

全国计算机等级考试教材系列

National Computer Rank Examination

全国计算机等级考试

二级 公共基础知识 与C语言真题汇编

周丽娟 主 编
冯 雪 段雪莹 副主编
邢满堂 主 审

Computer Rank
National Examination



中国水利水电出版社
www.waterpub.com.cn

最新大纲

全国计算机等级考试教材系列

全国计算机等级考试—— 二级公共基础知识与 C 语言真题汇编

周丽娟 主 编

冯 雪 段雪莹 副主编

邢满堂 主 审

中国水利水电出版社

内 容 提 要

本书按照全国计算机等级考试最新大纲的要求,根据等级考试出题的顺序,将大纲要求的内容贯穿其中,帮助考生更好地理解和掌握等级考试的内容、范围和难度,同时也便于考生把握命题规律,快速提升应试能力。

本书内容分为三部分:公共基础知识,C程序设计各章练习题,全国计算机等级考试模拟试题及真题汇编。本书题型丰富,知识点覆盖全面,重点突出,针对性强,很好地应用本书对通过等级考试会起到事半功倍的效果。

本书适合作为全国计算机等级考试考生的教材,也可作为广大计算机初学者及大中专院校师生的参考书。

图书在版编目(CIP)数据

全国计算机等级考试二级公共基础知识与 C 语言真题汇编 / 周丽娟主编. —北京:中国水利水电出版社, 2006

(全国计算机等级考试教材系列)

ISBN 7-5084-3643-1

I. 全… II. 周… III. C 语言—程序设计—水平考试—习题
IV.TP312-44

中国版本图书馆 CIP 数据核字 (2006) 第 021646 号

书 名	全国计算机等级考试——二级公共基础知识与 C 语言真题汇编
作 者	周丽娟 主 编 冯 雪 段雪莹 副主编 邢满堂 主 审
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 63202266 (总机) 68331835 (营销中心) 82562819 (万水) 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京北医印刷厂
规 格	787mm×1092mm 16 开本 20.25 印张 498 千字
版 次	2006 年 3 月第 1 版 2006 年 3 月第 1 次印刷
印 数	0001—3000 册
定 价	30.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前 言

当今社会是信息时代，计算机与软件技术日新月异，发展迅速。计算机作为一种工具已渗透到人们日常生活的各个领域，这样，对计算机知识的掌握程度和应用能力已经成为衡量一个新型人才的重要环节。

自国家教育部考试中心于1994年推出“全国计算机等级考试”(National Computer Rank Examination, 简称NCRE)以来，每年都有大量考生参加考试，其中一些人获得不同级别的证书，成为用人单位选聘人才的一项重要指标。

本书是根据教育部考试中心最新公布的《全国计算机等级考试大纲(二级C语言程序设计)》编写的。为适合考生应考，本书紧扣考试大纲，全面囊括大纲中所有的知识点，同时针对教育部考试中心全国计算机等级考试二级C语言笔试试题进行全面分析，并在此基础上甄选大量典型练习题。通过对本书的学习，使考生能够抓住考试的重点与难点，了解考试中经常出现的题型，并掌握每种题型的解法，同时也使考生把握等级考试的出题思路、命题规律，从而提高应试能力，提高通过率。

本书内容分为三部分：公共基础知识，C程序设计各章练习题，全国计算机等级考试模拟试题及真题汇编。本书题型丰富，知识点覆盖全面，重点突出，针对性强，很好地应用本书对通过等级考试会起到事半功倍的效果。本书不仅适合全国计算机等级考试(二级C语言程序设计)的考生使用，而且适合学习C语言程序设计的读者参考。

本书作者是多年从事计算机基础教学的老师，多年参与等级考试的辅导，有着丰富的教学经验。本书由周丽娟任主编，冯雪、段雪莹任副主编，全书由邢满堂主审，参加本书编写工作的还有吴琼、侯冬薇、刘斌文、于雪晶、郎建军、沈喜权、张守伟等，在此对他们的辛勤劳动表示衷心的感谢。

由于本书出版时间比较仓促，谬误之处在所难免，恳请广大读者能及时给予批评和指正，不胜感激！

编者

2006年1月

目 录

前言

第一部分 公共基础知识

第1章 软件工程基础.....	1
1.1 软件工程基本概念	1
1.1.1 软件定义与软件特点	1
1.1.2 软件危机与软件工程	2
1.1.3 软件工程过程与软件生命周期.....	4
1.1.4 软件工程的目标与原则	5
1.1.5 软件开发工具与软件开发环境.....	6
1.2 结构化分析方法	7
1.2.1 需求分析与需求分析方法.....	7
1.2.2 结构化分析方法	8
1.2.3 软件需求规格说明书	12
1.3 结构化设计方法	14
1.3.1 软件设计的基本概念	14
1.3.2 概要设计	16
1.3.3 详细设计	21
1.4 软件测试	26
1.4.1 软件测试的目的	26
1.4.2 软件测试的准则	26
1.4.3 软件测试技术与方法综述.....	27
1.4.4 软件测试的实施	34
1.5 程序的调试	37
1.5.1 基本概念	37
1.5.2 软件调试方法	38
习题	40
第2章 程序设计基础.....	41
2.1 程序设计方法与风格	41
2.2 结构化程序设计	42
2.2.1 结构化程序设计的原则	42
2.2.2 结构化程序的基本结构与特点.....	43
2.2.3 结构化程序原则和方法的应用.....	44

