

—— 计算机专业规划教材 ——

计算机软件技术基础 及实验指导

席晓慧 袁玲 王永玲 编著



赠电子课件

 **机械工业出版社**
CHINA MACHINE PRESS

计算机专业规划教材

计算机软件 技术基础及实验指导

席晓慧 袁 玲 王永玲 编著



机械工业出版社

本书共分3篇,第1篇介绍了计算机软件技术基础,包括4章内容:计算机软件工程概述、算法、程序设计语言、数据结构;第2篇介绍了计算机软件开发基础,包括3章内容:数据库技术概述、关系型数据库系统理论、数据库系统设计;第3篇为实验指导,共有14个实验。

本书力求使软件技术的内容浑然一体,而避免成为计算机软件技术基本内容的罗列。本书突出“讲”字,把软件开发过程中的原理和方法用通俗易懂的语言详细描述出来。书中提到的算法,在实验指导中都作了详细的讲解,给出了完整的实现算法的程序。本书涵盖了计算机等级考试中的相关内容,每章配有标准化习题,全书最后还给出了部分习题参考答案。为方便教师教学和学生自学,本书配套电子课件,读者可到 www.cmpedu.com 免费注册审核后登录下载,也可联系责任编辑(010-88379934)索取。本书可以作为各大院校教材也可作为读者自学参考。

图书在版编目(CIP)数据

计算机软件技术基础及实验指导/席晓慧,袁玲,王永玲编著. —北京:机械工业出版社, 2009.2

ISBN 978-7-111-26238-1

I. 计... II. ①席... ②袁... ③王... III. 软件—高等学校—教学参考资料
IV. TP31

中国版本图书馆CIP数据核字(2009)第015781号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

策划编辑:孔熹峻 蔡岩 责任编辑:蔡岩

封面设计:鞠杨 责任印制:杨曦

北京市朝阳展望印刷厂印刷

2009年2月第1版第1次印刷

184mm×260mm·18.75印张·452千字

0 001—3 000册

标准书号:ISBN 978-7-111-26238-1

定价:30.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

销售服务热线电话:(010) 68326294

购书热线电话:(010) 88379639 88379641 88379643

本社服务邮箱:marketing@mail.machineinfo.gov.cn

投稿热线电话:(010) 88379194

编辑热线电话:(010) 88379934

投稿邮箱:Kongxijun@163.com

封面无防伪标均为盗版

前言

和以往同类教材比较，本书具有以下特点：

- 1) 尝试规范了计算机软件技术领域的专业术语。
- 2) 对于图书当中复杂的算法，调试出了能在计算机上直接运行的 C 语言程序。
- 3) 对于重要的基本概念作了深入的讲解，详细描述了概念和概念的出处。
- 4) 对于每个算法，详细描述了算法的基本思路和实现算法的策略。
- 5) 力求使软件技术的内容浑然一体，避免成为计算机软件技术基本内容的罗列。

全书贯穿软件设计的基本过程，即“数据设计（数据结构）+算法设计+语言描述=程序”这样的理念。第一部分软件技术基础，讲解软件技术的基本知识，包括软件开发的方法、规则、基本步骤（软件工程），算法，程序设计语言，数据结构；第二部分软件开发基础，以数据库系统的开发为模板，讲解了软件开发的基本知识，包括数据库系统开发所需要的软件支持，数据设计原理、方法，应用程序设计的理论支持等；第三部分实验，主要是针对数据结构中的算法而设置的，由于数据结构中的问题理解起来有一定的难度，因此读者可以通过调试实验中的程序、分析程序运行的结果，对相关的内容进行深刻理解。

本书第 1 篇中的第 1 章和第 3 章由袁玲编著，第 2 章由王永玲编著，第 4 章由王永玲、席晓慧编著；第 2 篇和第 3 篇由席晓慧编著。全书由席晓慧策划、统稿。

