高等学校教材

软件设计基础

崔俊芝 杜 藏 黄思曾 於崇华

高等教育出版社

高 等 学 校 教 材

软件设计基础

崔俊芝　杜藏　黄思曾　於崇华

高等教育出版社

(京)112号

JS582/17

内 容 提 要

本教材是按照国家教育委员会“八·五”出版计划的安排和高等学校数学与力学教学指导委员会计算数学教材建设组提出的教材大纲编写而成的。目的是培养学生具有独立设计和开发科学与工程软件的能力。主要讲述软件设计的基础知识、方法、实用技术。内容包括：数据结构与非数值算法分析，结构化程序设计方法和表现技术，软件开发的工程化方法和技术等。

本书可作为计算数学及其应用软件专业和其他非计算机专业的计算机基础课教材，亦可供有关专业的教师和科技人员阅读、参考。

图书在版编目（CIP）数据

软件设计基础/崔俊芝等编著. -北京：高等教育出版社，1995

高等学校教材

ISBN 7-04-005385-3

Ⅰ.软… Ⅱ.崔… Ⅲ.软件-程序设计-基本知识-高等学校-教材 Ⅳ.TP311

中国版本图书馆CIP数据核字（95）第18580号

*

高等教育出版社出版

北京沙滩后街55号

邮政编码：100009 传真：4014048 电话：4054588

新华书店总店北京发行所发行

北京市顺新印刷厂印装

*

开本 787×1092 1/16 印张 17.25 字数 430 000

1995年10月第1版 1995年10月第1次印刷

印数0001—3 500

定价：12.35元

前　言

随着计算机科学和技术的发展，以及计算机广泛深入地应用于各种科学和工程技术领域，计算机与软件科学知识和技术已经成为众多科学和工程专业的基础课。但是非计算机专业对这些知识和技能的需求，与计算机有关专业相比，无论是深度还是广度，都有较大区别。尽管现在已经出版了许多有关计算机科学的教材，但由于学科分割过细且各书自成体系，仍然没有一套适合于非计算机专业的、有机地综合了计算机系统科学领域里多种实用知识和技术的、比较精练的教材。

本书是一本有关现代计算机软件设计方法的基础课教材。它属于一套四本教材中的一本，其它三本是《计算机系统原理》、《数据结构与算法》和《软件工程方法》。这套教材由高等学校数学与力学教学指导委员会计算数学教材建设组组织编写，除了本书已经通过教学指导委员会和有关专家审查并已出版外，《计算机系统原理》已经由高等教育出版社出版。

计算机与软件科学和技术是一个知识极为丰富、分支学科很多、更新发展十分迅速的科学技术领域，精练地选择一族满足非计算机专业需求的关于计算机系统与软件的知识和技术，并把它们有机地组织在一套书中，显然不是一件容易的事情，但是鉴于“计算数学及其应用软件”专业建设的需求，鉴于众多的非计算机专业的老师和同学们的渴望，促使我们下决心尝试完成编写这套教材的任务。

对于这套教材，我们是基于如下原则选择内容和实施系统组织的：以知识和技术的基础性、实用性、先进性和系统性为主，局部地照顾知识体系的完整性；不追求高深的理论体系，但应使基本的理论和方法、先进实用的知识和技巧兼备；为了减轻学生负担，使学生易于理解和掌握，并避免重复讲授，在这套教材中，坚持了循序渐进，避免了同一水平、同一功能的知识和技术的重复出现。

本书是针对应用软件开发，特别是科学和工程应用软件开发，而编写的软件设计基础，目的是使读者了解和掌握从事软件设计的基本知识和技术，具有开发高品质应用软件产品的能力。全书分十章。第一章概论，为全书的引言、内容的浓缩和纲要；第二～四章主要讲述数据表达、数据结构以及非数值算法方面的基本知识和方法；第五、六章主要讲述结构化程序设计的基本理论、方法和表现技术；第七章介绍软件开发的工程化方法，第八章主要讨论数值软件开发方法；为了使读者通过本书能够掌握一些实用可靠的软件资源和工具，并了解软件开发方法论的现状及其发展动向，我们特意选编了第九、十两章。以上主要材料分别选自于数据结构、数据库系统原理、程序设计方法学、算法设计与分析、软件工程、系统分析和设计等各种计算机科学和技术的专著，在融汇了作者及其同事们多年来在软件开发和软件教学中所积累起来的经验和成果之后，我们努力地将其有机地组织成一本系统的教材。为了便于叙述，书中采用了类PASCAL、类C和类FORTRAN的表述形式。建议本书的讲述学时为60～80学时；在教学过程中应该安排一个有一定规模的软件开发课题，以促进学生掌握所学内容。