2.3 面向对象的程序设计	44
2.3.1 关于面向对象方法	44
2.3.2 面向对象方法的基本概念	47
习题二	50
第3章 数据结构与算法	52
3.1 算法	52
3.1.1 算法的基本概念	52
3.1.2 算法复杂度	55
3.2 数据结构的基本概念	56
3.2.1 什么是数据结构	57
3.2.2 数据结构的图形表示	59
3.2.3 线性结构与非线性结构	60
3.3 线性表及其顺序存储结构	60
3.3.1 线性表的基本概念	60
3.3.2 线性表的顺序存储结构	61
3.4 栈和队列	62
3.4.1 栈及其基本运算	62
3.4.2 队列及其基本运算	63
3.4.3 线性链表的基本概念	66
3.5 树与二叉树	68
3.5.1 树的基本概念	68
3.5.2 二叉树及其基本性质	71
3.5.3 二叉树的存储结构	74
3.5.4 二叉树的遍历	74
3.6 查找技术	75
3.6.1 顺序查找	75
3.6.2 二分法查找	76
3.7 排序技术	75
3.7.1 交换类排序法	76
3.7.2 插入类排序法	78
3.7.3 选择类排序法	80
习题三	81
第4章 数据库设计基础	84
4.1 数据库系统的基本概念	84
4.1.1 数据、数据库和数据库管理系统	84
4.1.2 数据库系统的发展	88
4.1.3 数据库系统的基本特点	89

4.1.4 数据库的内部结构	90
4.2 数据模型	92
4.2.1 数据模型的基本概念	92
4.2.2 E-R 模型	93
4.2.3 层次模型	96
4.2.4 网状模型	97
4.2.5 关系模型	98
4.3 关系代数	101
4.4 数据库设计与管理	106
4.4.1 数据库设计概述	107
4.4.2 数据库设计的需求分析	107
4.4.3 数据库概念设计	108
4.4.4 数据库的逻辑设计	111
4.4.5 数据库的物理设计	113
4.4.6 数据库管理	113
习题四	114

第二部分 C 语言各章知识点及练习

第 5 章 C 语言基础知识	117
第 6 章 C 语言控制结构	129
第 7 章 指针与数组	143
第 8 章 函数与预处理	162
第 9 章 结构体与共用体	178
第 10 章 文件操作	182

第三部分 模拟试题及真题汇编

模拟试题一	187
模拟试题二	197
模拟试题三	207
模拟试题四	218
模拟试题五	231
模拟试题六	245
模拟试题七	258
模拟试题八	272
2005 年 4 月二级 C 笔试试题	284
2005 年 9 月二级 C 笔试试题	299
附录 二级 C 语言程序设计考试大纲	313

第一部分 公共基础知识

随着社会的发展,计算机在我们的生活中越来越重要,已渐渐密不可分。为了更好地胜任将来的专业技术工作,我们不但要具备使用计算机及网络处理日常事物的基本能力,更要具备应用现代计算机环境,设计、开发计算机应用系统解决专业问题的能力。为此,我们有必要进一步掌握数据分析与信息处理以及信息系统开发的基本方法。

例如,开发一个简单的学生信息管理系统,对高校学生的基本信息、班级信息、课程信息、成绩信息进行管理。通过该系统,可以做到信息的规范管理、科学统计和快速的查询,从而减少管理方面的工作量。开发这样一个系统,就像盖一座大楼一样是一项工程,这就是软件工程研究的内容。上万学生的各种信息要存储在计算机中,现在普遍采用的技术是数据库技术。存储在计算机中的学生信息不是杂乱无章的,是按照一定的原则来组织和存储的,而且要便于使用,这是算法和数据结构研究的内容。而计算机实现的各种功能是通过执行程序来完成的,相应的程序设计方法我们也要作初步的了解。