在此，感谢我们的领导和同事在本书编著过程中给予的鼓励和帮助！也感谢我们的学生，正是他们渴望、求知的目光增加了我们完成书稿的决心和动力！

尽管作者倾其所能，并参考了大量的书籍和资料，尽心尽力地完成本书的编著，然而疏漏和不足之处在所难免，希望读者和同行提出宝贵意见。

编 著 者

目 录

前言

第1篇 软件基础

第1章 软件工程..... 2	1.7.1 面向对象的基本概念..... 40
1.1 软件工程概述..... 2	1.7.2 面向对象软件的开发过程..... 42
1.1.1 软件工程的定义..... 2	1.7.3 面向对象的设计..... 50
1.1.2 软件生存周期..... 4	习题..... 54
1.1.3 软件开发模型..... 4	第2章 算法..... 61
1.2 软件的分析..... 6	2.1 算法的概念..... 61
1.2.1 问题定义..... 6	2.1.1 算法的基本概念..... 61
1.2.2 可行性研究..... 6	2.1.2 算法的特性..... 62
1.2.3 需求分析..... 6	2.2 算法的描述..... 62
1.2.4 软件分析方法..... 7	2.3 算法的评估..... 63
1.2.5 表达工具..... 8	2.3.1 算法设计的要求..... 63
1.3 软件的设计..... 12	2.3.2 算法效率的度量..... 64
1.3.1 软件设计概述..... 13	习题..... 67
1.3.2 软件设计准则..... 14	第3章 程序设计语言..... 68
1.3.3 总体设计方法..... 16	3.1 程序设计语言的分类..... 68
1.3.4 详细设计方法..... 20	3.1.1 低级程序设计语言..... 68
1.4 软件编程..... 24	3.1.2 高级程序设计语言..... 70
1.4.1 编程语言的选择..... 25	3.2 高级程序设计语言介绍..... 70
1.4.2 编程风格..... 25	3.2.1 面向过程的程序设计语言..... 70
1.5 软件的测试..... 26	3.2.2 面向对象程序设计语言的特点..... 79
1.5.1 软件测试概述..... 26	习题..... 82
1.5.2 测试用例的设计..... 27	第4章 数据结构..... 83
1.5.3 测试实施策略..... 34	4.1 数据结构概述..... 83
1.5.4 软件的调试..... 37	4.1.1 数据结构的定义..... 83
1.6 软件的维护..... 39	4.1.2 数据结构的基本内容..... 84
1.6.1 维护的基本概念..... 39	4.2 线性表..... 85
1.6.2 软件的可维护性..... 39	4.2.1 线性表的逻辑结构..... 85
1.6.3 维护的实施..... 39	4.2.2 线性表的存储结构..... 86
1.7 面向对象的方法..... 40	

4.2.3 算法评价及改进算法的 各种策略	93	4.5 图	135
4.3 特殊线性表	102	4.5.1 图的定义及存储结构	135
4.3.1 栈	102	4.5.2 图的存储	137
4.3.2 队列	108	4.5.3 图的常用算法—— 图的遍历	139
4.3.3 串	113	4.6 查找	142
4.3.4 数组	116	4.6.1 静态查找	143
4.4 树	121	4.6.2 动态查找	147
4.4.1 树的定义及存储结构	121	4.6.3 哈希查找	148
4.4.2 二叉树的定义及存储	123	4.7 排序	151
4.4.3 二叉树的存储结构	125	4.7.1 选择排序	152
4.4.4 树、森林与二叉树 之间的转换	127	4.7.2 插入排序	152
4.4.5 二叉树的算法—— 遍历二叉树	129	4.7.3 冒泡排序	153
4.4.6 二叉树的应用	131	4.7.4 快速排序	154
		4.7.5 归并排序	156
		习题	157

第2篇 计算机软件开发基础

第5章 数据库技术概述	170	6.2 关系型数据库标准语言 SQL	198
5.1 数据管理技术的发展	170	6.2.1 SQL 概述	198
5.1.1 人工管理阶段	170	6.2.2 数据的定义	201
5.1.2 文件系统阶段	171	6.2.3 数据操纵	204
5.1.3 数据库系统阶段	171	6.2.4 数据控制	212
5.1.4 数据的传统管理方式与 数据库管理方式的比较	172	6.2.5 嵌入式 SQL	214
5.2 数据及数据模型	174	6.3 关系数据库的规范化理论	216
5.2.1 数据描述的三个领域	174	6.3.1 函数依赖	219
5.2.2 数据模型	176	6.3.2 规范化的关系模式	221
5.2.3 数据库系统的体系结构	180	6.3.3 关系模式的分解	223
5.3 数据库系统的组成	182	习题	226
5.3.1 软件部分	183	第7章 数据库系统设计	230
5.3.2 硬件部分	184	7.1 概述	230
5.3.3 数据库管理员	184	7.1.1 数据库系统设计的任务	231
习题	184	7.1.2 数据库系统设计的特点	232
第6章 关系型数据库理论	188	7.1.3 数据库系统设计的步骤	232
6.1 关系及关系代数	188	7.2 需求分析	233
6.1.1 关系及关系模型	188	7.2.1 需求信息的收集	233
6.1.2 关系代数	191	7.2.2 需求信息的整理	234
		7.3 概念设计	235

