软件工程实务

罗先文　黄　煜　徐　军　编著

重庆大学出版社

内容提要

本书是《软件工程》一书的配套练习和实验指导，结合教材的内容，分别对应每章，共设计了10个实习，每个实习以案例为基础，给出了各文档的编写内容和编写方法，可以帮助读者提高实际动手能力，也可以作为学生结合所学内容进行实务的题目，本书附有大量练习题和参考答案，可供各类学生准备软件工程学科的各类考试。

图书在版编目(CIP)数据

软件工程实务/罗先文，黄煜，徐军编著. —重庆：重庆大学出版社，2005.3
(高职高专计算机系列教材)
ISBN 7-5624-3333-X

Ⅰ.软… Ⅱ.①罗…②黄…③徐… Ⅲ.软件工程—高等学校：技术学校—教材 Ⅳ.TP311.5

中国版本图书馆CIP数据核字(2005)第010844号

软件工程实务

罗先文 黄 煜 徐 军 编著

责任编辑：谭 敏 版式设计：谭 敏

责任校对：廖应碧 责任印制：秦 梅

*

重庆大学出版社出版发行

出版人：张鸽盛

社址：重庆市沙坪坝正街174号重庆大学(A区)内

邮编：400030

电话：(023) 65102378 65105781

传真：(023) 65103686 65105565

网址：http://www.cqup.com.cn

邮箱：fxk@cqup.com.cn (市场营销部)

全国新华书店经销

自贡新华印刷厂印刷

*

开本：787×1092 1/16 印张：8.5 字数：212千

2005年3月第1版 2005年3月第1次印刷

印数：1—3 000

ISBN 7-5624-3333-X 定价：12.00元

前言

软件工程是计算机专业的一门重要的专业基础课，同时也是一门方法学，它对于培养学生的软件素质，提高学生的软件开发能力与软件项目管理能力具有重要的意义。

本实验书是想通过一个项目的开发过程，结合各章知识，让学生学会编写各种文档，因为大多数学生都只注重代码的编写，轻视文档编撰。通过本书的学习，使学生在软件设计的实践及其前后的准备与总结中，复习、领会、巩固和运用软件工程课堂上所学的软件开发方法和知识，为学生适应毕业后团队合作开发规模稍大项目和综合应用本专业所学习的多门课程知识创造实践机会。为学生提供主动学习，积极探索与大胆创新的机会。使学生通过参加小组团队的开发实践，了解项目管理、团队合作、文档编写、口头与书面表达的重要性。同时了解软件工具与环境对于项目开发的重要性。

作为方法性比较强的课程，学生在学习过程中感到比较粗糙，通过加强学生的实务，深入实践的机会，并且通过设计实践中，提高学生的自学能力、创造能力和与团队其他成员交往和协作开发软件的能力，提高学生今后参与开发稍大规模实际软件项目和探索未知领域的能力和自信心。

本书共分两部分，软件工程实务部分主要分为 10 个实务部分，对应教材的各章，习题和参考答案部分也是对应各章的内容。全书由西南农业大学罗先文副教授统稿，黄煜主要负责第一部分的编写，徐军主要负责第二部分的编写。在书稿的编写中，得到了重庆大学出版社和西南农业大学信息学院的大力支持和帮助，在此一并表示衷心的谢意。

作　者

2004 年 12 月于重庆

第 1 部分
软件工程实验实例

实验 1　可行性分析报告

实验目的: 掌握如何制作软件工程的可行性分析报告, 掌握可行性分析报告所包含的内容。

实验要求: 制作一个完整的可行性分析报告。

实验作业: 根据通过的参考实验项目, 写一份软件项目的可行性分析报告。

实验范例:

可行性分析报告

1. 引言

(1) 编写目的

可行性研究的目的是为了对问题进行研究, 以最小的代价在最短的时间内确定问题是否可解。

经过对此项目进行详细调查研究, 初拟系统实现报告, 对软件开发中将要面临的问题及其解决方案进行初步设计及合理安排。明确开发风险及其所带来的经济效益。本报告经审核后, 交项目经理审查。