这部分的主要内容:软件工程基础、程序设计基础、算法与数据结构、数据库基础。

第1章 软件工程基础

1.1 软件工程基本概念

1.1.1 软件定义与软件特点

计算机软件(Software)是计算机系统中与硬件相互依存的另一部分,是包括程序、数据及相关文档的完整集合。其中,程序是软件开发人员根据用户需求开发的、用程序设计语言描述的、适合计算机执行的指令(语句)序列。数据是使用程序能正常操纵信息的数据结构。文档是与程序开发、维护和使用有关的图文资料。可见软件由两部分组成:一是机器可执行的程序和数据;二是机器不可执行的,与软件开发、运行、维护、使用等有关的文档。

国际(GB)中对计算机软件的定义为:与计算机系统的操作有关的计算机程序、规程、规则以及可能有的文件、文档及数据。

软件在开发、生产维护和使用等方面与计算机硬件相比存在明显的差异。深入理解软件的定义需要了解软件的特点:

(1) 软件是一种逻辑实体,而不是物理实体,具有抽象性。软件的这个特点使它与其他工程对象有着明显的差异。人们可以把它记录在纸上或存储介质上,但却无法看到软件本身的形态,必须通过观察、分析、思考、判断,才能了解它的功能、性能等特性。

(2) 软件的生产与硬件不同, 它没有明显的制作过程。一旦研制开发成功, 可以大量拷贝同一内容的副本。所以对软件的质量控制, 必须着重在软件开发方面下功夫。

(3) 软件在运行、使用期间不存在磨损、老化问题。软件虽然在生存周期后期不会因为磨损而老化, 但为了适应硬件、环境以及需求的变化要进行修改, 而这些修改又会不可避免地引入错误, 导致软件失败率升高, 从而使得软件退化。

(4) 软件的开发、运行对计算机系统具有依赖性, 受计算机系统的限制, 这导致了软件移植的问题。

(5) 软件复杂性高, 成本昂贵。软件是人类有史以来生产复杂度最高的工业产品。软件涉及人类社会各行各业、方方面面, 软件开发常常涉及其他领域的专门知识。软件开发需要投入大量、高强度的脑力劳动, 成本高, 风险大。

(6) 软件开发涉及诸多的社会因素。许多软件的开发和运行涉及软件用户的机构设置、体制问题以及管理方式等, 甚至涉及到人们的观念和心理、软件知识产权及法律等问题。

软件根据应用目标的不同, 是多种多样的。软件按功能可以分为: 应用软件、系统软件、支撑软件(或工具软件)。应用软件是为解决特定领域的应用而开发的软件。例如, 事务处理软件、工程与科学计算软件、实时处理软件、嵌入式软件、人工智能软件等应用性质不同的各种软件。系统软件是计算机管理自身资源、提高计算机使用效率并为计算机用户提供各种服务的软件。如操作系统、编译程序、汇编程序、网络软件、数据库管理系统等。支撑软件是介于系统软件和应用软件之间, 协助用户开发软件的工具性软件, 包括辅助和支持开发以及维护应用软件的软件, 如需求分析工具软件、设计工具软件、编码工具软件、测试工具软件、维护工具软件等, 也包括辅助管理人员控制开发进程和项目管理的工具软件, 如计划进度管理工具软件、过程控制工具软件、质量管理及配置管理工具软件等。

1.1.2 软件危机与软件工程

软件工程概念的出现源自软件危机。

20 世纪 60 年代末以后, “软件危机”这个问题频繁出现。所谓软件危机是泛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。实际上, 几乎所有的软件都不同程度地存在这些问题。

随着计算机技术的发展和应用领域的扩大, 计算机硬件性能价格比和质量稳步提高, 软件规模越来越大, 复杂程度不断增加, 软件成本逐年上升, 质量没有可靠的保证, 软件已成为计算机科学发展的“瓶颈”。

具体地说, 在软件开发和维护过程中, 软件危机主要表现在:

(1) 软件需求的增长得不到满足。用户对系统不满意的情况经常发生。

(2) 软件开发成本和进度无法控制。开发成本超出预算, 开发周期大大超过规定日期的情况经常发生。

(3) 软件质量难以保证。

(4) 软件不可维护或维护程度非常低。

(5) 软件的成本不断提高。

(6) 软件开发生产率的提高赶不上硬件的发展和应用需求的增长。

总之, 可以将软件危机归结为成本、质量、生产率等问题。

分析带来软件危机的原因,宏观方面是由于软件日益深入社会生活的各个层面,对软件需求的增长速度大大超过了技术进步所能带来的软件生产率的提高。而就每一项具体的工程任务来看,许多困难来源于软件工程所面临的任务和其他工程之间的差异以及软件和其他工业产品的不同。

在软件开发和维护过程中,之所以存在这些严重的问题,一方面与软件本身的特点有关,例如在软件运行前,软件开发过程的进展难衡量,质量难以评价,因此管理和控制软件开发过程相当困难。在软件运行过程中,软件维护意味着改正或修改原来的设计。另外,软件的显著特点是规模庞大;复杂度超线性增长,在开发大型软件时,要保证高质量极端复杂困难,不仅涉及技术问题(如分析方法、设计方法、版本控制),更重要的是必须有严格而科学的管理。另一方面与软件开发和维护方法不正确有关,这是主要原因。