7.3.1 设计局部概念模式.....	236	7.4.3 物理设计	241
7.3.2 设计全局概念模式.....	237	7.4.4 存储记录的格式设计	241
7.4 逻辑设计	240	7.4.5 存储方法设计	241
7.4.1 E-R 图到关系模式的转换.....	240	7.4.6 存取方法设计	242
7.4.2 数据逻辑设计	240	习题	242

第3篇 实验指导

实验 1 线性表的建立	244	实验 10 二叉树的应用—— 二叉排序树的建立和访问.....	274
实验 2 线性表的插入	248	实验 11 图的邻接表建立及图的 遍历.....	277
实验 3 线性表的删除	252	实验 12 折半查找算法	281
实验 4 顺序栈的建立和操作	255	实验 13 哈希查找算法	283
实验 5 链栈的建立和操作	258	实验 14 快速排序算法	285
实验 6 队列的建立和操作	261	部分习题参考答案.....	288
实验 7 稀疏矩阵压缩存储的转置运算.....	265	参考文献	293
实验 8 二叉树的顺序存储及访问.....	268		
实验 9 二叉树动态存储的构建及遍历.....	271		

第 1 篇

软件基础

- 第 1 章 软件工程
- 第 2 章 算法
- 第 3 章 程序设计语言
- 第 4 章 数据结构

第1章

软件工程

在软件研制开发过程中，若能严格遵循软件工程的方法论，则可提高软件开发的成功率，减少软件开发和维护中出现的问题。软件工程学的目标是以解决软件生产的质量与效率问题为宗旨，指导软件生产全过程的所有活动。本章以软件生命周期为基础，较系统地介绍了软件工程学的内容，主要包括：

- ① 软件的分析
- ② 软件的设计
- ③ 软件编程
- ④ 软件的测试技术
- ⑤ 软件的维护

1.1 软件工程概述

1.1.1 软件工程的定义

计算机系统由硬件系统和软件系统组成。计算机软件包括程序、相关数据及其说明文档。其中程序是指完成预定功能和性能的可执行指令序列，数据是程序能正常操作信息的数据结构，文档是与程序开发维护和使用相关的各种图文资料。

随着计算机硬件的发展，计算机软件也从规模、功能等方面得到很大发展，软件的发展大致经历了程序设计时代、软件时代和软件工程时代。

1. 程序设计时代

这个时代的硬件处于电子管时代。当时注重的是硬件的性能和指标，程序的编写处于从属地位。程序设计的工具是机器语言、汇编语言，其方法追求编程技巧，追求效率高、内存省。人们仅根据需要来编制一些可以直接运行的程序，而不考虑系统地开发软件。这个时期计算机的应用主要限于科学计算，程序设计方法尚能满足计算机应用的要求。

2. 软件时代

这个时代硬件进入晶体管、小规模集成电路时期。软件工具已使用第二代语言，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，



计算机的应用扩大到数据处理和过程控制领域；各种应用软件开发项目大量提出，软件规模越来越大，结构也越来越复杂。然而在计算机科学领域中，程序设计方法和软件开发技术没有重大突破，这样，在软件产品开发的复杂需求与软件开发技术的能力之间就产生了尖锐的矛盾，当时的软件开发技术无法解决各种规模庞大而结构又非常复杂的软件系统的开发问题，在许多大型软件项目开发过程中，都遇到了一系列严重问题：费用一再增加，周期一再延长，软件中的许多错误无法排除，软件可靠性很差，许多系统失败了，导致了所谓的“软件危机”。

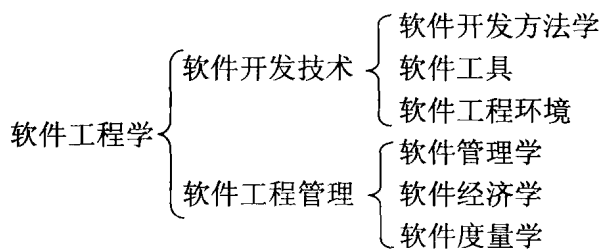
1968年，北大西洋公约组织（NATO）在前联邦德国召开的一次会议上，讨论软件危机问题时正式提出并使用了“软件工程”一词。它的中心思想是把软件当作一种工业产品，从管理和技术两个方面对计算机软件进行开发和维护。从此软件生产开始进入软件工程时代。

