

全国高职高专规划教材

计算机软件 技术基础

Computer
Fundamentals of Software

樊登焕 主 编
颜晶晶 杨端容 副主编



科学出版社

www.sciencep.com



全国高职高专规划教材

计算机软件技术基础

樊登焕 主 编
颜晶晶 杨端容 副主编

科学出版社

北 京

内 容 简 介

本书是讲述计算机软件技术的基础性书籍, 全书共分9章, 讲述了软件的概念、特征、分类, 线性数据结构, 非线性数据结构, 查找和排序, 操作系统基础, 典型操作系统平台下的编程模式, 数据库系统基础, 软件工程, 传统的软件开发方法等内容。

本书内容全面、概念清晰、实例丰富、图文并茂、重视基础、强调实用, 书后附有大量的实验和习题。本书既可作为高职高专计算机专业的教材, 又可作为初学者的入门教材, 还可作为工程人员的参考用书。

图书在版编目(CIP)数据

计算机软件技术基础 / 樊登焕主编. —北京: 科学出版社, 2004

(全国高职高专规划教材)

ISBN 7-03-012626-2

I. 计... II. 樊... III. 软件—高等学校: 技术学校—教材 IV. TP31

中国版本图书馆 CIP 数据核字(2003)第 120926 号

策划编辑: 李振格 / 责任编辑: 熊盛新

责任印制: 吕春珉 / 封面制作: 东方人华平面设计部

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>

双 青 印 刷 厂 印 刷

科学出版社发行 各地新华书店经销

*

2004年1月第一版 开本: 787×1092 1/16

2004年1月第一次印刷 印张: 14 1/4

印数: 1—5 000 字数: 323 000

定价: 20.00 元

(如有印装质量问题, 我社负责调换〈路通〉)

全国高职高专规划教材编委会名单

主任 俞瑞钊

副主任 陈庆章 周必水 刘加海

委员 (以姓氏笔画为序)

王雷 王筱慧 方程 方锦明 卢菊洪 代绍庆
吕何新 朱炜 刘向荣 江爱民 江锦祥 孙光弟
李天真 李永平 李良财 李明钧 李益明 余根墀
汪志达 沈凤池 沈安衢 张元 张学辉 张锦祥
张德发 陈月波 陈晓燕 邵应珍 范剑波 欧阳江琳
周国民 周建阳 赵小明 胡海影 秦学礼 徐文杰
凌彦 曹哲新 戚海燕 龚祥国 章剑林 蒋黎红
董方武 鲁俊生 谢川 谢晓飞 楼丰 楼程伟
鞠洪尧

秘书长 熊盛新

本书编写人员名单

主 编 樊登焕

副主编 颜晶晶 杨端容

编写人员 樊登焕 颜晶晶 郭 萍

杨端容 李 娜 陈雪芬

前 言

随着科学技术的飞速发展,计算机在经济与社会发展中的地位日趋重要。为满足高职院校计算机及相关专业教学的要求,加快我国高素质应用型人才培养的步伐,我们策划、出版了计算机专业的全国高职高专规划教材,本书是该系列教材之一。

本书力求用较短的篇幅,介绍计算机软件技术的基本知识,突出教学重点,重视基础,强调实用,书后附有大量的实验。突显高职教育注重实践操作能力培养的特点。

本书共分9章,主要内容如下:

第1章是概述,介绍软件的概念、特征、分类,软件的发展与危机,分析原因和解决途径;第2章线性数据结构、第3章非线性数据结构、第4章查找和排序是数据结构的内容,要求读者在学习数据结构内容时按要求完成相应的实验项目;第5章操作系统基础、第6章典型操作系统平台下的编程模式是操作系统的内容,要求学生在完成实验时,更注重自身能力的培养;第7章数据库系统基础,介绍数据库的基本概念和基本技术,为进一步应用数据库应用软件奠定基础;第8章软件工程、第9章传统的软件开发方法是软件工程的内容,让学生理解支持软件开发和维护的理论、方法、技术、标准以及计算机辅助工具和环境。

由于目前高职教育的课程和课时安排相差很大,我们将第6章和第7章加“*”作为可选教材,不同学校或不同专业可自做取舍。

本书由工作在教学第一线并有较丰富的教学和操作实践经验的多位教师共同编写。其中,第2章和4章由颜晶晶编写,第3章由郭萍编写,第5章由杨端容编写,第7章由李娜编写,第8章和9章由陈雪芬编写,其余部分由樊登焕编写。本书由樊登焕任主编,颜晶晶、杨端容任副主编。杨端容协助审稿,全书由樊登焕负责最后修改定稿。