为了消除软件危机,通过认真研究解决软件危机的方法,认识到软件工程是使计算机软件走向工程科学的途径,逐步形成了软件工程的观念,开辟了工程学的新兴领域——软件工程学。软件工程就是试图用工程、科学和数学的原理与方法研制、维护计算机软件的有关技术及管理方法。

关于软件工程的定义,国标(GB)中指出,软件工程是应用于计算机软件的定义、开发和维护的一整套方法、工具、文档、实践标准和工序。

1968年在北大西洋公约组织会议(NATO会议)上,讨论摆脱软件危机的办法,软件工程(Software Engineering)作为一个概念首次被提出,这在软件技术发展史上是一件大事。其后的几十年里,各种有关软件工程的技术、思想、方法和概念不断地被提出,软件工程逐步发展成为一门独立的科学。在会议上,德国人Fritz Bauer认为:“软件工程是建立并使用完善的工程化原则,以较经济的手段获得能在实际机器上有效运行的可靠软件的系列方法”。

1993年,IEEE(Institute of Electrical & Electronic Engineers,电气和电子工程师学会)给出了一个更加综合的定义:“将系统化的、规范的、可度量的方法应用于软件的开发、运行和维护的过程,即将工程化应用于软件中”。

这些主要思想都是强调在软件开发过程中需要应用工程化原则。

软件工程包括3个要素,即方法、工具和过程。方法是完成软件工程项目的手段;工具支持软件的开发、管理、文档生成;过程支持软件开发的各个环节的控制、管理。

软件工程的进步是近几十年软件产业迅速发展的重要原动力。从根本上来说,其目的是研究软件的开发技术,软件工程的名称意味着用工业化的开发方法来替代小作坊式的开发模式。但是,几十年的软件开发和软件发展的实践证明,软件开发既不同于其他工业工程,也不同于科学研究。软件不是自然界的有形物体,它作为人类智慧的产物有其本身的特点,所以软件工程的方法、概念、目标等都在发展,有的与最初的想法有了一定的差距。但是认识和学习过去和现在的发展演变,真正掌握软件开发技术的成就,并为进一步发展软件开发技术,以适应时代对软件的更高期望是有极大意义的。

软件工程的核心思想是把软件产品(就像其他工业产品一样)看作是一个工程产品来处理。把需求计划、可行性研究、工程审核、质量监督等工程化的概念引入到软件生产当中,以期达到工程项目的3个基本要素:进度、经费和质量的目标。同时,软件工程也注重研究不同于其他工业产品生产的一些独特特性,并针对软件的特点提出了许多有别于一般工业工程技术的一些技术方法。代表性的有结构化的方法、面向对象方法和软件开发模型及软件开发过程等。

特别地,从经济学的意义上来说,考虑到软件庞大的维护费用远比软件开发费用要高,因而开发软件不能只考虑开发期间的费用,而且应考虑软件生命周期内的全部费用。因此,软件生命周期的概念就变得特别重要。在考虑软件费用时,不仅仅要降低开发成本,更要降低整个软件生命周期的总成本。

1.1.3 软件工程过程与软件生命周期

1. 软件工程过程 (Software Engineering Process)

关于软件工程过程的 ISO 9000 定义如下:软件工程过程是把输入转化为输出的一组彼此相关的资源和活动。

该定义支持了软件工程过程的两方面内涵。

(1) 软件工程过程是指为获得软件产品,在软件工具支持下由软件工程师完成的一系列软件工程活动。基于这个方面,软件工程过程通常包含 4 种基本活动:

- 1) P (Plan) ——软件规格说明。规定软件的功能及其运行时的限制。
- 2) D (Do) ——软件开发。产生满足规格说明的软件。
- 3) C (Check) ——软件确认。确认软件能够满足客户提出的要求。
- 4) A (Action) ——软件演进。为满足客户的变更要求,软件必须在使用的过程中演进。

事实上,软件工程过程是一个软件开发机构针对某类软件产品为自己规定的工作步骤,它应当是科学的、合理的,否则必将影响软件产品的质量。

通常把用户的要求转变成软件产品的过程也叫做软件开发过程。此过程包括对用户的要求进行分析,解释成软件需求,把需求转换成设计,把设计用代码来实现并进行代码测试,有些软件还需要进行代码安装和交付运行。

(2) 从软件开发的观点看,它是使用适当的资源(包括人员、软硬件工具、时间等)为开发软件进行的一组开发活动,在过程结束时将输入(用户要求)转化为输出(软件产品)。