本书的第一～三章由中山大学黄思曾先生执笔，第四～六章由南开大学杜藏先生执笔，第七～九章由复旦大学於崇华先生执笔，第十章由黄思曾和於崇华先生共同执笔；本书从立题、

选材到编写组织，包括总体规划、章节安排和内容取舍，都是在崔俊芝组织和直接参与下进行的。初稿完成后，於崇华对全书的格式和行文进行了统一调整和润色；杜藏在南开大学进行了试讲，并根据试讲情况进行了局部修改；定稿前，崔俊芝又对全书进行了逐句的审阅和修改。编著者署名是按姓氏在字典中的位置排序的。

在本书的立题和编写过程中，我们得到了李岳生、李荣华、应隆安、孙彻、黄友谦等教授的关心和指导，还得到了南开大学诸位老师的大力支持；在本书初稿完成后，南京大学的苏煜城、郑国梁、庄建南等教授对全书进行了认真审阅，提出了许多中肯的修改意见，在此一并向他们表示衷心感谢。

我的三位同事，杜藏、黄思曾、於崇华，在本书的编写过程中都是非常精心和认真负责的，他们对本书倾注了全部心血，表现出人民教师的高尚品质和献身精神，他们是本书的主要贡献者。虽然本书在定稿前已经过了多次修改，但由于我们水平所限，在进行跨学科写作中，错误和缺陷在所难免，敬请讲授本书的老师和读者多提宝贵意见。随着计算机科学和技术的发展，本书的部分内容很快就需要更新和修改，盼望大家多提意见，以便能在本书再版之时不断改进，使您更为满意。

谢谢！

崔　俊　芝

1995.9 于天津

目　录

第一章 概 论

§1 计算机软件

1.1 从程序到软件

电子计算机是由于科学技术发展的需要而出现的。第二次世界大战期间，各种新型火器的弹道计算和原子弹的研制提出了大量的数学、物理和工程方面的计算问题，迫切需要解决快速计算的工具，第一台电子计算机 ENIAC（Electronic Number Intergrator And Calculator）正是在这种情况下于 1946 年应运而生的。

事实上，计算机的唯一功能是数据处理：把输入的原始数据加工变换为含有特定信息的另一种数据。为了使计算机实现预期的加工变换目的(如求解某一计算问题或控制某一过程)而编排的，能为计算机识别、理解和执行的一系列有序的代码步骤的总体称为程序。计算机程序就是用某种约定的记号(语言)表达的数据对象和操作命令的集合，计算机依靠对程序的处理和执行来完成某项约定的数据处理任务。

早期，人们把用计算机的机器指令编的程序仅仅看成是计算机的使用说明。直至 1947 年，Von Neumann 首先提出了用流程图描述计算机的运行过程，从那时起大家才开始认识到，程序是计算机中不同于硬件的一个组成部分，程序设计是不同于硬件研制的一种开发研究工作，因此，编写计算机程序就成为计算机应用的核心。

长期以来，人们以极大的努力来研究程序的设计方法，把所涉及的课题发展为计算机科学的一个又一个的分支，但时至今日，看来研究的速度还赶不上计算机应用的突飞猛进的要求。不过总的说来，新的理论在不断地提出，新的方法也层出不穷，一个认知体系正在逐步形成。

在 60 年代之前不存在程序设计的系统化方法。人们以个人的手工技艺方式来编写程序，程序员专注于各种不规范的、很大程度上有赖于灵感的编程技巧。除了保证程序的正确动作外，特别强调时间和空间效率，往往绞尽脑汁只是为了缩短几条指令。程序的设计过程保留在程序员的头脑当中，提交的工作结果一般只是源程序文本，而这样的文本常常是其他人甚至程序员自己事后都难以读懂的。