3. 软件工程时代

随着大规模和超大规模集成电路的出现，使得计算机系统的复杂程度大大提高，软件开发工具已使用第三代语言 C、PASCAL，数据库已使用，第四代语言面向对象的语言也开始投入使用，计算机应用领域也大大扩展。这个时期，软件工程的各種概念、新方法、新思想不断出现。先后提出了自顶向下、逐步求精、模块化的结构程序设计方法；提出了软件生存周期模型及若干新的软件测试技术、测试原理和若干处理需求定义方面的技术。20世纪80年代后，人们开始把软件工程各阶段的工具集成到一起，形成软件开发的支撑环境，更有效地支持软件开发的工程化，使软件产品的质量和可靠性都得到了提高。

软件工程是指导计算机软件开发和维护的一门工程学科。它采用工程的概念、原理、技术和方法来开发与维护软件，把先进的管理技术和先进的计算机科学技术方法结合起来，经济地开发出高质量的软件并有效地维护它。

软件工程是一门交叉学科，涉及到计算机科学、管理科学、工程学和数学。软件工程的理论、方法、技术都是建立在计算机科学的基础上，它是用管理学的原理、方法来进行软件生产管理；用工程学的观点来进行费用估算、制定进度和方案；用数学方法来建立软件可靠性模型以及分析各种算法和性质。



软件工程研究的对象是大型软件系统的开发过程，它研究的内容是生产流程，各生产步骤的目的、任务、方法、技术、工具、文档和产品规格。

国际上制定了若干个关于软件工程的技术标准，自1988年起，我国也制定了相应的国家标准，在之后的使用过程中，经过修订后，现行的标准如 GB/T8566—2007《信息技术 软件生存周期过程》、GB/T8567—2006《计算机软件文档编制规范》以及 GB/T1526—1989

《信息处理 数据流程图、程序流程图、系统流程图、程序网络图和系统资源图的文件编制符号及约定》等。这些标准的制定及修订对软件工程的发展、软件生产和文档的编制都起到了很大作用。

1.1.2 软件生存周期

软件和其他产品一样，周期性地重复着一些阶段。从用户开发要求的提出，直到该软件淘汰为止的这一全过程为软件生命周期。通常，软件生命周期包括三个时期：软件定义期、软件开发期、软件维护期。为使各时期的任务更明确，又可以分为以下几个阶段：问题定义、可行性研究、需求分析、系统设计、详细设计、编码、测试和运行维护。

(1) 软件定义期 包括问题定义、可行性研究和需求分析三个阶段。

(2) 软件开发期 包括系统设计、详细设计、编码和测试四个阶段。

(3) 软件维护期 即运行维护阶段。

软件生存周期的五个阶段构成了软件开发过程的模型，如图 1-1 所示，模型中每个阶段的任务具有相对独立性，这样便于开发人员的分工协作，从而降低整个软件开发工作的困难程度，若流程中某个阶段发生错误，还要回到前面有关阶段去运行修改。

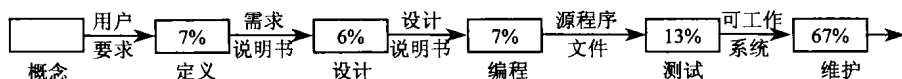


图 1-1 软件生存周期模型

1.1.3 软件开发模型

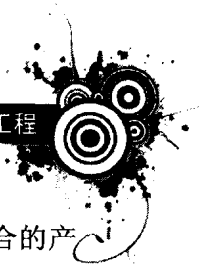
开发一个高质量的软件产品，首先要根据软件生存周期为各项开发活动的流程确定一个合理的框架，称为软件生存周期模型或软件开发模型。生命周期模型规定了把生命周期划分成哪些阶段以及各阶段执行的顺序，因此，也称为过程模型。目前有许多种软件生命周期模型用于指导软件生产过程，如瀑布模型、快速原型模型、增量模型、螺旋模型和喷泉模型等。

1. 瀑布模型

瀑布模型将软件生命周期的各项活动规定为依固定顺序连接的若干阶段工作，它包括系统需求、软件需求、可行性分析、软件设计、软件实现、软件测试与软件维护。它规定了由前至后、相互衔接的固定次序，就如瀑布流水，逐级下落，最终得到软件产品。

2. 快速原型模型

快速原型模型是指在进行需求分析时，以较小的代价快速建立起一个能够反映用户需求的原型系统，原型系统仅包括未来系统的主要功能，以及系统的重要接口。让用户在计算机上试用它，通过实践来了解目标系统的概貌，以便判断哪些功能是符合需要的，哪些方面还需要改进，开发人员按照用户的意见快速地修改原型系统，然后再次请用户试用、评价、提出意见，重复这一过程，直到用户满意为止。最终建立完全符合用户要求的新系统。