所以,软件工程过程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。软件工程过程应确定方法使用的顺序、要求交付的文档资料、为保证质量和适应变化所需要的管理、软件开发各个阶段完成的任务。

2. 软件生命周期 (Software Life Cycle)

通常,将软件产品从提出、实现、使用维护到停止使用退役的过程称为软件生命周期。也就是说,软件产品从考虑其概念开始,到该软件产品不能使用为止的整个时期都属于软件生命周期。一般包括可行性研究与需求分析、设计、实现、测试、交付使用以及维护等活动,如图 1.1 所示。这些活动可以有重复,执行时也可以有迭代。

还可以将软件生命周期分为如图 1.1 所示的软件定义、软件开发及软件运行维护 3 个阶段。图 1.1 所示的软件生命周期的主要活动阶段是:

(1) 可行性研究与计划制定。确定待开发软件系统的开发目标和总的要求,给出它的功能、性能、可靠性以及接口等方面的可能方案,制定完成开发任务的实施计划。

(2) 需求分析。对待开发软件提出的需求进行分析并给出详细定义,编写软件规格说明书及初步的用户手册,提交评审。

(3) 软件设计。系统设计人员和程序设计人员应该在反复理解软件需求的基础上,给出软件的结构、模块的划分、功能的分配以及处理流程。在系统比较复杂的情况下,设计阶段可

分解成概要设计阶段和详细设计阶段,编写概要设计说明书、详细设计说明书和测试计划初稿,提交评审。

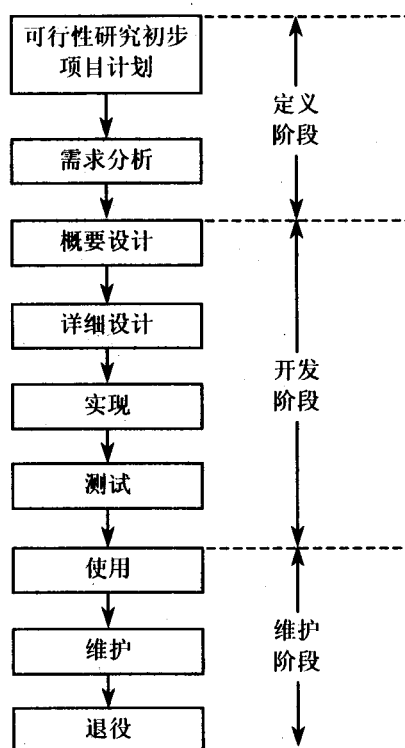


图 1.1 软件生命周期

(4) 软件实现。把软件设计转换成计算机可以接受的程序代码,即完成源程序的编码,编写用户手册、操作手册等面向用户的文档,编写单元测试计划。

(5) 软件测试。在设计测试用例的基础上,检验软件的各个组成部分。编写测试分析报告。

(6) 运行和维护。将已交付的软件投入运行,并在运行使用中不断地维护,根据新提出的需求进行必要而且可能的扩充和删改。

1.1.4 软件工程的目标与原则

1. 软件工程的目标

软件工程的目标是:在给定成本、进度的前提下,开发出具有有效性、可靠性、可理解性、可重用性、可适应性、可移植性、可追踪性和可互操作性且满足用户需求的产品。

软件工程需要达到的基本目标是:付出较低的开发成本,达到要求的软件功能;取得较好的软件性能;开发的软件易于移植;需要较低的维护费用,能按时完成开发,及时交付使用。

基于软件工程的目标,软件工程的理论和技术性研究的内容主要包括:软件开发技术和软件工程管理。

(1) 软件开发技术。软件开发技术包括:软件开发方法学、开发过程、开发工具和软件

工程环境,其主体内容是软件开发方法学。软件开发方法学是根据不同的软件类型,按不同的观点和原则,对软件开发中应遵循的策略、原则、步骤和必须产生的文档资料做出规定,从而使软件的开发能够进入规范化和工程化的阶段,以克服早期的手工方法生产中的随意性和非规范性做法。

(2) 软件工程管理。软件工程管理包括:软件管理学、软件工程经济学、软件心理学等内容。

软件工程管理是软件按工程化生产时的重要环节,它要求按照预先制定的计划、进度和预算执行,以实现预期的经济效益和社会效益。统计数据表明,多数软件开发项目的失败,并不是由于软件开发技术方面的原因,它们的失败是由于不适当的管理造成的。因此人们对软件项目管理重要性的认识有待提高。软件管理学包括人员组织、进度安排、质量保证、配置管理、项目计划等。