60 年代之后，随着计算机应用的发展及一些高级语言的出现和完善，程序日益庞大复杂，迫使程序的编写不得不采用合作方式，因而就需要用方法和规则来协调和制约程序员们的集体工作。于是，人们开始研究一些和程序设计过程密切关联的方法。由此而产生的程序设计方法学主要关注数据对象的抽象表达、算法的设计和分析以及程序的合理结构。

进入 70 年代后，程序的规模和复杂程度以空前的速度猛增，以致往往被称为“程序系统”。大的系统的源程序行数逾百万，工作量超过几百人年，因此，程序的开发遇到了前所未有的困难。作坊式的组织和开发手段在这些新问题面前一筹莫展，束手无策，现在常用“软件危机”来表达当时程序编写人员所面临的困境。危机促使人们开拓新的思路——用工程化的方法来开发程序，从而迎来了软件工程学的诞生。

众所周知，计算机系统由硬件和软件两部分组成。一般情况下，人们习惯于把计算机运行

涉及的有关程序称为软件，但是软件工程学为了强调工程化的开发方法，认为“软件”并非是“程序”的同义词，软件是与计算机系统的操作有关的程序、规程、规则及任何与之有关的文档和数据。

所以，软件是程序的完善和发展，是用一定的规范化方法开发、经过较严格的测试和试用、文本相对稳定，而且具有一系列完整的文档资料的计算机程序。一般说来，计算机软件具有商品性，可以在市场上出售和购买。

1.2 软件的分类

按照软件与计算机硬件和用户的关系，可以把它分为系统软件、支持软件和应用软件三大类，其层次结构关系如图 1.1 所示。

(1) 系统软件

系统软件和计算机硬件的关系最为密切，其核心是计算机的操作系统。它的主要功能是实施硬件资源(CPU、内存、I/O 设备)和软资源(外存数据)的管理，管理的目的是提高计算机系统资源的利用效率并方便用户的使用。操作系统扩充了计算机的功能，用户使用计算机时必须利用操作系统用户界面上定义的手段，如键盘命令或作业控制语言(JCL)来表达用户的作业要求，换句话说，用户面对的实际上是一台操作系统虚拟机器。

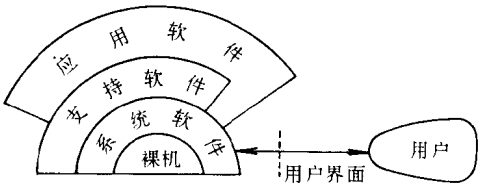

图 1.1 软件分类的层次结构

除了操作系统之外，一些具有基础性、公用性的软件，如系统诊断软件、网络通信软件等，也可以列入系统软件之中。

(2) 支持软件

支持软件也叫支撑软件，亦具有公用性的特点，其主要功能是支持用户开发或运行自己的应用软件。而支持软件本身的运行亦需要系统软件的支持。所以，它是介于系统软件和应用软件之间的一类软件。

典型的支持软件如：

- 语言处理软件。例如各类程序设计语言的编译程序、连结程序、解释程序、诊断程序等。
- 数据库管理系统(DBMS)。
- 软件工具及服务性的实用程序。例如在软件的开发、维护和管理中提供各种独立的支持服务的程序。
- 软件开发与运行环境。软件开发方法和软件工具的结合体，对软件开发的过程提供集成性的支持服务，是一类正在迅速发展的先进的支持软件。

(3) 应用软件

为了把计算机应用于各行各业的数据处理工作，便需要有各种相应的应用软件。对于具有共同性处理的一类工作，如文字处理、会计业务、统计分析、科学计算、绘图等，可以开发通用的应用软件投入市场供用户选择，通常把它们称为应用软件包。在数据处理比较特殊、无法找到适用的现成的应用软件包时，就必须按实际需要自行开发专用的应用软件了。

按应用领域的工作性质来分，应用软件又可以分成下面三大类：

①科学工程软件

这类软件的特征是以科学研究和工程技术问题中的数值计算方法为中心。任何科技问题都离不开数值计算，因此这类软件是科学家和工程师的极重要的工具。科学工程软件是数值计算方法的研究成果，也是把它们转化为生产力的关键环节。同时，随着软件的开发，又会提出新的计算方法课题，成为促使数值计算方法研究发展的动力和源泉。

