

CHENGXU SHEJI
SHIXUN LILUN JIAOCHENG

程序设计 实训理论教程 (C语言版)

梁志剑 尹四清 洪军 等 编著



国防工业出版社
National Defense Industry Press

程序设计实训理论教程 (C 语言版)

梁志剑 尹四清 洪军 王博 编著

内 容 简 介

本书从程序设计思想理论出发,介绍了软件开发过程经历的需求分析、概要设计、详细设计及软件测试四个阶段,目的是让读者学会利用计算机来解决生活、工作中的实际具体问题。

按照软件开发过程的各阶段要求,本书选择了多个应用领域中的实例加以介绍,内容涉及数据库、计算机网络、图形处理、自动控制、模拟仿真、游戏开发等领域,适用面广;为提高读者举一反三的能力,本书最后介绍了常用的经典算法,对每一种算法的讲解均按照“算法功能→实现方法→函数定义→实例讲解”的思路完成,使读者能透彻理解算法。

本书是作者多年教学和软件开发实践经验的总结,适合作为高等院校计算机、自动控制、自动化、机械、电子等相关专业课程设计或实训的指导书,也可供C语言开发人员参考使用。

图书在版编目(CIP)数据

程序设计实训理论教程: C语言版 / 梁志剑等编著.

—北京:国防工业出版社,2011.6

ISBN 978-7-118-07607-3

I. ①程... II. ①梁... III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第138220号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路23号 邮政编码100048)

北京奥鑫印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 16 $\frac{3}{4}$ 字数 419 千字

2011年6月第1版第1次印刷 印数 1—4000册 定价 35.00元

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422

发行邮购:(010)68414474

发行传真:(010)68411535

发行业务:(010)68472764

前 言

在大多数计算机相关专业的课程体系中,程序设计实践是一项不可或缺的教学环节,多数高校开设了 C 语言作为实践教学环节的先修课程。本书编写的目的之一是使读者在实践中充分应用 C 语言知识,除此之外,作者更希望读者掌握利用计算机解决实际问题的方法。

本书分 3 部分。第一部分是基础篇,共 6 章。第 1 章综述了实训目的、要求、内容、软件开发过程及实训报告的书写方法;第 2 章到第 6 章从软件开发阶段的角度分别介绍了软件需求分析、概要设计、详细设计、编写代码及软件测试五个阶段的基本理论。第二部分是应用篇,共 6 章,针对不同学科、不同专业、不同应用领域为读者提供了多个实例,涉及应用领域包括数据库、计算机网络、图形处理、自动控制、模拟仿真、游戏开发等;每一个实例均按照需求分析、概要设计、详细设计、测试及运行结果步骤加以讲解。第三部分是提高篇,主要介绍了一些常用的算法,包括数值计算、非数值计算两大方面。

全书采用 C 语言作为程序描述语言,目的是让读者更容易掌握,本书中对 C 语言知识的应用,不再局限于基本语法、简单库函数的调用,更多涉及模块化程序设计思想、系统的整体开发过程及有关 C 语言的高级应用等。所有实例程序均在 Turbo C2.0 或 Visual C++6.0 中进行了功能测试及结果验证。考虑到 C 语言的优点及特色,尽量在实例实现过程中充分体现,如采用十字链表数据结构实现文本编辑。

参与编写本书的都是从事了多年程序设计课程教学、实践实训课程教学的具有丰富教学实践经验的教师。全书由梁志剑、尹四清统稿,洪军、王博参与了编写;梁志剑编写了第 7、13 章;尹四清编写了第 1~第 6、第 8 章和附录;洪军编写了第 10、12 章;王博编写了第 9、11 章。另外,在构思本书框架及审阅时得到了潘广贞、康珺、薛海丽等老师的支持与帮助;在实例程序调试、验证、全书格式修改等方面得到了徐菲、吴宗锴等研究生的帮助,在此一并表示深深的感谢。