软件工程经济学是研究软件开发中成本的估算、成本效益分析的方法和技术,用经济学的基本原理来研究软件工程开发中的经济效益问题。

软件心理学是软件工程领域具有挑战性的一个全新的研究视角,它是从个体心理、人类行为、组织行为和企业文化等角度来研究软件管理和软件工程的。

2. 软件工程的原理

为了达到上述的软件工程目标,在软件开发过程中,必须遵循软件工程的基本原则。这些原则适用于所有的软件项目。这些基本原则包括抽象、信息隐蔽、模块化、局部化、确定性、一致性、完备性和可验证性。

(1) 抽象。抽取事物最基本的特性和行为,忽略非本质细节。采用分层次抽象、自顶向下、逐层细化的办法控制软件开发过程的复杂性。

(2) 信息隐蔽。采用封装技术,将程序模块的实现细节隐藏起来,使模块接口尽量简单。

(3) 模块化。模块是程序中相对独立的成分,是一个独立的编程单位,应有良好的接口定义。模块的大小要适中,模块过大会使模块内部的复杂性增加,不利于对模块的理解和修改,也不利于模块的调试和重用。模块太小会导致整个系统表示过于复杂,不利于控制系统的复杂性。

(4) 局部化。要求在一个物理模块内集中逻辑上相互关联的计算资源,保证模块间具有松散的耦合关系,模块内部有较强的内聚性,这有助于控制解的复杂性。

(5) 确定性。软件开发过程中所有概念的表达应是确定的、无歧义的且规范的。这有助于人与人的交互不会产生误解和遗漏,以保证整个开发工作的协调一致。

(6) 一致性。包括程序、数据和文档的整个软件系统的各模块应使用已知的概念、符号和术语。程序内、外部接口应保持一致,系统规格说明与系统行为应保持一致。

(7) 完备性。软件系统不丢失任何重要成分,完全实现系统所需的功能。

(8) 可验证性。开发大型软件系统需要对系统自顶向下,逐层分解。系统分解应遵循容易检查、测评、评审的原则,以确保系统的正确性。

1.1.5 软件开发工具与软件开发环境

现代软件工程方法之所以得以实施,其重要的保证是软件开发工具和环境的保证,使软件在开发效率、工程质量等多方面得到改善。软件工程鼓励研制和采用各种先进的软件开发方

法、工具和环境。工具和环境的使用进一步提高了软件的开发效率、维护效率和软件质量。

1. 软件开发工具

早期的软件开发除了一般的程序设计语言外,尚缺少工具的支持,致使编程工作量大,质量和进度难以保证,导致人们将很多的精力和时间花费在程序的编制和调试上,而在更重要的软件需求和设计上反而得不到必要的精力和时间投入。软件开发工具的完善和发展将促进软件开发方法的进步和完善,促进软件开发的高速度和高质量。软件开发工具的发展是从单项工具的开发逐步向集成工具发展的,软件开发工具为软件工程方法提供了自动的或半自动的软件支撑环境。同时,软件开发方法的有效应用也必须得到相应工具的支持,否则方法将难以有效的实施。

2. 软件开发环境

软件开发环境或称软件工程环境是全面支持软件开发全过程的软件工具集合。这些软件工具按照一定的方法或模式组合起来,支持软件生命周期内的各个阶段和各项任务的完成。

计算机辅助软件工程(Computer Aided Software Engineering, CASE)是当前软件开发环境中富有特色的研究工作和发展方向。CASE将各种软件工具、开发机器和一个存放开发过程信息的中心数据库组合起来,形成软件工程环境。CASE的成功产品将最大限度地降低软件开发的技术难度并使软件开发的质量得到保证。

1.2 结构化分析方法

软件开发方法是软件开发过程所遵循的方法和步骤,其目的在于有效地得到一些工作产品,即程序和文档,并且满足质量要求。软件开发方法包括分析方法、设计方法和程序设计方法。

结构化方法经过30多年的发展,已经成为系统、成熟的软件开发方法之一。结构化方法包括已经形成了配套的结构化分析方法、结构化设计方法和结构化编程方法,其核心和基础是结构化程序设计理论。

1.2.1 需求分析与需求分析方法

1. 需求分析

软件需求是指用户对目标软件系统在功能、行为、性能、设计约束等方面的期望,需求分析的任务是发现需求、求精、建模和定义需求的过程。需求分析将创建所需的数据模型、功能模型和控制模型。

(1) 需求分析的定义。1997年IEEE软件工程标准词汇表对需求分析定义如下:

- 1) 用户解决问题或达到目标所需的条件或权能。
- 2) 系统或系统部件要满足合同、标准、规范或其他正式规定文档所需具有的条件或权能。
- 3) 一种反映1)或2)所描述的条件或权能的文档说明。

由需求分析的定义可知,需求分析的内容包括:提炼、分析和仔细审查已收集到的需求;确保所有利益相关者都明白其含义并找出其中的错误、遗漏或其他不足的地方;从用户最初的非形式化需求到满足用户对软件产品的要求的映射;对用户意图不断进行提示和判断。