3. 增量模型

增量模型也称渐增模型，是瀑布模型的顺序特征与快速原型法的迭代特征相结合的产物。使用增量模型开发软件时，把软件产品作为一系列的增量构件来设计、编码、集成和测试。每个构件由多个相互作用的模块构成，并且能完成特定的功能。使用增量模型时，第一个增量构件往往实现软件的基本需求，提供最核心的功能。增量模型中，过程在一系列步骤中开发：首先完成整个软件包的简化版本，第一个简化版本通常只包括主模块及对空白子模块的调用，主模块调用每个子模块，而每个子模块只返回它的信息，说明它被调用；在第二个版本中，更多的子模块将被完成，再次测试软件保证主模块可以正确地调用子模块，如果有问题，开发者找出问题所在，直到软件包正常，最后所有软件包增加完毕。

有两种渐增模型：

1) 增量构造模型：在该模型中，需求分析和设计阶段都是按照瀑布模型的整体方式开发，而在编码和测试阶段则按增量方式开发。这样，用户可以及早地看到部分软件，发现问题，以便在开发其他软件功能时及时地加以考虑。

2) 演化提交模型：在该模型中，先对某部分功能进行需求分析，然后按顺序进行设计、编码与测试，直接把该功能的软件交付用户使用再对另一部分功能按同样的方式进行开发并提交用户，直至所有功能全部增量开发完毕为止。

4. 螺旋模型

软件开发几乎总要冒一定风险。螺旋模型是在结合快速原型及其他方法如瀑布模型的基础上增加了风险分析。螺旋模型的基本做法是在模型的每一个开发阶段之前，引入非常严格的风险识别、风险分析和风险控制。采取了消除风险的措施之后，再开始计划下一阶段的开发工作，否则，项目就很可能被取消。

5. 喷泉模型

喷泉模型以面向对象的软件开发技术为基础，以用户需求为动力。它克服了瀑布模型不支持软件重用和多项开发活动集成的局限性，使得开发过程具有迭代性和无间隙性。其特点如下：

1) 喷泉模型规定软件的开发过程有四个阶段：即系统分析、系统设计、软件设计和编码实现。

2) 喷泉模型各阶段相互重叠，能够充分反映软件开发过程的并行性。

3) 喷泉模型采用增量开发方式，整个过程是一个迭代的、逐步提炼的过程。

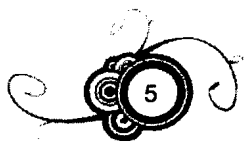
4) 喷泉模型属于对象驱动过程，而对象是所有活动作用的实体，也是项目管理的基本内容。

6. 构件组装模型

面向对象技术是构件组装模型的基础，它为基于构件的过程建模提供了技术框架。面向对象技术强调类的创建，将数据和操纵该数据的算法封装为一个类。将一个或多个相关类的组合称为一个构件，构件也是可重用的软件成分。

构件组装模型融合了螺旋模型的特征，本质上是演化的，并且支持软件开发的迭代方法，它是利用预先包装好的软件构件来构造应用程序的。构造过程如下：

1) 标识候选类。



2) 根据标识的类搜索该类库, 如果该类存在, 就从类库中提取出来复用。

3) 如果该类不存在, 就采用面向对象的方法开发它, 并将新类放入库中。

4) 从库中提取的类和为了满足应用程序的特定要求而建造的新类组织一个完整的软件版本, 提供用户评价, 由此完成待开发应用程序的第一次迭代。重复上述开发过程。

7. 第四代技术模型

第四代技术 (the Fourth-Generation Techniques, 4GT) 包含了一系列的软件工具。它们的共同点是能使软件设计者在较高级别上说明软件的某些特征, 由软件工具根据其说明自动生成源代码。第四代技术模型应用的关键在于软件描述的能力, 它用一种特定的语言或者以一种用户可以理解的问题描述方法来描述需要解决的问题。

目前, 支持第四代技术模型的软件开发环境及工具有: 数据库查询的非过程语言、报告生成器、数据操纵、屏幕交互及定义, 以及代码生成、高级图形功能、电子表格功能等。

要将一个第四代技术模型生成的功能变成最终产品, 开发者还必须进行测试, 写出有意义的文档, 并完成其他软件工程模型中同样要求的所有集成活动。此外, 采用第四代技术开发的软件还必须考虑维护是否能够迅速实现。

