算机硬件紧密结合,使计算机系统各个部件及相关软件协调一致地工作。比如操作系统,就是典型的系统软件,它支持所有支撑软件和应用程序的运行。

在微型计算机中,常用的操作系统有 DOS、Windows、UNIX 以及 Linux 等。

(2) 支撑软件

支撑软件也称为开发工具软件,是用来建立开发环境、支持程序员或用户进行各种软件开发与程序设计的软件。例如人们常说的编辑程序、语言处理程序、数据库管理系统以及各种集成开发环境与工具等,均属于支撑类软件,比如 Edit、Office Word、ASM、Visual C、Visual Basic、PowerBuilder、Oracle、Delphi 和 SQL Server 等。

(3) 应用软件

应用软件是为解决某一具体的任务或达到某一特定目标而设计的程序。这类软件是在系统软件的支持下,由设计人员或用户根据某一特定的需求,使用特定的支撑软件设计的程序或者建立的数据管理系统,比如高阶微分方程求解程序、航天飞行器控制程序、图像识别程序、企业信息管理系统、CAI 教学以及股票分析软件等。

2. 按规模分类

软件根据服务目的和自身的组成,有大小之分。服务目的狭小、自身组成简单的软件一般属于小型软件;服务目的宏大、自身组成复杂的软件一般属于大型软件。为了表征软件的大小,常根据软件开发所需的人力、时间和源程序的行数,把软件分成微型、小型、中型、大型和超大型等五种类型。具体划分如表 1.1 所示。

表 1.1 软件规模的划分

类型	开发人员数	开发周期	源程序行数
微型	1	1~4 周	小于 500 行
小型	1~2	1~6 个月	500~5 000 行
中型	2~5	1~2 年	5 000~50 000 行
大型	5~20	1~3 年	50 000~100 000 行
超大型	20 以上	3 年以上	100 000 行以上

微型软件也称为袖珍软件,常用于简单的数值计算、数据采集、数码转换与显示等应用

中;超大型软件也称为巨型软件,常用于大型操作系统、大系统信息分析与控制、多功能组合软件、大型智能图形图像处理以及大型动画游戏中。

这里介绍的按规模分类,目的在于使读者了解软件的大小,并非掌握其定量划分的标准。因为软件处于不断的发展之中,其规模也在不断地膨胀,因此定量划分的标准难以确定。表 1.1 中的参数仅供读者参考。

除此之外,从不同的目的出发,还会有不同的类型。比如按技术特征,还可分为科学计算软件、嵌入式软件、实时软件、个人计算机软件、网络软件和人工智能软件等。

1.3 软件的产生、发展与软件危机

1.3.1 软件的产生与发展

1. 软件的产生

任何一个数学问题的求解,都有一定的步骤;任何一种计算工具的使用都要按照一定的步骤进行操作。我国唐宋时期产生的算盘被誉为世界上最早的计算机,它是在口诀的支配下进行各种复杂的四则运算。以后在欧洲出现的各种计算工具,都有严格的操作步骤和使用规则。电子计算机也要按照一定的步骤进行工作,即按程序工作。

19 世纪中叶,被誉为世界计算机之父的查尔斯·巴贝奇(Charles Babbage)在当时一台穿孔卡织布机的启发下研制一种分析机,艾达·拉夫拉斯伯爵夫人(Ada Augusta Lovelace)作为助手,在为这台分析机编排操作步骤时提出了用二进制数存储的思想,而且发现了编程的基本要素。由于这种开创性的工作,艾达·拉夫拉斯伯爵夫人被誉为世界上的第一位程序员。

由此可见,计算机上的任何一种计算工作都是在程序的支配下工作的。程序伴随着计算机的产生而产生,伴随着计算机的发展而发展。自从第一台电子数字计算机问世以后,相应的程序也就随之而产生,用以描述解题步骤和操作过程。这些程序就是人们今天所说软件的雏形。