本书内容全面、实例丰富、概念清晰、图文并茂,既可作为高职高专计算机专业的教材,又可作为初学者的入门教材。同时,本书的编写人员参加过大量的软件工程项目,书中融入了他们的工程实践经验,因此,本书也可作为工程人员的参考书。

本书虽经多次讨论并反复修改,由于时间仓促,加之作者水平有限,不当之处仍在所难免,敬请广大读者和同仁指正。

编 者

2003年10月

目 录

第 1 章 概述.....	1
1.1 软件的基本概念.....	1
1.1.1 软件的概念.....	1
1.1.2 软件的特征.....	2
1.1.3 软件分类.....	3
1.2 软件的发展和软件危机.....	4
1.2.1 概述.....	4
1.2.2 产生软件危机的原因和解决软件危机的途径.....	6
小结.....	7
习题.....	7
第 2 章 线性数据结构.....	8
2.1 线性表.....	8
2.1.1 线性表的逻辑结构及运算.....	8
2.1.2 顺序表.....	10
2.1.3 链表.....	14
2.1.4 线性表的基本应用.....	21
2.2 栈和队列.....	24
2.2.1 栈.....	24
2.2.2 队列.....	27
2.2.3 栈和队列的基本应用.....	32
2.3 串和数组.....	33
2.3.1 串的顺序和链式存储结构.....	33
2.3.2 数组的顺序存储结构.....	37
2.3.3 矩阵的压缩存储.....	39
小结.....	41
习题.....	42
第 3 章 非线性数据结构.....	45
3.1 树和二叉树.....	45
3.1.1 树的定义.....	45
3.1.2 二叉树.....	47
3.1.3 线索二叉树和排序二叉树.....	50
3.1.4 树与森林.....	51

3.1.5 哈夫曼树	52
3.2 图	55
3.2.1 图的遍历	58
3.2.2 图的应用	59
小结	62
习题	62
第4章 查找和排序	63
4.1 什么是查找	63
4.2 查找的方法	63
4.2.1 顺序查找	63
4.2.2 折半查找	64
4.2.3 分块查找	65
4.2.4 二叉查找树	66
4.3 散列表	68
4.3.1 什么是散列表	68
4.3.2 散列表的建立	68
4.3.3 散列表的查找	71
4.3.4 平均查找长度的计算举例	71
4.4 排序的有关概念	72
4.5 几种简单的排序方法	72
4.5.1 插入排序	72
4.5.2 选择排序	73
4.5.3 交换排序	74
4.5.4 归并排序	75
小结	76
习题	77
第5章 操作系统基础	79
5.1 系统概述	79
5.1.1 什么是操作系统	79
5.1.2 进程管理	83
5.1.3 内存管理	92
5.1.4 文件管理	97
5.1.5 外设管理	103
5.1.6 作业管理与用户界面	110
5.2 三种典型操作系统的特点及比较	111
小结	113
习题	113

*第 6 章 典型操作系统平台下的编程模式	115
6.1 DOS 操作系统编程模式.....	115
6.1.1 DOS 编程流程及开发工具.....	115
6.1.2 内存分配图.....	117
6.1.3 操作系统程序员接口.....	118
6.1.4 实例.....	119
6.2 Windows 编程模式.....	120
6.2.1 事件及消息机制.....	122
6.2.2 多线程.....	122
6.2.3 编程流程及开发包.....	123
6.2.4 程序员接口.....	124
6.2.5 实例.....	125
小结.....	129
习题.....	129
*第 7 章 数据库系统基础	130
7.1 概述.....	130
7.2 数据模型.....	131
7.3 数据库系统的结构.....	136
7.3.1 三级模式结构体系.....	136
7.3.2 数据库系统的体系构成.....	137
7.3.3 数据库管理系统.....	138
7.4 关系模型和关系代数.....	139
7.4.1 关系模型的数学定义.....	139
7.4.2 关系代数及关系运算.....	140
7.5 关系数据库理论.....	143
7.5.1 数据信赖.....	143
7.5.2 规范化.....	144
7.6 关系数据库标准查询语言 (SQL).....	147
7.6.1 DDL.....	148
7.6.2 DML.....	150
7.6.3 DCL.....	151
7.7 数据库的安全性和完整性.....	152
7.7.1 安全性和完整性.....	152
7.7.2 并发控制.....	154
小结.....	154
习题.....	155