1.2 软件的分析

这是整个软件过程的第一个时期, 也是关系到软件开发成败的关键步骤。软件分析进一步可分为三个阶段: 问题定义、可行性研究和需求分析。此阶段结束时要产生如下文档: 可行性报告、软件计划、需求说明书等。

1.2.1 问题定义

问题定义的任务是确定问题的性质、工程目标以及规模。应力求使软件开发人员、用户以及使用部门负责人对问题的性质、工程目标与规模取得完全一致的看法, 这对确保软件开发的成功是非常重要的。通过对用户的访问调查, 分析员应提交书面报告给用户与使用部门负责人进行审查。

1.2.2 可行性研究

可行性研究的任务是用最小的代价确定在问题定义阶段的系统目标和规模是否现实, 所确定的问题是否可以解决, 给出若干个解决问题方案, 论证系统方案在经济上、技术上和操作上是否可行。若可行, 则提出一个推荐方案, 草拟开发计划, 书写文档提交审查。

1.2.3 需求分析

这个阶段的任务是理解和表达用户的要求, 对用户的业务活动进行分析, 明确在用户的业务环境中软件系统应该“做什么”。确定对系统的综合要求, 它包括: 软件系统的范围、功能、性能、限制和约束以及将来可能会提出哪些要求。软件系统的范围是指软件的规模

有多大,处理的对象及性质是什么;软件系统的功能是指能做什么样的加工和处理,如数据录入、查询、统计分析、打印报表等;软件系统的性能是指处理数据量的多少、系统响应时间、查询速度、数据的精度、系统工作可靠性程度等;软件系统的限制和约束是指开发费用、开发周期、可使用的资源等。其中功能要求是基本的,它又包括数据要求和处理要求两个方面。

该阶段结束时,编写需求说明书,以文档的形式把用户要求的功能、性能表达出来,需求规格说明书主要包括以下内容:引言、数据描述(数据流图、数据字典)、功能描述、性能描述、质量保证和其他。

需求说明书主要有三个作用:作为用户和软件开发人员之间的合同;作为开发人员进行设计和编程的根据;作为软件开发完成后验收的依据。

需求分析对整个开发阶段都具有重大的影响,它是软件开发的基础。一旦需求分析出现错误,将导致整个软件开发的失败。如果在需求分析时产生一个错误,这个错误发现越晚,则花费的代价越高。这是需求分析的特点之一。

1.2.4 软件分析方法

软件分析方法可分为结构化分析方法和面向对象的分析方法。下面采用结构化分析方法,分析需求定义。面向对象的分析方法见第3章。

自从20世纪70年代末,E.Yourdon等人提出结构化分析方法以来,该方法得到了广泛的应用。

建立系统模型

结构化分析方法是面向数据流进行需求分析的一种方法。它使用数据流图、数据字典等工具进行分析。结构化分析方法适合于数据处理类型软件的需求分析。其核心思想是分解、化简问题,将物理与逻辑表示分开,对系统进行数据与逻辑的抽象。具体地讲,就是用抽象模型的概念,按照软件内部数据传递、变换的关系,自顶向下逐层分解,直到找到满足功能要求的所有可实现的软件为止。

使用结构化分析方法进行软件需求分析时,可按如下的步骤进行:

- 1) 建立当前系统的物理模型,通过对现实环境的调查研究,获得当前系统的物理模型。通常用数据流图把当前系统的工作流程描述出来。
- 2) 建立当前系统的逻辑模型。去掉具体模型中的非本质因素,抽象出当前系统的功能和性能,建立当前系统的逻辑模型。
- 3) 建立目标系统的逻辑模型。分析当前系统与目标系统的差别,建立目标系统的逻辑模型。目标系统是一个使用计算机的系统。它的功能应该比当前的现行系统更强。
- 4) 对目标系统进行完善和补充、并写出完整的需求说明。
 - ① 确定目标系统中哪些功能交给计算机去做,哪些功能由人工完成。
 - ② 补充迄今尚未考虑过的细节,例如确定系统的响应时间,增加出错处理等。