本书参考引用了一些相关书籍和网站的内容,作者深表感谢。由于时间仓促,书中难免存在疏漏和错误之处,恳请广大读者批评指正。

编 者
2011 年 4 月

目 录

基 础 篇

第1章 实训指导	1	2.6 需求实现	12
1.1 实训目的	1	2.6.1 实现过程定义	12
1.2 实训要求	1	2.6.2 需求状态的变化	12
1.3 实训内容	1	2.6.3 状态变化的追踪	12
1.4 软件开发过程	1	2.6.4 需求的变更控制	12
1.5 实训报告	2	第3章 总体设计	13
第2章 需求分析	3	3.1 软件设计	13
2.1 需求概述	3	3.1.1 软件设计概述	13
2.1.1 过程	3	3.1.2 软件设计策略	13
2.1.2 任务	3	3.2 总体设计	14
2.1.3 原则	4	3.2.1 任务	14
2.1.4 方法	4	3.2.2 结构图	17
2.2 需求收集	4	3.2.3 面向数据流的设计	17
2.2.1 目标	4	方法	17
2.2.2 任务	4	3.2.4 变换分析设计	18
2.2.3 成果	4	3.2.5 事务分析设计	23
2.3 需求结果分析	5	3.2.6 HIPO图	23
2.3.1 目标	5	第4章 详细设计	24
2.3.2 任务	5	4.1 任务	24
2.3.3 成果	5	4.2 结构化程序设计	24
2.4 需求规约	5	4.2.1 基本要点	24
2.4.1 目标	5	4.2.2 控制结构	24
2.4.2 任务	6	4.2.3 特点	25
2.4.3 成果	6	4.3 描述工具	25
2.4.4 结构化分析方法	9	4.4 Jackson方法	25
2.5 需求验证	11	4.4.1 Jackson结构图	25
2.5.1 目标	11	4.4.2 Jackson设计步骤	25
2.5.2 任务	11	4.5 详细设计说明书	26
2.5.3 成果	12		

第5章 编写代码	28
5.1 目的	28
5.2 程序设计	28
5.2.1 程序设计语言分类	28
5.2.2 程序设计语言特性	29
5.2.3 程序设计语言选择	29
5.2.4 程序设计风格	29
第6章 软件测试	31
6.1 测试概述	31
6.1.1 测试目的	31
6.1.2 测试原则	31
6.1.3 调试	31
6.2 测试过程	32

6.2.1 单元测试	32
6.2.2 集成测试	33
6.2.3 确认测试	34
6.2.4 系统测试	34
6.3 测试方法	36
6.3.1 静态测试与动态测试	36
6.3.2 黑盒测试法与白盒测试法	36
6.4 测试工具	36
6.4.1 白盒测试工具	36
6.4.2 黑盒测试工具	36
6.4.3 测试设计和开发工具	37
6.4.4 测试执行和评估工具	37
6.5 测试报告	37

应 用 篇

第7章 数据库管理	39
7.1 个人小管家	39
7.1.1 需求分析	39
7.1.2 概要设计	39
7.1.3 详细设计	41
7.1.4 测试及运行结果	52
7.2 电子通讯录	55
7.2.1 需求分析	55
7.2.2 概要设计	56
7.2.3 详细设计	57
7.2.4 测试及运行结果	71
第8章 图形处理	76
8.1 图形绘制系统	76
8.1.1 需求分析	76
8.1.2 概要设计	76
8.1.3 详细设计	79
8.1.4 测试及运行结果	85
8.2 图形动画系统	89
8.2.1 需求分析	89
8.2.2 概要设计	89

8.2.3 详细设计	92
8.2.4 测试及运行结果	98
第9章 自动控制	101
9.1 交通灯	101
9.1.1 需求分析	101
9.1.2 概要设计	101
9.1.3 详细设计	102
9.1.4 测试及运行结果	107
9.2 电梯控制	108
9.2.1 需求分析	108
9.2.2 概要设计	109
9.2.3 详细设计	110
9.2.4 测试及运行结果	125
第10章 模拟仿真	129
10.1 万年历	129
10.1.1 需求分析	129
10.1.2 概要设计	130
10.1.3 详细设计	131
10.1.4 测试及运行结果	146
10.2 简易DOS操作系统	148

10.2.1	需求分析	148
10.2.2	概要设计	149
10.2.3	详细设计	149
10.2.4	测试及运行结果	162
第 11 章	桌面游戏	165
11.1	贪吃蛇	165
11.1.1	需求分析	165
11.1.2	概要设计	165
11.1.3	详细设计	166
11.1.4	测试及运行结果	172
11.2	推箱子	174
11.2.1	需求分析	174
11.2.2	概要设计	174
11.2.3	详细设计	176
11.2.4	测试及运行结果	197