第 8 章 软件工程	158
8.1 软件工程概述	158
8.1.1 基础知识	158
8.1.2 传统的软件工程模式与现代软件工程模式	159
8.2 软件生存周期	161
8.2.1 软件生存周期各个阶段的主要任务	161
8.2.2 开发模型	164
小结	166
习题	167
第 9 章 传统的软件开发方法	168
9.1 结构化开发方法概述	168
9.1.1 结构化软件开发的基本原则	168
9.1.2 软件开发方法	168
9.2 软件定义	170
9.2.1 系统分析	170
9.2.2 软件项目计划	170
9.2.3 需求分析	170
9.3 软件开发	174
9.3.1 软件设计	174
9.3.2 编码	180
9.3.3 测试	181
9.4 维护	184
小结	186
习题	187
附录	188
主要参考文献	217

第 1 章 概 述

本章要点

- 软件的基本概念。
- 软件的发展和软件危机。

本章难点

- 软件的特征。
- 软件的危机。
- 解决软件危机的途径。

随着计算机技术、电子技术的发展以及计算机与 Internet 的连接,人类社会进入了以计算机为核心的信息社会。在信息社会中,信息的收集、处理、交流和决策等都需要大量高质量的计算机软件,这样就促使人们对计算机软件的品种、数量、功能、质量、成本和开发时间等提出了越来越高的要求。为使软件的资源为人类所共享,人们越来越重视软件、软件开发及运行环境的标准化。计算机各类程序设计语言、多媒体技术和人机交互工具已被越来越多的人所掌握。本章介绍软件和软件工程的基本概念,以及软件的特征、分类、发展和出现的危机并分析其原因及解决的途径等。

1.1 软件的基本概念

1.1.1 软件的概念

软件 (Software) 一词在 20 世纪 60 年代初传入我国,最初,人们对软件的确切含义理解不深入,肤浅地把它理解为程序。随着计算机应用技术的发展和软件工程的出现,现在对它的一种普遍认同的解释为:计算机软件是与计算机系统操作有关的程序、规程、规则及任何与之有关的文档及数据。

计算机是由硬件和软件两部分组成的,软件是计算机系统中与硬件相互依存的另一半。如果没有软件的支持,任何高档的计算机都将一事无成,而普通计算机在基本配置的情况下,只需安装多种应用软件,就能拓宽应用范围,以解决更多的问题。这就是软件的作用。

软件由两部分组成:一是机器可执行的程序及有关数据;二是机器不可执行的,与软件开发、运行、维护、使用和培训有关的文档。

程序 (Program) 是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合,是软件开发人员根据用户需求开发出来的,是人们求解问题的逻辑思维活动的代码化描述。程序是在计算机中运行,要让计算机“读懂”人的意图,所以,程

序正确与否直接关系到求解问题的成败。如何做到让用户正确地运行程序，出现问题时能及时排除故障，让用户“读懂”程序是最彻底的解决办法。而文档资料是让用户能读懂程序、正确地掌握程序的重要媒介，文档记录软件开发的活动和阶段成果，是理解软件所必须的阐述性资料，它用于软件开发过程中的管理和运行阶段的维护，具有永久性地促进软件的开发、管理和维护，便于专业人员和用户之间的交流与沟通的作用。因此，与程序有关的各种文档与程序一样至关重要，许多国家都制定了详尽、具体的规定，颁布了各种规范和标准。我国也有《计算机软件开发规范》、《计算机软件需求说明编制指南》和《计算机软件测试文件编制规范》等。

1.1.2 软件的特征

软件是逻辑产品而不是物理实体。因此，软件在开发、生产、维护和使用等方面与硬件相比存在明显的差异。软件的主要特征如下。

1. 软件是一种逻辑产品

软件是逻辑产品而不是具体的实物实体。它具有抽象性，区别于计算机硬件和其他工程实体，可以把它记录在纸张上或保存在计算机存储设备中，但是，无法看到软件的形态，只能通过观察、分析、思考、判断等逻辑思维的方式来了解它的功能、性能和其他特征等。

2. 软件的开发与硬件不同

软件的开发过程无明显的“制造”过程，不像硬件一旦研制成功，可重复制造，在制造过程中进行质量控制，以保证产品质量。软件是通过人的智力活动，把知识和技术转化成信息的一种产品。软件一旦研制成功，可以大量复制生成同一产品的副本。因此，软件质量控制，必须把重点放在软件的开发过程中。