科学工程软件可再细分为几类：

·数学软件

数学软件是计算数学标准算法程序的总称。每个数学软件是实现一个特定的计算方法的程序模块，由若干个标准过程组成。大量的数学软件组成了各种数学软件包，如 QUADPACK（数值积分软件包）、ITPACK（线性方程组迭代解法软件包）、EISPACK（特征值问题解法软件包）等，都已在被广泛使用着。

·结构工程软件

结构工程软件主要用于各种结构的强度分析，属于计算固体力学的应用范畴。这类软件大都以有限元方法和矩阵结构分析方法为主体，外加适当的前后处理程序。比较知名的有 NASTRAN（综合的通用有限元软件）、ANSYS（线性和非线性分析系统）等。

·CAD/CAM/CAE 软件

计算机辅助设计/制造/工程系统是综合利用计算机的工程计算、逻辑处理、数据处理、图形处理功能和人的经验所形成的一类应用软件，用来帮助各种产品或工程的设计和制造工作。一个 CAD/CAM/CAE 软件通常包含如下四部分：面向专业技术人员的用户接口处理系统；相关专业的工程计算软件；工程数据库及其管理系统；功能强大的图形处理系统。

·专家系统

专家系统是一个以某特定领域内人类专家的知识水平来解决该领域问题的应用软件。专家系统里存储着经过抽取和整理的该领域专家头脑中的知识，并建立了推理机制，使它能够模拟人类专家思维和决策的智力运转过程，从而对该领域的问题做出推理、决策、咨询和推荐。

在科学工程应用方面，较著名的专家系统有 DENDRAL（分子结构分析）、MACSYMA（数学公式处理）、SACON（结构工程分析）、BACON（物理定理证明）等。

·专业性的技术软件

在特定专业领域上使用的应用软件，如电网调度软件、地球物理勘探软件、电站仿真软件等。它们之间在结构上有较大的差异，其内容除了数值计算外，还涉及到大规模的科技数据处理和实时控制等问题。

·辅助性软件

这是指支持科学和工程应用软件开发和运行的一类共享性软件，如图形显示软件和工程数据采集管理系统等。

②事务处理及信息管理软件

这类软件使得各行各业的数据处理事务计算机化，能完成数据的采集、编辑、存储、加工、输出等任务，它通常是使用一个面向企业的数据库来存储浩如烟海的数据。

计算机系统总数的三分之二是用作非数值的数据处理的。所谓“非数值”是指在这类任务中，对数据的运算比起科学计算来要远为简单。人们正越来越多地用事务处理软件来代替人工数据处理过程，如用管理信息系统（MIS）来产生企业管理所需要的各种信息；而决策支持系

统（DSS）则把数学模型并入信息系统软件，能够支持相对非结构化的决策活动，为高级管理人员提供有用的信息；利用图象处理软件将卫星传送回来的图片信息重组为清晰的图象；图书资料和科技情报检索系统则极大地减轻了科研人员查找文献资料的负担。

③过程控制软件

由于计算机的运算速度快且计算精度高，因而已成为过程控制系统的核心部件。在如导弹、人造卫星、现代军用飞机之类要求反应速度极快、控制精度极高或者人无法亲自操作的过程控制系统中，计算机的过程控制软件正在大显身手，发挥着神经中枢的作用。此外，现代大工业生产的规模越来越大，工艺流程和操作控制日益复杂，使计算机在生产过程的自动化控制中扮演着越来越重要的角色。

过程控制软件一般先把被控制过程送来的信号变换为离散数字信号，再按预定的控制规律加以处理，最后将结果反馈给被控制过程的接收元件，依靠状态的调节来实现控制。

§2 软件设计方法的内涵

软件设计是一门实践性极强的学科，经过近 40 年的开发实践，已经总结出一整套软件设计应遵循的方法和原则。事实证明，只有在正确的方法论指导下不断实践，才能提高软件设计的能力和软件产品的质量。本书将重点讨论数据的表达方法、算法设计与分析、程序构造方法和软件开发的工程化方法等四个在软件开发过程中必然要面对的问题。本节将对这些问题作一个概貌性的介绍。