第 12 章	网络编程	199
12.1	TCP 连接建立系统	199
12.1.1	需求分析	199
12.1.2	概要设计	199
12.1.3	服务器端详细设计	200
12.1.4	客户端详细设计	208
12.1.5	测试及运行结果	213
12.2	ping 命令系统	215
12.2.1	ping 命令工作原理	215
12.2.2	ping 命令参数	216
12.2.3	需求分析	217
12.2.4	概要设计	217
12.2.5	详细设计	217
12.2.6	测试及运行结果	225

提 高 篇

第 13 章	常用算法	227
13.1	数值计算	227
13.1.1	一元多项式求值	227
13.1.2	定积分计算	228
13.1.3	矩阵转置	229
13.1.4	随机数的产生	230
13.2	非数值计算	232
13.2.1	查找	232
13.2.2	排序	234
13.3	其他	238
13.3.1	交换	238
13.3.2	计数	239
13.3.3	累加(乘)	240

13.3.4	求阶乘	241
13.3.5	求最大公约数和 最小公倍数	242
13.3.6	求最值	243
13.3.7	判断素数	244
13.3.8	数制转换	245
13.3.9	字符处理	247
13.3.10	迭代	250
13.3.11	数字分离	251

附录 1	实训报告格式	253
-------------	---------------------	------------

附录 2	参考文献格式示例	260
-------------	-----------------------	------------

参考文献	262
-------------------	------------

基 础 篇

第1章 实训指导

1.1 实训目的

程序设计实训是在学生学完某门程序设计语言课程后进行的一次全面的综合练习,是教学过程中的重要环节,其主要目的是使学生巩固和加深对程序设计语言课程基本知识的理解,掌握程序设计语言编程和程序调试的基本技能,熟悉软件项目开发过程,掌握书写程序设计说明文档的方法,提高运用程序设计语言解决实际问题的能力。

1.2 实训要求

通过分析实训题目的要求,书写需求规格说明书;对题目进行模块划分、总体设计,绘制程序结构图及各模块流程图,并完成概要设计说明书;然后对每一个模块进行详细设计,编写程序代码,调试程序使其正确运行,同时要保证完成的软件便于操作和使用,并完成详细设计说明书;最后进行软件测试并提交实训报告。

1.3 实训内容

生产、生活中的各个领域都离不开程序设计。本书针对以下应用领域设计了实训题目,并进行了详细分析。实训题目涉及的领域包括数据库、计算机网络、图形处理、自动控制、模拟仿真、游戏开发及文本编辑等。

1.4 软件开发过程

一个软件项目的开发过程包括:市场调研、需求分析、概要设计、详细设计、编写代码、测试等阶段。

【第一阶段】市场调研。技术和市场结合才能体现最大价值。

【第二阶段】需求分析。该阶段需要产生3个结果:用户视图、数据词典和用户操作手册。用户视图是该软件用户(包括终端用户和管理用户)看到的页面样式,包含了很多操作方面的流程和条件;数据词典是指明数据逻辑关系并加以整理的内容;用户操作手册是操作流程的说明书。用户操作流程和用户视图是由需求决定的,因此应该在软件设计之前完成,它为程序研发提供了约束和准绳。

【第三阶段】概要设计。将系统功能模块初步划分,并给出合理的开发流程和资源要求。作为快速原型设计方法,完成概要设计就可以进入编码阶段了,通常采用这种方法是因为涉及的研发任务属于新领域,技术主管人员最初无法给出明确的详细设计说明书,并非详细设计阶段不重要,事实上在完成原型代码后,根据测试结果和经验教训的总结,快速原型法还要重新进行详细设计的步骤。

【第四阶段】详细设计。这是考验技术工程师设计思维的重要关卡,本阶段产生的结果是详细设计说明书。详细设计说明书应当把具体的模块以最“干净”的方式(黑箱结构)提供给程序员,使得系统整体模块化达到最大;一份好的详细设计说明书,可以使编码的复杂度降到最低。严格来讲,详细设计说明书应当把每个函数的每个参数的定义都准确提供出来。