例 1-1: 建立学生选课的模型。

建立当前系统的物理模型:学生选课的手续是先找所在院(系)办公室的张秘书开证明,凭证明找教务科李秘书选课,然后凭选课单找赵秘书注册。图1-2是这一过程的具体模型。

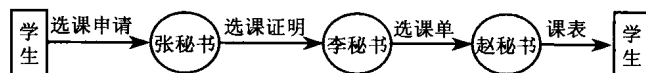


图 1-2 学生选课的具体模型

建立当前系统的逻辑模型：在图 1-2 中，张、李等具体的人是可变动的，但需要他们处理的工作，例如审查选课有效性、开选课单则是不变的，后者才是本质的内容。经过这样的分析，就可抽象出学生选课这一系统的逻辑模型，如图 1-3 所示。



图 1-3 学生选课的逻辑模型

建立目标系统的逻辑模型：在计算机系统中，可省去开有效申请单的手续，可由学生登录选课系统输入学号、密码来完成，如图 1-4 所示。

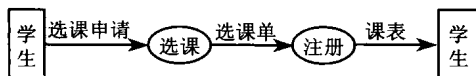


图 1-4 计算机选课系统的逻辑模型

完善和补充目标系统模型：实际上在选课之前，至少要查阅两个文件：选课清单，用以核对该学生所选的课程是否是本学期开设；选课注册表，核对该生所选的课是否已经人满。把这两个文件加进图中。在本例中，假如选课单中出现了本学期没有开设的课或已经选满的课，可用“无效选课单”将相应的情况通知学生。

在图 1-4 的系统中，选课和注册由计算机完成，可合并进行。

经过以上的修正和补充，就可以得到改进后的目标系统逻辑模型，如图 1-5 所示。

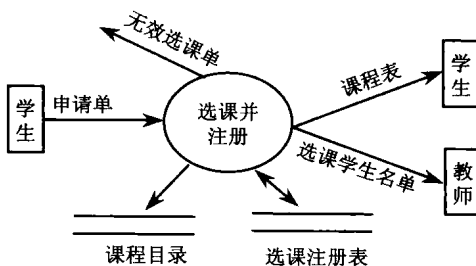


图 1-5 计算机选课系统的数据流程图

1.2.5 表达工具

结构化分析方法在表达问题时尽可能用图形的方法，数据流图和数据字典是常用的描述工具。数据流图表示了软件的信息流向和信息加工，但在数据流图中还有许多事情无法说明，需要数据字典对信息和加工进行更详细的描述。还可使用结构英语、判定表、判定树对信息加工的加工逻辑进行描述。

1. 数据流图

数据流图（DFD）是软件系统逻辑模型的一种图形表示，图 1-5 是一个描述选课系统的数据流图。图中包括两个加工（选课、注册），2 个文件（课程目录、选课注册表），文件与加工之间用双向箭头联接时，表示对文件中的数据有读有写。

（1）数据流图的基本成分 数据流图由四种基本成分组成：数据流、数据存储、加工、源点和终点。四种基本成分的图形如图 1-6 所示。

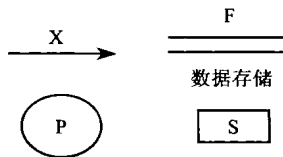


图 1-6 数据流图基本成分

1) 数据流。数据流用箭头表示，箭头旁边用文字加以标记。数据流相当于一组管道，并有一组数据（信息）流经它。数据流可从源点流向加工或从一个加工流向终点，可以有几股数据流流出或流入一个加工。图 1-5 中包括四个数据流：申请单、课程表、选课学生名单、无效选课单。

2) 数据存储。数据流图中的数据存储用两根平行线表示，双线旁边用文字加以标记。在计算机中常用文件来表示数据存储，文件名写在两条平行线旁边。如果某加工需要文件，则数据流方向是从文件流向该加工；如果某加工输出的数据要存入文件或修改文件，则数据流是从该加工流向文件。通常，进出文件的数据流的名字可以省略，例如，在图 1-5 中，包括两个文件“课程目录”和“选课注册表”。

3) 数据处理。数据处理又称加工。在数据流图中，加工用标有名字（处理名）的圆圈表示。其中处理名就是对数据进行操作的名称。指向加工的数据流表示该加工的输入数据，离开加工的数据流表示该加工的输出数据。例如，图 1-5 中的“选课并注册”就是加工。

4) 源点和终点。源点和终点用矩形框表示，矩形框中用文字加以标记。它表示系统中数据的来龙去脉，通常存于系统之外的人和组织中。

说明：

① 数据流图中的箭头仅能表示在系统中流动的数据。

② 数据流图与程序流程图不同，它不能表示程序的控制结构，例如选择或循环结构等。数据流图用作软件分析阶段的工具，而程序流程图用于表示程序的设计过程。

③ 数据流图表现的范围有很大的灵活性。在需求分析中常常用一组数据流图由粗到精地表示同一软件在不同抽象级别上的逻辑模型，并称其为分层数据流图。

（2）数据流图的结构 大型复杂的软件系统，其数据流图可能含有数百乃至数千个加工，不能设想一次就将它们全部画齐。软件工程中控制复杂性的基本手段有两个，就是分解和抽象。为了描述信息流向和加工，可用一套分层的数据流图来描述，包括顶层、中间层、底层。