(2) 需求分析阶段的工作。需求分析阶段的工作,可以概括为4个方面:

- 1) 需求获取。需求获取的目的是确定对目标系统的各方面需求。涉及到的主要任务是建

立获取用户需求的方法框架,并支持和监控需求获取的过程。

需求获取涉及的关键问题有:对问题空间的理解,人与人之间的通信,不断变化的需求。

需求获取是在同用户的交流过程中不断收集、积累用户的各种信息,并且通过认真理解用户的各项要求,澄清那些模糊的需求,排除不合理的,从而较全面地提炼系统的功能性与非功能性需求。一般功能性与非功能性需求包括系统功能、物理环境、用户界面、用户因素、资源、安全性、质量保证及其他约束。

要特别注意的是,在需求获取过程中,容易产生诸如与用户存在交流障碍,相互误解,缺乏共同语言,理解不完整,忽视需求变化,混淆目标和需求等问题,这些问题都将直接影响到需求分析和系统后续开发的成败。

2) 需求分析。对获取的需求进行分析和综合,最终给出系统的解决方案和目标系统的逻辑模型。

3) 编写需求规格说明书。需求规格说明书作为需求分析的阶段成果,可以为用户、分析人员和设计人员之间的交流提供方便,可以直接支持目标软件系统的确认,又可以作为控制软件开发进程的依据。

4) 需求评审。在需求分析的最后一步,对需求分析阶段的工作进行复审,验证需求文档的一致性、可行性、完整性和有效性。

2. 需求分析方法

常见的需求分析方法有:

(1) 结构化分析方法。主要包括:面向数据流的结构化分析方法(Structured Analysis, SA),面向数据结构的 Jackson 系统开发(Jackson System Development, JSD)方法,面向数据结构的结构化数据系统开发(Data Structured System Development, DSSD)方法。

(2) 面向对象的分析方法(Object-Oriented Analysis, OOA)。

从需求分析建立的模型的特性来分,需求分析方法又分为静态分析方法和动态分析方法。

1.2.2 结构化分析方法

1. 关于结构化分析方法

结构化分析方法是结构化程序设计理论在软件需求分析阶段的运用。它是 20 世纪 70 年代中期倡导的基于功能分解的分析方法,其目的是帮助弄清用户对软件的需求。

对于面向数据流的结构化分析方法,按照 DeMarco 的定义,“结构化分析就是使用数据流图(DFD)、数据字典(DD)、结构化英语、判定表和判定树等工具,来建立一种新的、称为结构化规格说明的目标文档。”

结构化分析方法的实质是着眼于数据流,自顶向下,逐层分解,建立系统的处理流程,以数据流图和数据字典为主要工具,建立系统的逻辑模型。

结构化分析的步骤如下:

- (1) 通过对用户的调查,以软件的需求为线索,获得当前系统的具体模型。
- (2) 去掉具体模型中非本质因素,抽象出当前系统的逻辑模型。
- (3) 根据计算机的特点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。
- (4) 完善目标系统并补充细节,写出目标系统的软件需求规格说明。
- (5) 评审直到确认完全符合用户对软件的需求。

2. 结构化分析的常用工具

(1) 数据流图 (Data Flow Diagram, DFD)。数据流图是描述数据处理过程的工具, 是需理解的逻辑模型的图形表示, 它直接支持系统的功能建模。

数据流图从数据传递和加工的角度, 来刻画数据流从输入到输出的移动变换过程。数据流图中的主要图形元素与说明如下:

○ 加工 (转换)。输入数据经加工变换产生输出。

→ 数据流。沿箭头方向传送数据的通道, 一般在旁边标注数据流名。

— 存储文件。表示处理过程中存放各种数据的文件。

□ 数据源。表示系统和环境的接口, 属系统之外的实体。

一般通过对实际系统的了解和分析后, 使用数据流图为系统建立逻辑模型。建立数据流图的步骤如下:

第1步: 由外向里——先画系统的输入、输出, 然后画系统的内部。

第2步: 自顶向下——顺序完成顶层、中间层、底层数据流图。

第3步: 逐层分解。

数据流图的建立从顶层开始, 顶层的数据流图形式如图 1.2 所示。

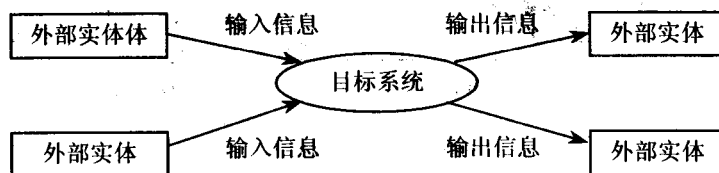


图 1.2 顶层数据流图

顶层数据流图应该包含所有相关外部实体以及外部实体与软件中间的数据流, 其作用主要是描述软件的作用范围, 对总体功能、输入、输出进行抽象描述, 并反映软件和系统、环境的关系。