(2) 项目背景

开发软件名称: 高校图书馆管理系统

项目任务提出者: 某某大学

项目开发者: 某某大学信息学院

用户: 某某大学图书馆

实现软件单位: 某某大学及某某大学信息学院

项目与其他软件, 系统的关系: 本项目采用客户机/服务器原理, 客户端的程序是建立在 Windows NT 系统上以 Microsoft Visual C ++ 为开发软件的应用程序, 服务器端采用 Linux 为操作系统的工作站, 是采用 Oracle 8 为开发软件的数据库服务程序。

(3) 定义

[专门术语]:

[缩写词]:

(4) 参考资料

张海藩,《软件工程导论》. 北京: 清华大学出版社

郑人杰等,《实用软件工程》. 北京: 清华大学出版社

2. 可行性研究的前提

(1) 要求

主要功能: 为学校图书馆的图书进行管理, 和对学生对图书的借阅管理和查询管理, 以及学生对图书的查询。

性能要求: 能够及时反映图书的库存信息, 正确完整的维护图书信息, 快速准确的完成图书信息的查询。

输入要求: 数据完整, 详实。

输出要求: 简捷, 快速, 实时。

安全与保密要求: 保证图书借阅的数据准确完整, 防止学生的借阅信息被修改。

完成期限: 预计 12 个月, 即从 2004 年 1 月到 2004 年 12 月。

(2) 目标

系统实现后, 大大提高图书馆图书借阅的效率, 提高图书库存信息的准确性和完整性, 提高对图书信息的检索效率。

(3) 条件, 假定和限制

建议软件寿命: 5 年。

经费来源: 某某高校图书馆。

硬件条件: 服务器 sun 工作站, 终端为 pc 机。

运行环境: Linux

数据库: Oracle8

投入运行最迟时间: 2005/02/01

(4) 决定可行性的主要因素

成本/效益分析结果, 效益 >> 成本。

技术可行, 现有技术可完全承担开发任务。

操作可行, 软件能被原有工作人员快速接受。

3. 技术可行性分析

(1) 系统简要描述

在图书馆中的终端是安装了 Windows NT 的 PC 机, 主要目的是向图书馆的服务器传递数据。当学生在图书馆进行查询时, 终端向服务器发出查询请求, 服务器根据图书信息库的实时数据, 向终端发送数据, 显示在终端的屏幕上。当学生向操作员借书时, 终端向服务器发出借阅请求, 服务器核对后, 存入借阅信息库, 并修改图书库存信息库。当学生还书时, 终端向服务器发出还书请求, 服务器接收后, 查询借阅信息库, 核对后, 修改图书库存信息库。