任何事物的发展都不是一帆风顺的,软件也是这样,它经历了从无到有、不断发展和成熟的过程,同时也经历了曲折和坎坷。其中最严重的一次,人们称之为“软件危机”。

2. 软件的发展过程

软件的发展过程主要是指电子数字计算机诞生以来的发展过程。概括起来,大致可分为 3 个时代。

(1) 程序设计时代

程序设计时代大约在 1946 年~1955 年,那时计算机比较简单,存储器的容量小,运行速度慢,外围设备少,系统稳定性差。因此,人们关心的重点是硬件系统的性能和技术指标。系统的研究与设计以硬件为中心,软件处于从属的地位。由于在这一时期,计算机主要用于科学计算,因此程序设计主要围绕着算法和数值的求解,由计算机设计人员和数学家以个体的方式独立进行,所编写的程序一般针对性强,规模较小,结构简单,而且讲究编程技巧,追求效率和节省内存空间,这就使得程序不易阅读、理解和修改。在设计过程中,一般以个

体为主,手工操作,相互之间难以交流。

在计算机语言方面,除了机器语言之外,产生了汇编语言。

(2) 软件行业化时代

软件行业化时代大约在 1955 年~1970 年。在这一时期,出现了晶体管和小规模集成电路,使计算机的硬件组成发生了质的飞跃:体积减小,速度提高,内存容量增大,稳定性增强,外围设备的种类和性能也得到了提高。为了提高编程速度,人们在程序设计“自动化”的研究方面取得了较大的突破,产生了 FORTRAN、ALGOL、COBOL 一类的高级语言,为非计算机专业人员进行程序设计创造了条件。同时,也扩大了计算机的应用领域。于是,许多非计算机专业人员纷纷转向程序设计,从而使程序设计进入了行业化的时代。但是在当时,这些人员缺乏训练,水平不高。

进入 20 世纪 60 年代以后,在计算机科学的研究中形式语言、自动机、编译原理以及人工智能有了较大的发展。但是在程序设计方面,却仍然停留在一般高级语言的使用上,程序设计方法与软件开发技巧没有大的突破。这样,一方面是程序的应用领域越来越广,对软件人员的需求急剧增加;另一方面是程序的规模越来越大,结构越来越复杂,而且没有统一的标准,因此在程序设计中出现的错误也就越来越多。于是,产生了软件需求与开发能力之间的尖锐矛盾,即所谓的“软件危机”。

(3) 软件工程化时代

软件工程化时代主要是指从 1970 年到现在。为了解决 20 世纪 60 年代出现的“软件危机”,人们把注意力转移到软件设计方法、技术与原理的研究上,开始用工程化的思想进行软件设计。在进入 20 世纪 70 年代以后,各种并行机、分布式处理、数据库、计算机网络等技术日趋成熟,计算机应用也逐步深入到社会的各个领域。

因此,随之而来的软件技术得到大力发展,各种新概念、新思想、新方法不断出现,使软件设计逐步进入了工程化的时代。在这一时期,相继提出了模块化的程序结构、自顶而下逐步求精的结构化程序设计方法,这称为第一代软件技术。但是,这些方法直到 20 世纪 80 年代才得到广泛的应用。

在 1972 年~1975 年人们提出了软件生存周期模型,开始关注软件的测试,从而出现了许多软件测试方法、技术、原理以及软件确认与验证的理论,这些被称为第二代软件技术。以后,为了进一步解决软件需求分析中的表达与描述,在 1976 年以后人们又提出了有关软件需求定义方面的技术,被称为第三代软件技术。

20 世纪 80 年代以后,人们又对软件开发工具进行优化,然后集中到一起,形成软件开发环境,从而更有效地支持了软件开发工程化的进程,使软件产品的质量与可靠性得到大幅度的提高。如今,随着知识经济时代的来临,软件技术又在迈向产业化的时代。

1.3.2 软件危机

进入 20 世纪 60 年代以后,各种集成电路大量涌现出来,使计算机的类型越来越多,规模越来越大,应用领域越来越广。这样,使计算机的产量和对程序设计的需求骤增。于是,许多非计算机专业人员转向程序设计,从而产生了软件行业。在这一时期,程序的规模越来越大,结构越来越复杂,一个大的程序往往需要上万条乃至数百万条指令。而程序设计人员多数缺乏训练,水平不高,设计时往往以个体为主;虽有多人合作,也因没有统一的规范和标