2.1 软件开发的工程化

1. 方法特征

从 50 年代到 70 年代，软件设计方式经历了从手工技艺式到小组合作式再到工程化方式的演变。70 年代以来，一类工程化的软件开发方法已逐步形成。透过形形色色的具体方法，可以把软件开发方法的共同特征概括如下：

（1）在软件的整个开发过程中要建立各种计划，如进度、质量控制、软硬件配置、测试等，各级管理人员必须要严格按计划进行管理。

（2）把软件开发过程划分成若干阶段，每个阶段有相对独立的工作目标。阶段内采用有效的开发方法，但不论使用哪一种方法，开发过程都必须遵循标准规范。规范明确规定每个阶段的任务、工作步骤和完成标志，以指导和约束开发人员遵照统一的方式来理解和处理开发过程。

（3）建立相当于现代化工业大生产方式的产品质量检查制度，对软件开发的每个阶段都进行技术和管理方面的评审。阶段评审可以尽早发现开发中的错误和偏差，减少纠错和修正的耗费代价，保证软件产品的质量。

（4）开发的每一阶段都要产生规定的文档资料。文档是开发过程中的记录、交流、控制和管理的主要手段，也是日后维护软件的重要依据；文档又是阶段性评审的主要对象，只有交出该阶段的符合规范的、通过复审了的全部文档，才意味着该阶段工作的结束。

（5）提供从原始计划到最终产品的一条可追溯的途径，以适应开发过程中的各种变更。

2. 软件的生存周期

软件的生存周期是指从开始构思软件产品到产品不能再使用为止的时间间隔。与此相关的一个概念是软件开发期，它是指从决定开发软件产品开始到产品交付使用为止的时间间隔。

软件开发的生存周期方法学从时间进程的角度把软件的开发和维护过程划分成若干个时期，如计划时期、开发时期和运行时期。每个时期可以再细分成一些工作阶段，典型的包括需求分析阶段、设计阶段、实现阶段、测试排错阶段、安装和验收阶段、运行和维护阶段，有时还包括退役阶段（此时，对软件产品的支持将被终止）。

软件生存周期是软件工程化开发方法的最基本的概念之一，各种开发的模式和方法都在不同程度上接受和承认了这一概念。虽然各种具体的开发规范对工作阶段的划分表面上略有差异，但其本质上都是完全一致的。由于各个阶段的工作任务相对独立且工作的性质相近，从而把软件的开发和维护这一整体任务分解成了一个个子问题，使得各类人员可以分工协作各个击破，这样就降低了整个开发工程的难度。随着开发周期的时间推进，需要解决的问题通过有限个步骤从抽象逻辑概念被逐步转化成具体的物理实现，直到最终产生源程序。

（1）计划时期

计划时期是软件生存周期的开始时期，这个时期的重要任务是确定软件开发任务的总目标，确定是否存在可行的系统解决方案及在确定开发任务进行的前提下制订出开发工程的人力、资源和进度计划。

计划时期可再细分成几个阶段：

a）问题定义

确定要解决的问题，并经过初步调查研究，明确问题的性质、工程的目标和规模。

b）可行性研究

确定问题是否可解。经过粗略然而全面的系统分析和设计之后，明确软件系统功能和性能方面的需求，确定系统结构、接口形式和资源配置，从而提出若干个系统方案。对每一个解决方案都要从技术上、经济上、社会因素上分析其可行性，即能不能做和应不应该做。

c）制订开发计划

在决定采用某个经可行性论证的方案后，制定相应的人、财、物诸资源和开发进度计划。

（2）开发时期

经过计划时期确定了软件开发可行之后，软件进入了开发时期，在开发时期中将通过分析和设计最终实现预想的软件系统。这一时期可按开发任务的变迁再细分为几个阶段：

a）需求分析

从调查用户要求和软件的运行环境入手，详细定义软件的功能、性能、输入输出、外部接口、安全等方面的要求。通过分析回答：软件究竟要做什么？

b）系统设计

制定系统开发原则，确定系统事务处理流程、软件的总体结构、全局的存储数据结构和其它方面（如输入输出、用户接口、安全控制等）的总体设计规划，从而回答软件应该怎样做才能满足已确定的需求规定。

c）详细设计

给出软件的每一个构成元素，如程序模块、存储数据结构、输入输出数据格式等的实现方案的详细策划。如按照系统设计时确定的各个程序模块的功能和接口定义，设计模块内部的算

法过程和逻辑结构，编写程序编码说明，设计输入输出数据格式等。

d）编码

按照详细设计，产生软件在预定运行环境中的源程序，即一般意义的“写程序”。

e）测试和排错

设法找出程序中的错误并加以修正，目的是产生一个可用的软件版本。测试时要对每个模块作单元测试，对全系统作组装测试，还要在接近真实运行环境的状态下进行确认测试。设计合适的测试用例并动态地执行被测试程序是测试的主要手段。