【第五阶段】编写代码。在规范化的开发流程中,编码工作在整个开发流程里最多不会超过1/2,通常占1/3的时间,所谓磨刀不误砍柴工,设计过程完成得好,编码效率就会极大提高。编码时不同模块之间的进度协调和协作是最需要注意的,一个小模块的问题就可能影响整体进度,让很多程序员因此被迫停止工作,等待合理的详细设计方案,这种问题在很多开发过程中都出现过。

【第六阶段】测试。软件测试是软件开发中相当重要的一个步骤。测试方法有多种:如按照测试执行方,可分为内部测试和外部测试;按照测试范围,可分为模块测试和整体联调;按照测试条件,可分为正常操作情况测试和异常情况测试;按照测试的输入范围,可分为全覆盖测试和抽样测试。

1.5 实训报告

实训报告是实训过程总结的一种方式。一份合格的实训报告内容包括软件开发过程中各个阶段的描述。读者可参考附录1完成实训报告。

第2章 需求分析

2.1 需求概述

需求分析阶段的目的在于清楚地理解用户需求,并把这些需求系统地组织到一个规约文档中。

需求是产品特性的源头,需求工作的优劣对产品影响最大,就像一条河流,如果源头被污染了,那么整条河流也将被污染。

在产品开发过程中,一个人可能犯下的最大错误是在没有正确记录需求的情况下开始开发活动。需求分析被认为是软件开发中一个很重要的阶段,应该小心仔细地完成。

对于任何一个软件产品而言,需求来源于用户的“需要”,这些“需要”经分析、确认后,以文档的形式记录下来,该文档详细说明了产品“必须或应当”做什么的特性,即需求是正在构建的系统必须符合的事务。

我们知道,一个明确的、保持稳定的用户需求几乎是不可能的,即需求具有不确定性,因此不能以一成不变的观点对待需求,不能只强调需求的记录,更要重视需求实现与维护过程的需求管理。所谓需求管理,即一种获取、组织并记录系统需求的系统化方案以及一个使客户与项目团队对不断变更的系统需求达成并保持一致的过程。换句话说,需求管理就是找出需求是什么,将它们记录下来,进行组织,并在发生变化时对它们进行追踪的活动过程。

2.1.1 过程

需求分析一般由需求收集、需求分析、需求规约、需求验证及需求实现等5个步骤构成。

(1) 需求收集:指通过各种途径收集用户的需求信息,产生《用户需求说明书》,并获得用户的确认。

(2) 需求分析:指针对收集的用户需求信息,从用户需要和系统可能两个方面进行分析,综合平衡,消除其中不一致、异常和不完整的地方,确定需求的细节等。

(3) 需求规约:指根据需求调查和需求分析的结果,进一步定义准确无误的产品需求,产生软件需求规约(SRS)文档。

(4) 需求验证:首先由项目团队对SRS文档进行内部评审,确保它是可理解的、一致的、没有歧义和完整的,之后再由客户评审,使项目开发能最终满足用户的需要。

(5) 需求实现:是在客户与开发方之间建立对需求的共同理解的基础上,实现需求并在实现的过程中维护需求与目标和项目范围的一致性,还要控制需求的变更。此阶段中应以SRS文档作为未来所有开发活动的基础,同时也作为客户和开发组织之间的一份合约文档。

2.1.2 任务

需求分析阶段需完成如下任务:

- (1) 确定对系统的综合要求:功能要求、性能要求、运行要求、将来可能提出的要求;
- (2) 分析系统的数据要求:E-R图(概念模型);
- (3) 导出系统的逻辑模型:数据流图、数据字典、加工处理说明书等;
- (4) 修正系统开发计划;
- (5) 开发原型系统:使用户对目标系统有一个更直接、更具体的概念,从而能更准确地提出用户需求。

2.1.3 原则

需求分析应遵守如下原则:

- (1) 必须能够表达和理解问题的数据域和功能域(数据域:数据流、数据内容和数据结构;功能域:加工变换);
- (2) 必须按自顶向下、逐层分解的方式对问题进行分解和不断细化;
- (3) 给出系统的逻辑视图和物理视图(逻辑视图:软件要达到的功能和要处理的数据之间的关系;物理视图:给出处理功能和数据结构的实际表示形式)。