(2) 处理流程和数据流程如图 1.1 所示

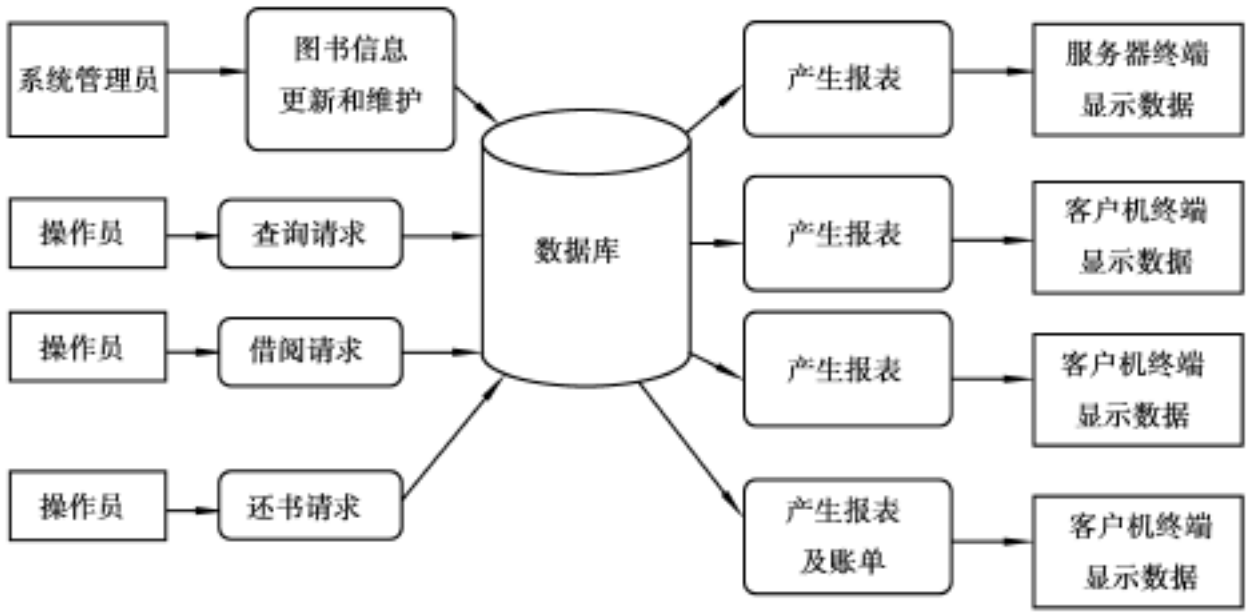


图 1.1　数据处理流程

4. 经济可行性分析

(1) 支出

基础投资: 终端 PC 机 20 台　5 000× 18 = 9 万

网络设备　10 万

辅助配置　5 万

共计:24 万

其他一次性投资: Oracle 8.0　10 万

Windows 2000　1 万

操作员培训费　1 万

共计:12 万

经常性支出: 人工费用　12(月)× 10(人)× 5 000(元) =60 万

其他不可知额外支出　4 万

共计:64 万

支出共计:　100 万

(2) 效益

一次性收益 0 元

经常性收益(按银行利率: 1%) ;

减少员工 20 人(1 000 元/人) 五年收益

$1\ 000\times [1.1+(1.1)^2+(1.1)^3+(1.1)^4+(1.1)^5]\times 20\times 12\times 5 = 120$ 万

工作效率提高收益(工作效率提高 30%)

$30\times [1.1+(1.1)^2+(1.1)^3+(1.1)^4+(1.1)^5]\times (30\%)\times 5 = 45$ 万

经常性收益共计: 160 万

不可定量收益

因服务质量提高增加 10%:

$1\ 000\text{ 万}\times 10\%\times [90\%+(90\%)^2+(90\%)^3+(90\%)^4+(90\%)^5] = 360$ 万

收益共计:　520 万

(3) 收益/投资比

520 万/151 万 = 344%

(4) 投资回收周期

2.3 年

(5) 敏感性分析

设计系统周期为 5 年, 估计最长可达 10 年

处理速度: 一般查询速度 <4 秒

关键数据查询速度: <2 秒

5. 社会因素可行性分析

(1) 法律因素

所有软件都选用正版

所有技术资料都由提出方保管

合同制定确定违约责任

(2) 用户使用可行性

使用本软件人员要求有一定计算机基础的人员, 系统管理员要求有计算机的专业知识, 所有人员都要经过本公司培训。

管理人员也需经过一般培训。

经过培训人员将会熟练使用本软件。

两名系统管理员, 一名审计员将进行专业培训, 他们将熟练管理本系统。

6. 其他可供选择的方案

客户端与服务器端联系在一起。

数据输入由终端输入, 所有数据都由服务器处理, 只在终端上显示数据结果。

此设计简化了数据处理, 但加重了服务器的数据处理, 而使用客户端/服务器机理, 简化数据流量, 加快数据处理。

7. 结论意见

由于投资效益比远大于 100%, 技术、经济、操作都有可行性, 可以进行开发。

注: 以上数据仅供参考, 非真实数据。

实验 2　项目开发计划

实验目的: 掌握如何制作软件工程的项目开发计划, 掌握项目开发计划的具体步骤。

实验要求: 制作一个完整的项目开发计划。

实验作业: 根据提供的实验项目制作项目开发计划。

实验范例:

项目开发计划

1. 引言

(1) 编写目的

本报告的主要作用是确定各个项目模块的开发情况和主要的负责人, 供各项目模块的负责人阅读, 做到及时协调, 按步有序地进行项目开发, 减少开发中的不必要损失。

具体步骤: 拟订开发计划书, 分配项目工作, 安排项目进度。

计划对象: 某某大学信息学院。

(2) 项目背景

由于老的操作程序已经不适应激烈的市场竞争了, 图书馆为了迎合现信息技术发展的需求, 快速发展, 提高工作效率, 提出了新的系统要求。

(3) 定义

(4) 参考资料

2. 项目概述

(1) 工作内容

各工作小组根据时间先后安排, 分别对项目进行开发:

各项主要工作: 需求分析小组对图书馆进行调研(为期一个月) 。

软件开发小组对调查结果进行分析, 拟订实现方案(如程序结构, 流程, 数据结构等) 。

软件编程小组对软件进行集中开发。

软件审核小组对软件进行评定, 审核。

(2) 条件与限制

完成项目应具备的条件:

资金

调研环境

开发平台

开发基础设施

开发人员

维护人员

开发单位已具有的条件:

开发基础设施

开发平台

尚需创造的条件:

良好的调研环境

资方应提供足够的资金和开发条件, 并详细的阐明要求

工方应依据资方的要求开发出满足合同要求的工程

(3) 产品

①程序

程序名称:《图书馆借阅管理系统》

使用语言: Microsoft Visual C ++
存储形式: 磁盘, 光盘, 移动硬盘
②文档
需提交的文档:
项目开发计划
资金分配方案
系统使用手册
系统维护手册
详细技术资料
(4) 运行环境
运行硬件环境:
工作站: PⅢ1.8G PC
运行软件环境: Oracle 8.0, Windows 2000
服务器: 开发单位向用户提供服务: 人员培训, 系统安装, 保修(3 年), 维护(5 年)。
验收标准: 系统运行流畅, 出错率为: 1 次/年。
3. 实施计划
(1) 任务分解

分析阶段(一个月)	调研小组
设计阶段(两个月)	设计小组
写代码及单元测试阶段(三个月)	开发小组
总测试及修改阶段(两个月)	测试小组
维护阶段(不定)	维护小组

(2) 进度
一个月进行调研。
五个月进行实现。
两个月进行测试, 维护。
一个月进行实践。
预算

分析阶段	5 000 元
设计阶段	五万至十万
写代码及单元测试阶段	十万左右
总测试及修改阶段	五万左右

(3) 关键问题
关键的问题是如何做到大容量, 多并发, 快速的即时演算能力和部分故障不停机的能力。此外开发本项目需要一定的风险, 主要是计算机发展速度的风险。详细的分析参见可行性分

析报告。

大多数技术问题都能通过数据库解决，所以选择好的数据库是保证开发完整的前提。

4. 人员组织及分工

调研小组：张强，王磊，刘晓宇

设计小组：张强，王磊，杨光强，薛海峰

开发小组：张强，赵茂，谭伟

测试小组：刘晓宇，杨光强

维护小组：赵茂，谭伟

5. 交付期限

最迟交付日期：2005 年 2 月 1 日

6. 专题计划要点

实验 3　需求分析说明书

实验目的：掌握系统分析需求所包含的 3 个内容，分别是功能需求、数据库需求以及逻辑结构需求，掌握各自的编写方法。

实验要求：制作一个完整的系统需求分析报告。

实验作业：根据范例，选择一个参考项目编写系统需求分析报告。

实验范例：

系统需求分析

1. 高校图书馆管理系统的功能要求

图书馆借阅管理系统的总目标是：在计算机网络，数据库和先进的开发平台上，利用现有的软件，配置一定的硬件，开发一个具有开放体系结构的、易扩充的、易维护的、具有良好人机交互界面的图书借阅管理系统，实现图书借阅管理的自动化和图书查询快速化的计算机系统，为提高图书借阅的效率和方便学生查询图书的信息管理系统。

根据可行性研究的结果和客户的要求，分析现有情况及问题，采用 Client /Server 结构，将图书借阅管理系统划分为 3 个子系统：图书管理员子系统，图书工作人员子系统、学生查询子系统。

图书管理系统需要满足来自三方面的需求，这三个方面分别是图书借阅者、图书馆工作人员和图书馆管理人员。图书借阅者的需求是查询图书馆所存的图书、个人借阅情况及个人信息的限制性修改；图书馆工作人员对图书借阅者的借阅及还书要求进行操作，同时形成借书或还书报表给借阅者查看确认，并处理学生超期和图书损坏、遗失的罚款处理；图书馆管理人员的功能最为复杂，包括对工作人员、图书借阅者、图书进行管理和维护，及系统状态的查看、维护并生成催还图书报表。