3. 软件产品质量的体现方式与硬件不同

硬件产品在整个生产过程中，从原材料的选购、设计图纸的绘制到每一个零部件的加工、生产的工艺、工序过程都有严格的质量控制和检验标准，不合格的零部件及产品通常在还没有流入到下一道工序前就被检测出来，从而可以有效地保证产品的质量。从另一方面讲，即使是定型的加工工艺、生产过程和定型的产品，也不能绝对杜绝废品产生，甚至产品的质量会因生产场地、操作人员的差异而不同。软件产品只有在实际问题求解过程中被证实是可行的且被用户接受了，才能成为产品，也才有存在的价值。软件产品一旦定型，其生产过程简单化，只是复制、拷贝而已。但是软件在设计、编程和实现过程中的各个阶段其质量难以保证和检验，只有用试运行的方式去检验才能证实其是否能用。由此可见，尽管软件产品开发与硬件产品开发有某些类似之处，但两者之间有着根本性的区别。

4. 软件产品的静态和动态属性

软件是由程序和相应文档组成的。程序是求解某个特定问题的算法和数据结构的代码集合，它具有双重属性。一方面，它是求解客观问题的逻辑描述，是供阅读和交流的，它的表示是静态的；另一方面，程序最终是通过运行去执行的特定操作和数据

处理，它又具有极其复杂和丰富内涵的动态属性。程序的双重属性决定了判别程序正确与否也要有双重标准。判定静态的程序正确与否是检查它的语法是否符合规则形式要求；动态的程序正确与否则要看动态的测试程序的所有逻辑流结构和数据结构是否正确。而后一种测试的难度和代价较之前的一种要大得多。

1.1.3 软件的分类

软件按不同的方式可以划分为不同的类型。

1. 按软件的功能划分

软件按功能通常被划分为系统软件和应用软件两大类。

- 系统软件是用以支持计算机系统的正常运行并实现用户提交的各种操作的那部分软件。系统软件是为计算机系统提供基本功能服务的程序集合，主要包括操作系统、网络系统、计算机语言编译（解释）系统以及实用工具软件系统等。
- 应用软件是为满足用户不同领域、不同问题的应用需求而提供的那部分软件。它可以拓宽计算机系统的应用领域，扩大硬件的功能，又可根据应用的不同领域和不同功能划分为若干子类。

2. 按软件的规模划分

按开发软件所需要的人力、时间以及源程序行数可以划分为 6 种不同规模的软件系统。

- 微型：一个人在几天之内完成的程序，程序不超过 500 行语句。
- 小型：一个人半年之内完成的、2000 行以内的程序。这种程序通常没有与其他程序的接口，但需要按一定的标准化技术、正规的资料书写以及定期的系统审查。
- 中型：5 个人以内在一年多时间内完成的、5000 到 10000 行的程序。计划、资料书写以及技术审查需要比较严格地进行。在开发中使用系统的软件工程方法是完全必要的，这对提高软件产品质量和程序人员的工作效率起着重要的作用。
- 较大型：5 至 10 个人在两年多的时间内完成的、50000 到 100000 行的程序。例如，编译程序、小型分时系统、应用软件包、实时控制系统等。
- 大型：100 至 1000 人参加用 4~5 年时间完成的、具有 1000000 行程序的软件项目。这种大型项目可能会划分成若干个子项目，每一个子项目都是一个较大型软件。例如，实时处理系统、远程通信系统、多任务系统、大型操作系统、大型数据库管理系统、军事指挥等通常具有这样的规模。
- 极大型：2000 人到 5000 人参加，10 年内完成的 10000000 行以内的程序。这类软件很少见，如在军事指挥、弹道导弹防御系统中使用的软件。

1.2 软件的发展和软件危机

1.2.1 概述

一、软件的发展经历

自从 1946 年第一台计算机问世以来,就有了程序的概念。可以认为程序就是软件的前身。软件伴随计算机技术的发展经历了程序设计、软件和软件工程 3 个不同的时代。

1. 程序设计时代 (1946~1955 年)

这个时代的计算机硬件的特点是:逻辑电路由电子管组成,计算机内存容量小,运行速度慢,外部设备少,系统稳定性差。这一时期大众只关心硬件的性能和指标;系统设计与实现是以硬件为中心,极不关心程序设计,编程处于从属地位。这时只有程序和程序设计的概念,程序设计的工具是机器语言、汇编语言、服务性程序;生产方式是手工生产、个体劳动;计算机的应用主要限于科学计算,程序规模小,结构简单,程序设计方法追求编程技巧,追求效率高、内存省。这使得程序不容易阅读、理解和修改,维护困难。但这时的程序设计方法尚能满足计算机应用的要求。