2.1.4 方法

需求分析方法有面向数据流和面向数据结构两种。

2.2 需求收集

2.2.1 目标

需求收集的目标在于获得用户的需求确认。

2.2.2 任务

通过采访最终客户及研究现有的文档,收集与系统相关的所有可能的信息。

收集工作的难度与客户现有的工作系统模型密切相关。如针对客户现有的工作流程开发一个办公自动化系统(OA系统),分析员只需研究输入数据、输出数据和现有办公流程的确切格式,由于客户对事务流程非常熟悉,分析员可以立刻获得输入和输出数据的格式,任务相对简单;然而,若客户对将要开发系统的需求非常模糊,那么,分析员就很难从客户那里提取完整的信息,此时,分析员可以根据自己的工作经验,充分发挥想象力和创新精神,重新定义一个更加全面、更加完善的功能,最终得到用户的确认。

2.2.3 成果

需求收集工作的成果如下:

- (1) 在问题定义上与客户达成共识;
- (2) 理解问题背后的根本原因;
- (3) 确定用户和项目干系人;
- (4) 定义问题解空间的边界;
- (5) 确定问题解决方案的约束和假设;

(6) 用户对系统目标的认可(获得用户签字确认的《用户需求说明》)。

2.3 需求结果分析

2.3.1 目标

需求收集是面向用户、在较高的抽象级别上对系统特性的定义,它的最终目标是得到用户的认可;而需求分析是面向系统实现、严格对系统需求的定义,所讨论的是在技术上系统应该做什么,即清楚地了解客户的确切需求。

2.3.2 任务

针对具体问题,把握如下几点:

- (1) 问题是什么?
- (2) 为什么解决该问题很重要?
- (3) 该问题有哪些可能的解决方案?
- (4) 输入到系统的数据是什么?
- (5) 系统要求的输出数据是什么?
- (6) 解决问题时可能发生的复杂性是什么?
- (7) 如果开发出的软件必须与外部软件或硬件进行数据交换,那么和外部系统的数据交换格式应该是什么?

2.3.3 成果

完成具有良好的需求规格说明属性的需求文档,应包含如下属性:

- (1) 无歧义性:每一个需求只有唯一的一种解释;
- (2) 完整性:需求文档包括功能、性能、时间响应要求、限制、接口等属性,不存在没有界定的、以为是隐含或默认而实际存在认知差异的需求;
- (3) 可检验性:存在有限的、经济与技术都是可行的检验方法和程序,对需求的实现与否进行检验,使得用户和组织通过该检验,确认需求是按照需求规格说明实现;
- (4) 一致性:需求作为一种要求是一致的,不存在系统内相互冲突的需求要求;
- (5) 可跟踪性:能够判断哪个设计组件对应哪个需求,哪个代码对应哪个设计组件以及哪个测试用例对应哪个需求等;
- (6) 可使用性:可为产品的各阶段特别是维护阶段提供充分有用的信息。

2.4 需求规约

2.4.1 目标

在需求获取阶段,项目产生了《用户需求说明书》,为用户和项目团队提供了共同确认的业务需求的宏观描述文档;需求分析阶段,结合系统设计和实现需要,充分考虑了系统非功能性需求和设计规范要求,使得需求更为全面和完整。而需求规约阶段是根据需求调查和需求分析的结果,进一步定义准确无误的产品需求,产生《需求规格说明书》,对项目需求进行有效

的追踪及管理。

需求规约文档针对不同对象有不同的作用:

- (1) 客户和销售人员:确保系统如 SRS 文档中的所述会满足他们的需要。
- (2) 软件开发人员:确保自己严格按照客户的需求进行开发。
- (3) 测试工程师:确保从功能的角度来看需求是可以理解的,这样他们就能够测试软件并验证其工作。
- (4) 用户手册作者:确保他们能透彻地理解该文档以便撰写用户手册。
- (5) 项目经理:确保只要参照 SRS 文档即能轻松地估测成本,并保证它包含良好规划一个项目所需的全部信息。
- (6) 维护工程师:该文档可帮助他们理解系统的功能,有助于理解设计和代码。

2.4.2 任务

需求规约的主要任务包括:需求文档化、需求形式化、需求分析及定义需求数据库。

(1) 需求文档化:以文档的形式定义需求,即需求一旦确定,就需要将它用文档的形式固定下来,这样可以减少口头的误传、增强需求描述的完整性和无歧义性,也有利于需求实现和变更控制管理。