图书借阅者可直接查看图书馆图书情况，如果图书借阅者根据本人借书证号和密码登录系统，还可以进行本人借书情况的查询和维护部分个人信息。一般情况下，图书借阅者只应该查询和维护本人的借书情况和个人信息，若查询和维护其他借阅者的借书情况和个人信息，就

要知道其他图书借阅者的借书证号和密码。这些是很难得到的,特别是密码,所以不但满足了图书借阅者的要求,还保护了图书借阅者的个人隐私。

图书馆工作人员有修改图书借阅者借书和还书记录的权限,所以需对工作人员登陆本模块进行更多的考虑。在此模块中,图书馆工作人员可以为图书借阅者加入借书记录或是还书记录,并打印生成相应的报表给用户查看和确认。同时还要处理借阅者的超期和对书的遗失、损坏等方面的罚款处理。

图书馆管理人员功能的信息量大,数据安全性和保密性要求最高。本功能实现对图书信息、借阅者信息、总体借阅情况信息的管理和统计、工作人员和管理人员信息查看及维护。图书馆管理员可以浏览、查询、添加、删除、修改、统计图书的基本信息;浏览、查询、统计、添加、删除和修改图书借阅者的基本信息,浏览、查询、统计图书馆的借阅信息,但不能添加、删除和修改借阅信息,这部分功能应该由图书馆工作人员执行,但是,删除某条图书借阅者基本信息记录时,应实现对该图书借阅者借阅记录的级联删除。并且还应具有生成催还图书报表,并打印输出的功能。

以下为该系统主要完成的功能:

◆设计不同用户的操作权限和登陆方法

◆对所有用户开放的图书查询

◆借阅者维护借阅者个人部分信息

◆借阅者查看个人借阅情况信息

◆维护借阅者个人密码

◆根据借阅情况对数据库进行操作并生成报表

◆根据还书情况对数据库进行操作并生成报表

◆根据借阅者的对书的保护情况进行罚款处理并生成相应的报表

◆查询及统计各种借阅者和图书信息

◆维护图书信息

◆维护工作人员和管理员信息

◆维护借阅者信息

◆处理信息的完整性

◆对借阅过期的图书生成报表

整个系统根据中型图书馆的实际情况进行设计的,并基本上完成了学校对借阅者的借阅管理和对图书的有效管理。

根据实际情况,图书馆借阅管理的流程如图 1.2 所示:

根据以下图书馆借阅管理处理流程以及功能需求情况,可以得到用户的功能需求。

2. 高校图书馆管理系统的数据需求

图书馆借阅管理系统的数据需求包括如下几点:

数据录入和处理的准确性和实时性

数据的输入是否准确是数据处理的前提,错误的输入会导致系统输出的不正确和不可用,从而使系统的工作失去意义。数据的输入来源是手工输入。手工输入要通过系统界面上的安排系统具有容错性,并且对操作人员要进行系统的培训。