(3) 运行时期

软件开发完成交付使用后就进入了运行时期，这个时期的主要工作是软件维护。尽管软件的使用是无“磨损”的，但却面临极繁重的维护任务。维护可以是校正性的（对运行中暴露的差错作修正），也可以是完善性的（更改或扩充软件以满足新的需求）。维护工作将持续到不开发新的软件已不能满足用户需要时为止，此时现有软件系统退役，结束了它的生命周期，而新的软件又开始了一轮新的生命周期。

3. 软件开发的规范

工程化的软件开发方法的实施强烈地依赖于开发过程的规范化。所谓“规范化”是指制订一系列指导和约束开发人员生产软件产品的标准规程，用于协调和管理开发的全过程，使开发人员有所遵循和制约，从而在一定程度上保证软件的优质高产。

一个软件开发规范应该规定开发人员“什么时侯应该做什么以及如何去做”，即三 W（WHEN－WHAT－HOW）。开发规范越明确具体，对实现优质高产地开发软件的目标的指导作用就越大。不过软件毕竟是一种逻辑性的产品，软件开发规范难免具有一定的概略性和灵活性，这将在一定程度上削弱其指导作用。因此在某个具体的软件组织和某项具体的软件开发任务中，还必须制订一些与某种规范精神原则一致的，然而更具体明确的实施细则，使开发人员随心所欲的自由度受到约束，保证开发中心的高度一致。

国家标准局在 1988 年颁布了《计算机软件开发规范》(GB8566－88)。这个规范采用了瀑布式的开发模式，把软件生存周期划分为如下八个阶段：

(1) 可行性研究与计划阶段

(2) 需求分析阶段

(3) 概要设计阶段

(4) 详细设计阶段

(5) 实现阶段

(6) 组装测试阶段

(7) 确认测试阶段

(8) 使用和维护阶段

规范详细规定了每一阶段的任务、实施步骤、实施要求、完成标志以及应交付的文档资料。

规范指出，软件产品从形成概念开始，经过开发、使用和不断增补修改，直到最后被淘汰的全过程中，应提交 14 种文档资料。由于工作阶段的明确划分和软件文档的具体定义是规范的实质性内容，因此可用“8 个阶段、14 种文档”来概括国家规范的本质。

4. 软件文档

文档（document）是指与程序开发、维护和使用有关的材料，也就是对开发的活动、需求、

过程或结果进行描述、定义、规定、报告或认证的任何文字或图示的信息。软件开发的工程化方法特别强调文档的产生和作用，认为文档是软件中不可缺少的一个重要部分，只讲程序不讲文档的概念在60年代后早已被人们摒弃了。

软件文档大体上可分为两类：一类是开发过程中填写的各种图表，称为工作表格；另一类是应编制的技术资料和技术管理资料，称为技术文件。

从编写形式上讲，可以使用自然语言、特别设计的形式语言、介乎二者之间的半形式语言（如结构化的描述语言）及各种图形和表格来编制文档，可以手工书绘，也可以在计算机支持系统中产生。但不管怎样，文档必须是可阅读的，且具有永久性。