2. 软件时代 (1955~1970 年)

这个时代硬件已广泛采用晶体管和小规模集成电路,计算机内存容量增大,运算速度加快,外部设备较为齐全,运行稳定性高;计算机产量急剧上升,销售额爆炸性增长,程序员需求量猛增,造成软件人员奇缺。这时已提出软件概念,使用第二代语言如 Fortran, Algol, Cobol 等编译系统。操作系统实用化,数据库处于试验阶段;生产方式已进入作坊式的小集团合作方式。

3. 软件工程时代 (1970 年至现在)

20 世纪 60 年代后期,传统的软件开发方法不能适应大型软件的生产,导致软件危机。

人们想到用工程化的方法来生产软件,把注意力集中到软件开发的方法、技术和原理上,从此软件生产开始进入软件工程时代。人们首先提出新的程序设计方法;自顶向下、逐步求精的结构化程序设计方法、模块化设计方法等,称为第一代软件技术。在 1972 年到 1975 年,提出软件生存周期模型,把注意力集中到软件测试方面,提出了若干新的软件测试技术、测试方法、测试原理以及软件确认和验证的理论,这些称为第二代软件技术。1976 年至 1980 年开始提出了若干处理需求定义方面的技术,这些称为第三代软件技术。80 年代后,人们开始把软件工程各阶段的工具集成到一起,形成软件开发环境,更有效地支持软件开发的工程化,使软件产品质量和可靠性得到提高。

二、软件危机

软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。20 世

纪 60 年代末至 20 世纪 70 年代初,“软件危机”一词在计算机界广为流传。事实上,几乎从计算机诞生的那一天起,就出现了软件危机,只不过到了 1968 年才被人们普遍认识到。

软件危机主要体现在以下 4 个方面。

1. 软件成本日益增长

在计算机发展的早期,大型计算机系统主要是被设计应用于非常狭窄的军事领域。在这个时期,研制计算机的费用主要由国家财政提供,研制者很少考虑到研制代价问题。随着计算机市场化和民用化的发展,代价和成本就成为投资者考虑的最重要的问题之一。20 世纪 50 年代,软件成本在整个计算机系统成本中所占的比例为 10%~20%。但随着软件产业的发展,软件成本日益增长。相反,计算机硬件随着技术的进步、生产规模的扩大,价格却不断下降。这样一来,软件成本在计算机系统中所占的比例越来越大。到 20 世纪 60 年代中期,软件成本在计算机系统中所占的比例已经增长到 50% 左右。

2. 开发进度难以控制

由于软件是逻辑、智力产品,软件的开发需建立庞大的逻辑体系,这是与其他产品的生产不一样的。例如,工厂里要生产某种机器,在时间紧的情况下可以要工人加班或者实行“三班倒”,而这些方法都不能用在软件开发上。在软件开发过程中,用户需求变化等各种意想不到的情况层出不穷,令软件开发过程很难保证按预定的计划实现,给项目计划和论证工作带来了很大的困难。BROOK 曾经提出:“在已拖延的软件项目上,增加人力只会使其更难按期完成”。事实上,软件系统的结构很复杂,各部分附加联系极大,盲目增加软件开发人员并不能成比例地提高软件开发能力。相反,随着人员数量的增加,人员的组织、协调、通信、培训和管理等方面的问题将更为严重。

3. 软件质量差

软件项目即使能按预定日期完成,结果却不尽人意。1965 年至 1970 年,美国范登堡基地发射火箭多次失败,绝大部分故障是由应用程序错误造成的。程序的一些微小错误可以造成灾难性的后果,例如,有一次,在美国肯尼迪发射一枚阿脱拉斯火箭,火箭飞离地面几十英里高空开始翻转,地面控制中心被迫下令炸毁。后经检查发现是飞行计划程序里漏掉了一个连字符。就是这样一个小小的疏漏造成了这支价值 1850 万美元的火箭试验失败。在“软件作坊”里,由于缺乏工程化思想的指导,程序员几乎是习惯性地以自己的想法去代替用户对软件的需求,软件设计带有随意性,很多功能只是程序员的“一厢情愿”而已,这是造成软件不能令人满意的重要因素。

4. 软件维护困难

正式投入使用的软件,总是存在着一定数量的错误,在不同的运行条件下,软件就会出现故障,因此需要维护。但是,由于在软件设计和开发过程中,没有严格遵循软件开发标准,各种随意性很大,没有完整的真实反映系统状况的记录文档,给软件维护造成了巨大的困难。特别是在软件使用过程中,原来的开发人员可能因各种原因