在系统中,数据的输入往往是大量的,因此系统要有一定的处理能力,以保证迅速的处理

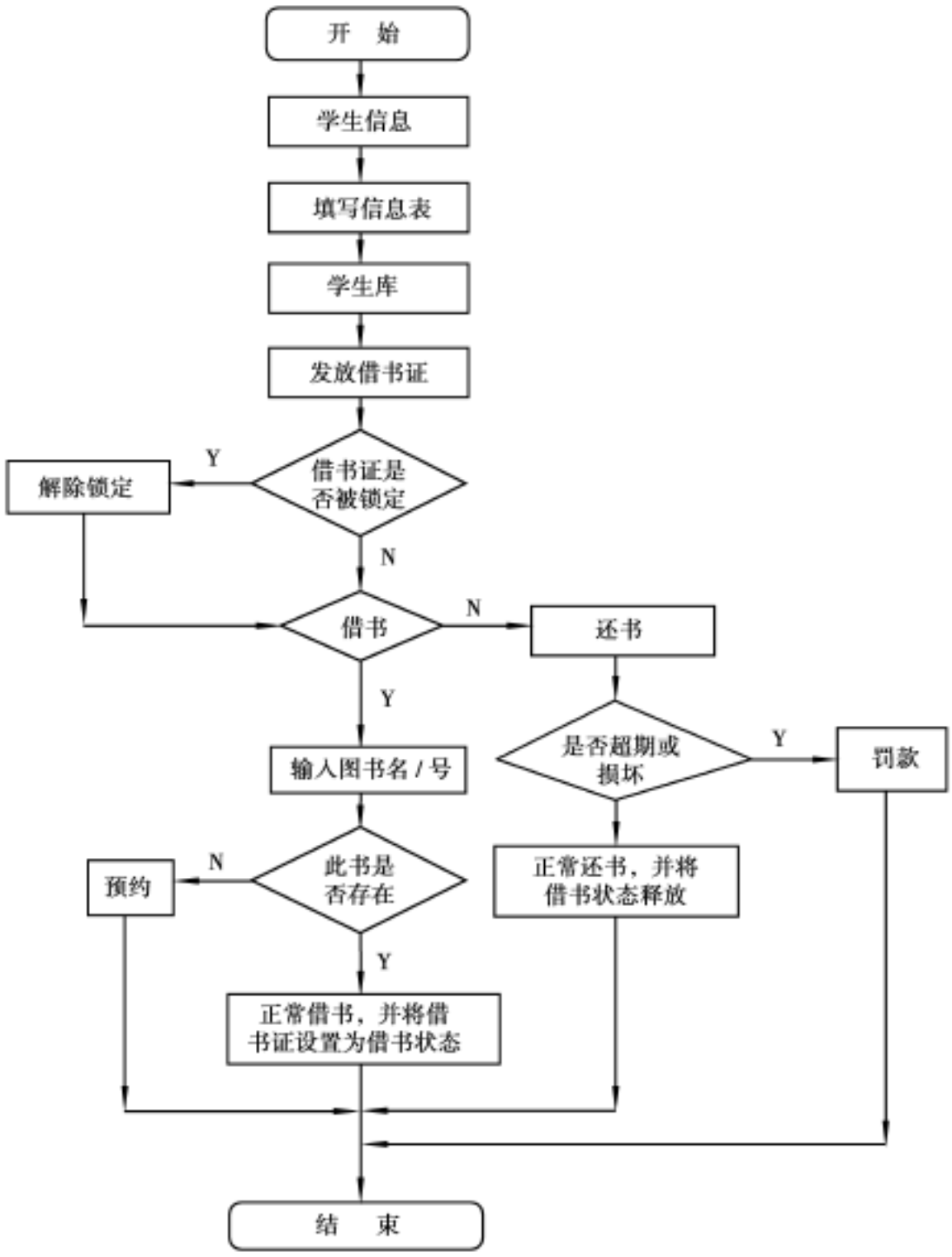


图 1.2　图书馆借阅管理系统处理流程图

数据。

数据的一致性与完整性

由于系统的数据是共享的，在不同的图书馆工作员所操作的计算机中，图书的库存量是共享数据，所以如何保证这些数据的一致性，是系统必须解决的问题。要解决这一问题，要有一定的人员维护数据的一致性，在数据录入处控制数据的去向，并且要求对数据库的数据完整性进行严格的约束。

对于输入的数据，要为其定义完整性规则，如果不能符合完整性约束，系统应该拒绝该数据。

数据的共享与独立性

整个图书借阅系统的数据是共享的。然而，从系统开发的角度上看，共享会给设计和调试带来困难。因此，应该提供灵活的配置，使各个分系统能够独立运行，而通过人工干预的手段

进行系统数据的交换。这样，也能提供系统的强壮性。

图书馆借阅管理系统的数据字典

(1) 图书信息数据字典

名字: 图书信息
别名:
描述: 图书的基本信息，用于图书的确认
定义: 图书信息 = 书号 + 书名 + 作者 + 出版社 + 出版日期 + 价格 + 页数 + 内容简介
位置: 输各个客户机端

用户信息数据字典

名字: 用户信息
别名:
描述: 用户的基本信息，用于用户的确认其自己的基本信息
定义: 用户信息 = 用户码 + 用户名 + 性别 + 班级 + 密码 + 类型 + 备注
位置: 输各个客户机端