文档的重要意义可以从以下几个方面来认识：

(1) 文档是记录手段。开发过程中的各种信息必须以可读形式保留下来，供开发、维护、管理活动使用。

(2) 文档是交流手段。在软件开发过程中，专业人员和用户之间，职责不同的专业人员之间都需要交流，只有凭借文档，交流才是全面而准确的。

(3) 文档对软件开发过程起着重要的控制作用。某一类文档的产生是某个开发阶段工作的成果和结束标志，也是下一阶段工作的依据和约束。

(4) 文档是开发过程的管理依据。任何质量保证体系，都必须依靠各类文档记录的信息来监督和协调生产过程。

(5) 文档是软件维护的依据。只凭源程序而缺乏开发文档作为参照，维护工作是难以进行的。

(6) 文档是软件的介绍媒介。阅读有关文档可以深入全面地认识和评价文档，尤其象操作手册一类的文档更是使用软件时不可缺少的资料。

从某种意义上说，软件文档是软件开发规范的体现和导引，按规范要求生成一整套文档的过程也就是按规范要求完成软件开发任务的过程。若有谁把文档编制视为写程序之外的负担，或者是开发完成之后有需要才补写的说明资料，甚至认为源程序就是全部文档，那只能说明他并没有掌握软件工程化开发方法的基本思想，其做法还停留在60年代以前的水平。

国家标准《计算机软件开发规范》(GB8566－88)规定了在开发全过程中应提交的14种文档，与之相关的另一个标准《计算机软件产品开发文件编制指南》(GB8567－88)则给出了这14种文档的编写大纲。

14种文档中有9种属于开发技术档案，它们是可行性研究报告、软件需求说明书，数据要求说明书、概要设计说明书、详细设计说明书、数据库设计说明书、模块开发卷宗、测试计划、测试分析报告。

其余5种是管理或说明性的资料，它们是项目开发计划、开发进度月报、项目开发总结报告、用户手册、操作手册。

考虑到不同软件在规模和复杂程度上的差别，国家规范允许软件文档的编制有一定的灵活性。对于一般规模的软件项目，可将产生的文档简略为以下5种：

(1) 可行性研究报告

(2) 软件需求说明书

(3) 系统设计说明书

(4) 模块开发卷宗

(5) 测试计划和测试分析报告

实践表明，这对大部分情况已足够使用了。

70年代以来，“软件工程学”作为一个崭新的学科已被确立，内容也被不断地充实，研究范围已从软件开发技术扩展到软件工程的管理方面。与软件开发技术的研究集中在软件开发方法、软件工具和软件开发环境等课题上不同，软件管理学关注的是在软件开发和维护过程中涉及的计划管理、成本管理、质量管理和组织管理等问题，旨在更合理地使用人力和物力资源，协调各种活动，按进度和预算完成开发计划并确保软件产品的质量。

2.2 数据对象的表达方法

“数据”这个术语在计算机科学中是用来表示客观事物属性的记录的，如一个人的姓名、地址和电话号码；一条梁的截面尺寸等都是数据。与此相关的另一个概念是“信息”，它指的是所观察事物给出的知识。可以说，信息是数据的内涵，数据是信息的录制形式。但在实际中并不需要处处严格区分“数据”和“信息”这两个术语的含义。

计算机内一切信息都是用各种物理元件的两种稳定的可转换的状态来表达的，如高电平和低电平、磁通量的两种方向等，我们用二进计数制的两个数字记号0和1来表示。

现实世界的对象如此众多和复杂，而机内的最终表示记号又是如此简单，因此就必须建立一套逐层抽象的数据表达方法学，其机制是：

• 从要表达的目标(现实对象)到表示手段(机内表记)划分若干个不同的抽象层次，各层次表示的抽象程度渐次降低。

• 每个抽象层次上分别定义一套完整的表达概念和术语，用以表示客观事物。高层次的表示较为接近人们对事物的认识，低层次的表示较为接近机器的表示手段。

• 层次间有确定的映射规则，使高层次上的概念通过低层次上的概念得以实现，直到所要表达的所有事物都表示为0和1组成的数字串为止。

1. 现实世界层

为了处理现实世界中复杂纷纭的万千事物，我们首先必须按一定的规则对这些事物进行分类，而分类的依据是事物本身所具有的特征。因此，现实世界中的一切事物都可以看成是某一些特征的集合体，定义一组特征就可以划分出一类事物，例如，我们可以定义“学校、学号、姓名、性别、班级、出生年月、身高、体重、血型”等特征来刻划“学生”这样一类客观现实世界中的事物对象。而特征的每一组具体的取值就可以区别一个个别的具体事物，如“北京大学、92001、张三、男、74年10月、计92班、1米70、58公斤、A型”这样一组特征值就可以唯一地标定一个学生。