对复杂系统的表达应采用控制复杂度策略, 顶层数据流图应该包含所有相关外, 需要按照问题的层次结构逐步分解细化, 使用分层的数据流图表达这种结构关系, 分层的数据流图的形式如图 1.3 所示。

为保证构造的数据流图表达完整、准确、规范, 应遵循以下数据流图的构造规则和注意事项:

1) 对加工处理建立惟一、层次性的编号, 且每个加工处理通常要求既有输入又有输出。

2) 数据存储之间不应该有数据流。

3) 数据流图的一致性。它包括数据守恒和数据存储文件的使用, 即某个处理用以产生输出的数据没有输入, 即出现遗漏。另一种是逐个处理的某些输入并没有在处理中使用以产生输出; 数据存储 (文件) 应被数据流图中的处理读和写, 而不是仅读不写或仅写不读。

4) 父图、子图关系与平衡规则。相邻两层 DFD 之间具有父、子关系, 子图代表了父图中某个加工的详细描述, 父图表示了子图间的接口。子图个数不大于父图中的处理个数。所有子图的输入、输出数据流和父图中相应处理的输入、输出数据流必须一致。

(2) 数据字典 (Data Dictionary, DD)。数据字典是结构化分析方法的核心。数据字典是

对所有与系统相关的数据元素的一个有组织的列表，以及精确的、严格的定义，使得用户和系统分析员对于输入、输出、存储成分和中间计算结果有共同的理解。数据字典把不同的需求文档和分析模型紧密地结合在一起，与各模型的图形表示配合，能清楚地表达数据处理的要求。

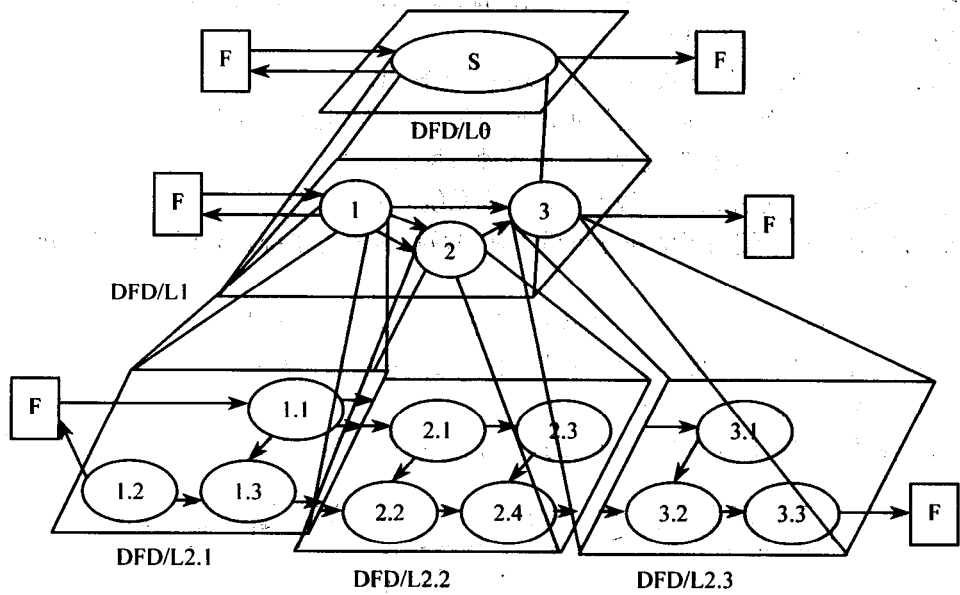


图 1.3 分层数据流

概括地说，数据字典的作用是对 DFD 中出现的被命名的图形元素的确切解释。通常数据字典包含的信息有：名称、别名、何处使用、如何使用、内容描述、补充信息等。例如，对加工的描述应包括：加工名、反映该加工层次的加工编号、加工逻辑及功能简述、输入/输出数据流等。

在数据字典的编制过程中，常使用定义式方式描述数据结构。表 1.1 给出了常用的定义式符号。

表 1.1 数据字典定义式方式中出现的符号

符号	含义
=	表示“等于”，“定义为”，“由什么构成”
[.....]	表示“或”，即选择括号中用“ ”号分隔的各项中的某一项
+	表示“与”，“和”
n{m	表示“重复”，即括号中的项要重复若干次，n, m 是重复次数的上下限
(...)	表示“可选”，即括号中的项可以没有
**	表示“注释”
..	连接符

例如，银行取款业务的数据流图（见图 1.4）中，存储文件“存折”的 DD 定义如下：
存折=‘户名+所号+账户+开户日+性质+(印密)+1{存取行}50