(2) 需求形式化:通过文字或数据库记录等形式将需求固定下来,具体可以采用伪代码、有限状态机、决策表和决策树、流程图、实体联系模型和形式语言描述等。

(3) 需求分解:为了对需求进行使用和管理,应能够对需求进行分解、允许增加/删除/修改需求,且为每个项目定义其特定的属性集,根据追踪元素类型的不同,属性也应有不同的设置。

(4) 定义需求数据库:根据需求的属性描述,抽象出需求数据字典,建立需求数据库。可用 Rational 工具辅助完成需求数据库的管理。有了需求数据库,我们可以方便地对需求的变化做出完善记录,可以分配资源、评估状态、计算软件指标、管理项目风险、估算成本、确保用户的安全和管理项目规模。

2.4.3 成果

完成《需求规格说明书》,它是对目标产品明确或隐含功能的阐述文件,阐述了一个软件系统必须提供的功能、性能以及必须考虑的限制条件,也可以说,它是软件开发的“宪法”。

《需求规格说明书》的内容包含如下两方面:

(1) 功能需求:清楚地描述系统支持的每个功能,以及相应的输入和输出数据集合,即从一些输入数据到输出数据的转换。为了更好地理解功能需求,可将系统包含的各大模块称为高级需求,每一大模块又可能包含一些小模块,称为高级需求的子需求。在需求规约中,为准确地描述输入数据到输出数据的转换,应定义高级词汇来描述高级需求,子需求应在相应高级需求基础上再描述。为了准确记录功能需求,需要指明系统支持的功能集。要规约一个功能,可以通过识别数据输入到系统时的状态、输入数据域、输出数据域以及在输入数据上执行以获得输出数据的处理类型等来实现。当我们遇到具有复杂逻辑的需求时,可以采用决策树或决策表加以描述。

【例】借阅图书。

可将借阅图书作为高级功能需求(记为 R),包括两个子需求:查找图书(记为 R1)和借阅

图书(记为 R2)。R1 的需求又包括设置查找条件(记为 R1.1)和显示查找结果(记为 R1.2); R2 的需求又包括用户登录(记为 R2.1)和借阅(记为 R2.2)。以下具体说明各需求的系统状态、输入数据和输出数据:

R1:查找图书

用户设置查找条件后,系统会根据输入的关键字在数据库中查找图书,查找成功后,系统输出与查找条件一致的书名、作者、出版社等相关信息。

R1.1:设置查找条件

输入:设置查找选项,如多关键字查找时,采用“与”运算等。

输出:提示用户输入关键字。

R1.2:显示查找结果

输入:关键字。

输出:书名、作者等与查找条件匹配的记录,显示内容包括:书名、作者、ISBN、出版年份、出版社、可借出数量及馆藏位置。

R2:借阅图书

输入用户名及密码,系统验证通过后,显示该用户所有的借阅记录及提示用户还可借阅的数量,对还书日期还有 10 天的借出图书可进行续借,若有人预约则不可续借,显示提示信息。

R2.1:用户登录

输入:用户名和密码。

输出:若成功登录,显示该用户所有的借阅记录及提示可借图书数量。

R2.2:借阅

输入:用户选择欲借图书。

输出:确认图书成功借阅,或提示借阅失败等错误信息。

(2) 非功能需求:无法以函数形式表示的系统特征,如可维护性、可移动性、可用性、可靠性及结果的准确度等。《需求规格说明书》的主要特征如下:

①简洁:去除冗长和无关的描述,尽量简洁,但要保持无歧义、一致和完整性。

②结构化:随时间推移,当客户需求发生变更时,结构化的需求规格说明书易于修改。

③黑匣子视图:只求说清楚“做什么”,不要说明“如何做”。

④异常处理:当向系统输入不合法数据时,说明书应说明系统有正确做出回应的功能。

⑤可验证:说明系统各项功能哪些可验证,哪些不可验证。可验证的功能可供项目验收时参考。

《需求规格说明书》的一般格式如下:

1. 引言

(1) 编写目的

说明编写本文档的目的,指出预期的读者。

(2) 背景

说明:

a. 待开发软件系统的名称。

b. 本项目的任务提出者、开发者、用户及实现该软件的计算中心或计算机网络。

c. 该软件系统同其他系统或其他机构的相互关系。

(3) 定义

列出本文档中用到的专门术语的定义和缩略语。

(4) 参考文献

列出文中出现的重要参考资料,如:

- a. 本项目的经核准的计划任务书或合同、上级机关的批文。
- b. 属于本项目的其他已发表的文件。
- c. 本文档中各处引用的文件、资料,包括所要用到的软件开发标准。列出这些文件资料的标题、文件编号、发表日期和出版单位,说明这些文件资料的来源。

2. 任务概述

(1) 目标

叙述本软件开发的意图、应用目标、作用范围以及其他应向读者说明的有关该软件开发的背景材料。解释被开发软件与其他有关软件之间的关系。如果本软件产品是一项独立的软件,而且全部内容自含,则说明这一点。如果所定义的产品是一个更大系统中的一个组成部分,则应说明本产品与该系统中其他各组成部分之间的关系,为此可使用框图来说明该系统的组成和本产品同其他各部分的联系和接口。

(2) 用户的特点

列出本软件的最终用户的特点,充分说明操作人员、维护人员的教育水平和技术专长,以及本软件的预期使用频度。这些是软件设计工作的重要约束。

(3) 假定和约束

列出进行本软件开发工作的假定和约束,例如经费限制、开发期限等。

3. 需求规定

(1) 对功能的规定

用列表的方式(例如:IPO表即输入、处理、输出表的形式)逐项定量和定性地叙述对软件所提出的功能要求,说明输入什么量经怎样的处理、得到什么输出,说明软件应支持的终端数和应支持的并行操作的用户数。

(2) 对性能的规定

1) 精度

说明对该软件的输入、输出数据精度的要求,可能包括传输过程中的精度。

2) 时间特性要求

说明对于该软件的时间特性要求,如:

- a. 响应时间。
- b. 更新处理时间。
- c. 数据的转换和传送时间。
- d. 解题时间等要求。

3) 灵活性

说明对该软件灵活性的要求,即当需求发生变化时,该软件对这些变化的适应能力,如:

- a. 操作方式的变化。
- b. 运行环境的变化。
- c. 同其他软件接口的变化。
- d. 精度和有效时限的变化。
- e. 计划的变化或改进。

f. 对于为了提供这些灵活性而进行的专门设计应该标明。

(3) 输入输出要求

解释各输入输出数据类型,并逐项说明其媒体、格式、数值范围、精度等。对软件的数据输出及必须标明的控制输出量进行解释并举例,包括对硬拷贝报告(正常结果输出、状态输出及异常输出)以及图形或显示报告的描述。

(4) 数据管理能力要求

说明需要管理的文卷和记录的个数、表和文卷的大小规模,要按可预见的增长对数据及其分量的存储要求做出估算。

(5) 故障处理要求

列出可能的软件、硬件故障以及会对各项性能产生的后果和对故障处理的要求。

(6) 其他专门要求

如用户单位对安全保密的要求,对使用方便的要求,对可维护性、可补充性、易读性、可靠性、运行环境可转换性的特殊要求等。

4. 运行环境规定

(1) 设备

列出运行该软件所需要的硬设备。说明其中的新型设备及其专门功能,包括:

- a. 处理器型号及内存容量。
- b. 外存容量、联机或脱机、媒体及其存储格式,设备的型号及数量。
- c. 输入及输出设备的型号和数量,联机或脱机。
- d. 数据通信设备的型号和数量。
- e. 功能键及其他专用硬件。

(2) 支持软件

列出支持软件,包括要用到的操作系统、编译(或汇编)程序、测试支持软件等。

(3) 接口

说明该软件同其他软件之间的接口、数据通信协议等。

(4) 控制

说明控制该软件的运行的方法和控制信号,并说明这些控制信号的来源。

2.4.4 结构化分析方法

1) E-R 图

E-R 图指实体-联系(Entity-Relationship)方法。实体指客观存在并相互区分的事物;属性指实体所具有的某一特性;联系指现实世界事物之间的联系在信息世界的反映(一对一联系、一对多联系、多对多联系)。

E-R 图中用长方形表示实体型,在方框内写上实体名;用椭圆形表示实体的属性,并用无向边把实体与其属性连接起来;用菱形表示实体间的联系,菱形框内写上联系名;用无向边把菱形分别与有关实体连接,在无向边旁标上联系的类型。若实体之间的联系也具有属性,则把属性和菱形也用无向边连接上。实体联系的表示如图 2-1 所示。