已经离开原来的开发组织,使得软件几乎不可维护。另外,软件修改是一项很“危险”的工作,对一个复杂的逻辑过程,哪怕做一项微小的改动,都可能引入潜在的错误,常常会发生“纠正一个错误带来更多新错误”的问题,从而产生副作用。

1.2.2 产生软件危机的原因和解决软件危机的途径

一、产生软件危机的原因

从软件危机的种种表现和软件作为逻辑产品的特殊性可以发现软件危机的原因:

1. 需求不明确

在软件开发过程中,用户需求不明确问题主要体现在如下4个方面:

- 在软件开发出来之前,用户自己也不清楚软件的具体需求。
- 用户对软件需求的描述不精确,可能有遗漏、有二义性,甚至有错误。
- 在软件开发过程中,用户还提出修改软件功能、界面、支撑环境等方面的要求。
- 软件开发人员对用户需求的理解与用户本来愿望有差异。

2. 缺乏正确的理论指导

缺乏有力的方法学和工具方面的支持。由于软件不同于大多数其他工业产品,其开发过程是复杂的逻辑思维过程,其产品极大程度地依赖于开发人员高度的智力投入。由于过分地依靠程序设计人员在软件开发过程中的技巧和创造性,加剧软件产品的个性化,也是发生软件危机的一个重要原因。

3. 软件规模越来越大

随着软件应用范围的扩大,软件规模愈来愈大。大型软件项目需要组织一定的人力共同完成,而多数管理人员缺乏开发大型软件系统的经验,而多数软件开发人员又缺乏管理方面的经验。各类人员的信息交流不及时、不准确、有时还会产生误解。软件项目开发人员不能有效地、独立自主地处理大型软件的全部关系和各个分支,因此,容易产生疏漏和错误。

4. 复杂度越来越高

软件不仅仅是在规模上快速地发展扩大,而且其复杂性也急剧地增加。软件产品的特殊性和人类智力的局限性,导致人们无力处理“复杂问题”。所谓“复杂问题”的概念是相对的,一旦人们采用先进的组织形式、开发方法和工具提高了软件开发效率和能力,新的、更大的、更复杂的问题又摆在人们的面前。

二、解决软件危机的途径

人们在认真地研究和分析了软件危机背后的真正原因之后,得出了“人们面临的不光是技术问题,更重要的是管理问题,管理不善必然导致失败”的结论,便开始探索用工程的方法进行软件生产的可能性,即用现代工程的概念、原理、技术和方法进行计算机软件的开发、管理和维护。

软件工程是用工程、科学和数学的原则与方法研制、维护计算机软件的有关技术

及管理方法。软件工程包括如下3个要素:

- 方法。软件工程方法为软件开发提供了“如何做”的技术,是完成软件工程项目的手段。
- 工具。软件工具是人类在开发软件的活动中智力和体力的扩展和延伸,为软件工程方法提供了自动的或半自动的软件支撑环境。
- 过程。软件工程的过程则是将软件工程的方法和工具综合起来以达到合理、及时地进行计算机软件开发的目的。

迄今为止,软件工程的研究与应用已经取得很大成就,它在软件开发方法、工具、管理等方面的应用大大缓解了软件危机造成的被动局面。

小 结

软件是计算机系统重要的组成部分,是计算机的无形部分,离开软件计算机将不能做任何事情。

软件分系统软件和应用软件两部分。系统软件是计算机系统的必配软件,在购置计算机时根据需要由厂家提供。系统软件包括:操作系统、计算机语言编译系统,以及工具软件系统。应用软件是指除系统以外的所有软件,是解决各种实际问题的计算机程序。

软件是不同于硬件的逻辑产品,软件有它自身的特性。只有认识了软件的特性,才能认识软件开发的内在规律,才能把握软件开发的自由。

由于信息社会对软件的需求日益增加,而软件开发方式还很落后,效率低、成本高、质量无法保证,因此产生软件危机。为解决软件危机,由此诞生了一门计算机新兴学科——软件工程学。

软件工程是从工程角度来开发软件的方法和技术。采用工程化的方法和技术开发软件,可以提高软件的可靠性、可维护性、可移植性,它的目标是在给定时间和费用的条件下开发出一个满足用户功能和性能要求的、可靠的软件系统。

习 题

1. 软件的发展经历了哪几个时代?
2. 软件危机是什么?体现在哪几个方面?解决途径是什么?