2. 信息层

尽管一个事物几乎可以具有无穷个特征，但我们在处理一个具体问题时，并不需要去收集它的全部特征，而是经过分类、选择、命名等加工，确定一个只含有事物的一小部分最重要、对处理问题来说又是必不可少的那些特征的信息属性的集合，用它来表达一个信息实体。这样，通过对事物的简化，现实世界里的事物就被抽象为信息世界里的数据对象，这就是前面所说的“数据是客观事物的属性的记录”的意思。例如，在大部分学校的管理系统里，一般都选择学号、姓名、性别、班级、出生年月这样一些属性来表示学生数据，而舍弃如身高、体重、血型等特征；但对体育院校来讲，这些特征就不能被舍弃，也必须放进属性的集合里去。

数据有一定的表现形式和载体，常见的表现形式是自然语言符号、数字串和文字串。

3. 数据的逻辑结构层次

计算机处理问题时涉及的数据对象成千上万，相互间的关系错综复杂，这些数据可以杂乱无章地堆砌在机器的内部，换句话说，我们必须考虑数据的结构表示方法。

数据结构理论的基本点在于首先确定数据集合中的基本结构单位（称为“数据结点”），由结点相互之间的联系模式来体现整个数据集合的构造。数据结点可以看成化学中的分子，一方面它是数据结构的组成部分，另一方面它又有自己的内部的构造（“原子结构”）。

已确定了某种基本结构的结点只能改变自己的数据值，而无权改变其内部构造和取值的值域集。对于由结点构成的结构，在问题的处理过程中可以增删结点，这样，整个结构的规模可以动态化，但结点间的联系模式是不变的。

动态数据结构可按结点间的联系模式分成三类，即线性表、树、图，它们是数据集合内部数据关系的典型抽象。例如，数据对象是客人通信录，每个客人的数据是一个结点，其内部包括姓名、地址、电话号码三个数据属性。考虑到通信录中各客人的信息之间只存在先后排列的关系，因此，抽象的数据结点之间亦只存在一种线性互连的结构联系，即一个结点最多有一个直接与之相连的前驱结点和一个后继结点。当然我们可以随时增加一个新客人的数据或是删除一个过时的客人的数据，也可以改变一个客人的地址和电话号码，但结点间的线性联系结构保持不变。我们把这种数据结构称之为线性表。显然，线性表不足以表示所有的数据集合，如一个家谱，从能追溯的某个老祖宗开始，子生孙，孙生子，家族成员之间不构成线性的单传联系，线性表对这样的数据集合便无能为力了，而称之为“树”的数据结构就可以表达这种类型的数据集合。

4. 程序设计语言层次

需要用计算机处理的任何一个问题都必须转化成某一种程序设计语言的表达形式，这样，机器才能识别、理解并执行。数据结构及其操作也是如此，所以，数据连同对数据的加工处理都必须能在程序设计语言中得到表达。这里，语言的数据抽象起到了数据结构与机器表示层次的中介作用，首先寻求数据结构在语言环境中的表达实现，下一步才可以将语言中的数据表达向机器表示转化。

由于数据结构的动态特征，在程序设计语言中一般没有直接定义数据结构的手段。人们只能利用语言所规定特有的数据表达机制来间接地实现所需的数据结构及其操作。在称为高级语言的一类程序设计语言中，数据的基本表达手段是常量和变量，其名字是机器存储空间的抽象，其值是存储在特定空间上的形式为二进制数字串的数据值的抽象。为了能够表达不同类型的数据对象，高级语言规定了“数据类型”这个核心概念。数据类型刻划的是数据的静态结构特征，具有某种数据类型的数据占用的存储空间的大小是固定的，数据的构造方式以及取值的范围也是不变的。例如，“一个实型变量 A 的值为 1.23”表达了一个实数，“A”是存储这个实数的存储单元地址的抽象；1.23 是该单元内存放的二进制数字串的抽象；而实型表达了数据的一种给定的构造方式，因此 A 的所有可能的取值来自一个固定的数值的集合，这个集合是实数集的一个子集。

无论哪个层次上的数据对象的表达形式，归根结底都要被转换成机器层次上二进制的编码表示；一个实型数常会转化为浮点数形式，用两个定点数分别表示尾数和阶码；一个字符用若干位二进制整数代码表示，如一个汉字用规定的相应数字代码表示等等（这里还可分成一