E-R 图的特点包括:两个实体型间允许多种联系;多个实体型间可以有一个联系;一个实体型可以和自身联系;E-R 图与具体的 DBMS 无关,是概念模型中最常用的一种。

【例】用 E-R 图表示某工厂的物资管理的概念模型。

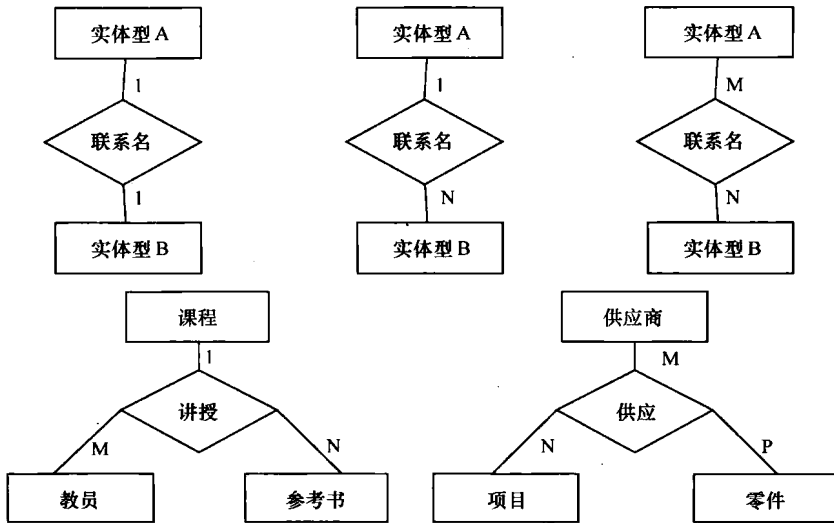


图 2-1 实体联系的表示

实体有：

仓库：仓库号、仓库面积、电话号码。

零件：零件号、名称、规格、单价、描述。

供应商：供应商号、名称、地址、电话号码、账号。

项目：项目号、预算、开工日期。

职工：职工号、姓名、年龄、职称。

联系有：

一个仓库可以存放多种零件，一种零件可以存放在多个仓库中。

一个仓库有多个职工当仓库保管员，一个职工只能在一个仓库工作。

职工之间具有领导和被领导关系，即仓库主任领导若干保管员。

E-R 图如下：

图 2-2 为某工厂的物资管理 E-R 图。

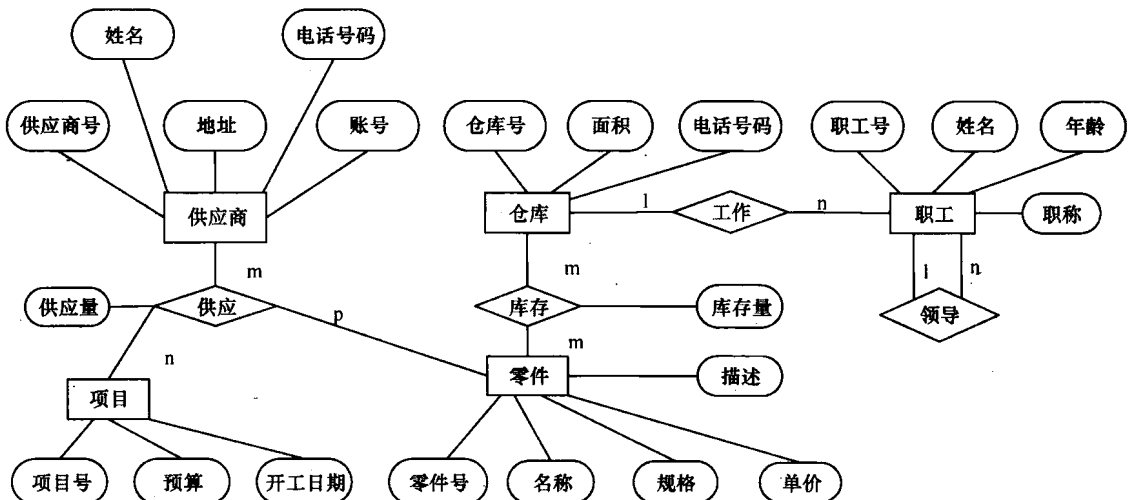


图 2-2 某工厂物资管理 E-R 图