1) 顶层。决定系统的范围，决定输入输出的数据流，它说明系统的边界，把整个系统的功能抽象为一个加工，顶层数据流图只有一张。

2) 中间层。顶层之下是若干中间层，某一中间层既是它上一层的加工的分解结果，又是它下一层若干加工的抽象，即它又可进一步分解。

3) 底层。底层数据流图的加工是由基本加工构成的，不能再进行分解。

正确的做法是：从系统的基本模型（把整个系统看成一个加工，如图 1-7 所示）逐层地对系统进行分解。每分解一次，系统的加工数量就增多一些，每个加工的功能也更具体一些。如图 1-8 所示，为了理解系统 A，把它分解为子系统 1，子系统 2，子系统 3...，若

子系统还很复杂，在此基础上再把它细分，继续重复这种分解，直到各部分都足够简单，不必再分解为止。通常把这种不需要再分解的加工称为“基本加工”，把上述分解方法称为“自顶向下、逐步细化”。通过这种分解，所分析的系统将获得一组分层的数据流图，以替代含有系统全部加工的包罗万象的总数据流图。

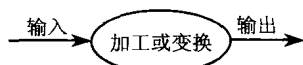


图 1-7 软件模型

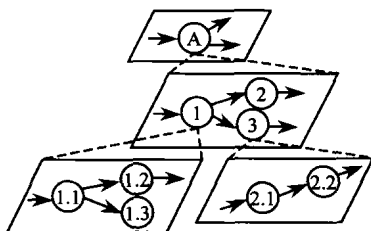


图 1-8 分解与抽象

(3) 分层数据流图的画法 画分层数据流图时，应根据分解与抽象的原则自顶向下逐层分解画出，在画各层数据流图时，要注意父图和子图的平衡、各层流图及其加工的编号和数据守恒问题。

1) 父图和子图的平衡。在分层数据流图直接相邻的两层中，上层是下层的父图，下层是上层的子图。一般地，父图中有几个加工，就对应下层有几个子图，但子图的个数也可少于父图中的加工个数，即父图中有些加工可能是基本加工，它就没有子图。

2) 子图的编号规则。子图的编号即为父图相应加工的编号：子图中加工的编号由子图号、小数点、局部号构成。顶层只有一张，只有一个加工，不必编号。第一层编号为 0，图中加工的编号为 0.1, 0.2, ..., 通常简化为 1, 2, ...。对应的子图编号为 1 图, 2 图, ..., 这是第二层数据流图编号，该层图中加工的编号为 1.1, 1.2, ..., 2.1, 2.2, ...。这样可以根据子图编号中小数点的个数来确定该子图在哪一层上。

3) 数据守恒。所谓数据守恒是指加工的输入输出数据流是否匹配，即一个加工既有输入数据流又有输出数据流。若有输入无输出，或有输出无输入，则数据都是不守恒的。

4) 完善数据流图。画出了分层数据流图后，应进一步完善，提高数据流图的可理解性。首先要简化加工关系，使得各部分尽可能相对独立，在对加工进行分解时，应注意分解的均匀性，经验证明一个加工一次分解不超过七个子加工为宜。分解过多，可能有较多的层次；分解过少，使人难以理解。一般分解应是自然的，概念上是合理的、清晰的。若一张子图上的所有加工都为基本加工，这时分解过程就可以结束了。

2. 数据字典

数据字典是结构化分析方法的另一个重要工具。数据流图中描述的是软件系统的信息流程和加工，没有对数据流、数据存储、加工进行详细说明，而结构化分析方法使用数据字典对这些进行了详细说明。数据流图中的数据流名、数据存储名、数据项名、基本加工名的严格定义的集合构成了数据字典。数据字典主要是给数据流图中的每一个数据流名、文件名以及处理名建立一个条目，在这些条目中给出各名字的定义。在每一个条目下又可以建立子条目，直到每一个组成部分不能再划分为止。数据字典与流程图配套，共同构成系统的逻辑模型。数据字典中有如下四种条目：数据流、数据存储、数据项和基本加工条目。