借书信息数据字典

名字: 借书信息
别名:
描述: 用于用户借书时的确认信息
定义: 用户信息 = 用户码 + 书号 + 拷贝号 + 借书日期 + 还书日期
位置: 输入到数据库服务器

图书状态信息表:

名字: 图书状态信息
别名:
描述: 用于记录图书的库存状况
定义: 图书状态信息 = 书号 + 拷贝号 + 状态
位置: 输出到客户端

(2) 安全性设计

考虑到数据库的安全性问题，我们进行了如下处理:

①防止用户直接操作数据库的方法

我们建立了图书馆管理系统口令表，用来管理图书馆管理员的信息，完成登录的限制。用户进入用户界面之后，仅可以进入“ 用户查询 ”和“ 新书通报 ”两个系统中，进行继续的查询，而

不能进入管理员的界面。

②用户账号密码的加密方法

管理员通过输入操作员名及口令进入管理系统,所用的口令进行加密处理,确保在任何地方都不会出现密码的明文。

③角色与权限

本系统面向三种角色,一种是图书馆管理员,另一种是图书馆工作人员,还有一种是借阅者。

图书馆管理员的操作权限是对整个数据库进行管理,如用户信息管理(注册,修改,删除等)、书籍信息管理(新书入库,旧书清理,信息修改等)、系统维护(数据备份,数据库恢复,注册管理员,更改密码等)。管理员可以浏览整个系统,并在系统中履行管理员的职责。

图书馆工作人员实现对借阅者的借书和还书的登记出来。

借阅者对该系统的使用仅限于查询所需书籍、查看新书信息并最后通过管理员完成借书及还书活动。

角色	可以访问的表与列	操作权限
图书馆管理员	书库　BOOK 用户库　USER 状态库　STATUS	用户信息管理、书籍信息管理、系统维护等
图书馆工作人员	书库　BOOK 借书库　LOAN	借还管理、罚款处理
借阅者	书库　BOOK 借书库　LOAN	查询所需书籍、查看新书信息

3. 高校图书馆管理系统的逻辑模型

高校图书馆借阅管理系统的逻辑模型如图 1.3 所示:

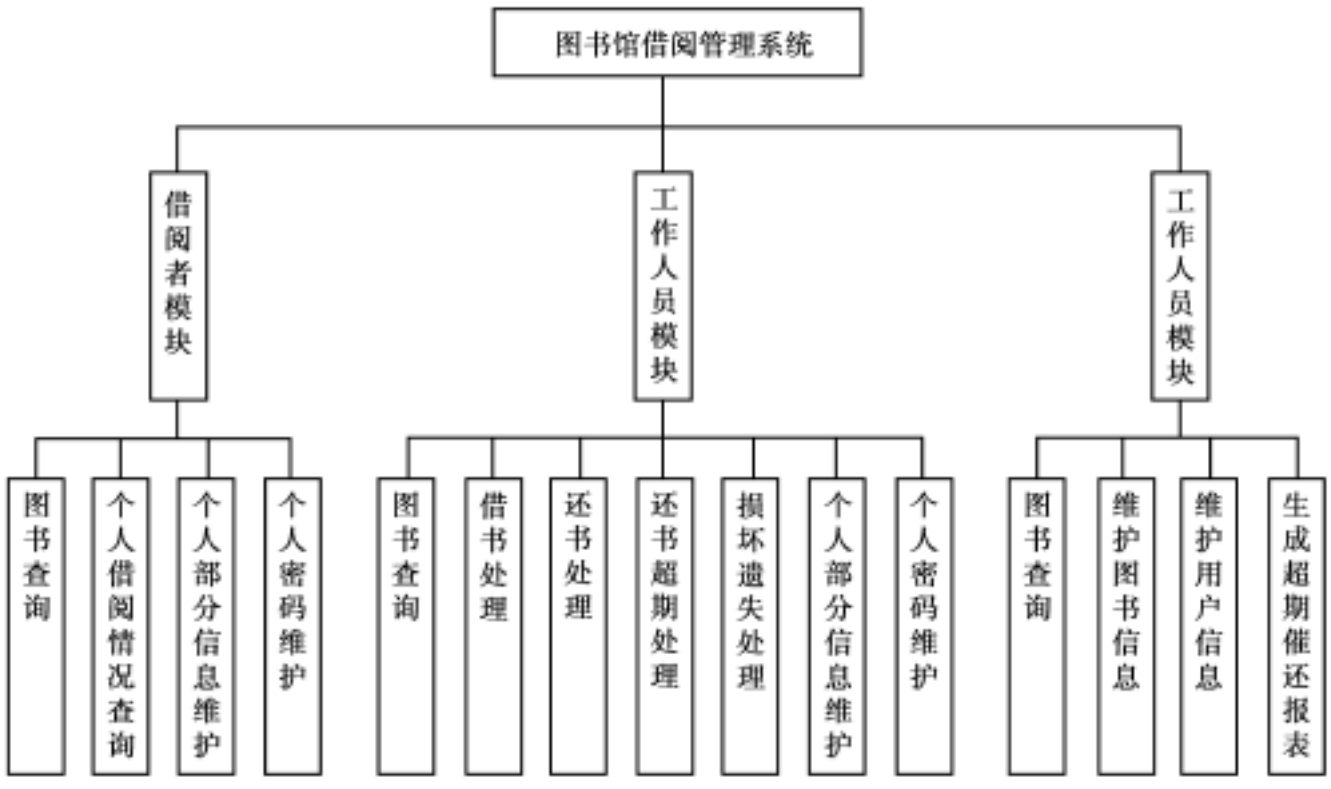


图 1.3　图书馆借阅管理系统的逻辑模型

本课题设计在充分进行用户需求分析的基础上把系统划分为 15 个子模块: 图书查询、读者借书、读者还书、图书信息维护、用户信息维护、图书超期和损坏处理以及相关的报表等。这 15 个模块之间, 紧密结合, 共享信息资源, 形成一种完美的学校图书馆借阅管理的解决方案。

其系统的主要功能结构说明如下:

(1) 用户信息维护及注册: 工作人员通过计算机对学生进行信息注册, 发放借书证, 并可以对各类用户相关信息进行维护;

(2) 图书浏览查询: 用户通过此模块来查询相关需要的图书;

(3) 读者借书: 借阅者通过此模块来完成借阅图书;

(4) 读者还书: 借阅者通过此模块来完成退还图书;

(5) 维护图书信息: 维护图书的基本信息和库存信息;

(6) 图书超期和损坏处理: 对超期或损坏遗失要进行罚款处理;

(7) 报表处理: 比如生成催还图书报表、借阅者信息报表等报表的生成。

实验 4　概要设计说明书

实验目的: 由需求分析而进一步细化的概要设计, 主要掌握用户的功能需求基础上的细化工作, 并能够尽可能地让项目组人员看得懂的说明书, 掌握其设计的方法和步骤。

实验要求: 学会编写系统概要设计说明书。

实验作业: 根据提供的范例编写一个概要设计说明书。

实验范例:

概要设计说明书

1. 引言

(1) 编写目的

在本图书馆借阅管理系统项目的前一阶段, 也就是需求分析阶段中, 已经将系统用户对本系统的需求做了详细的阐述, 这些用户需求已经在上一阶段中对用户需求的调研中获得, 并在需求规格说明书中得到详尽的叙述及阐明。

本阶段已在系统的需求分析的基础上, 图书馆借阅管理系统作概要设计。主要解决了实现该系统需求的程序模块设计问题。包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息, 以及数据结构、模块结构的设计等。在以下的概要设计报告中将对在本阶段中对系统所做的所有概要设计进行详细的说明。

在下一阶段的详细设计中, 程序设计员可参考此概要设计报告, 在概要设计对图书馆借阅系统所做的模块结构设计的基础上, 对系统进行详细设计。在以后的软件测试以及软件维护阶段也可参考此说明书, 以便于了解在概要设计过程中所完成的各模块设计结构, 或在修改时找出在本阶段设计的不足或错误。

(2) 项目背景

本项目(图书馆借阅管理系统) 受某某高校图书馆委托, 由 KAVA 软件开发小组负责开发。

图书馆借阅管理系统将由三部分组成: 学生前台查询管理子系统、图书管理员前台借阅管理子系统和图书管理员后台管理子系统。与其他系统的关系如图 1.4 所示:

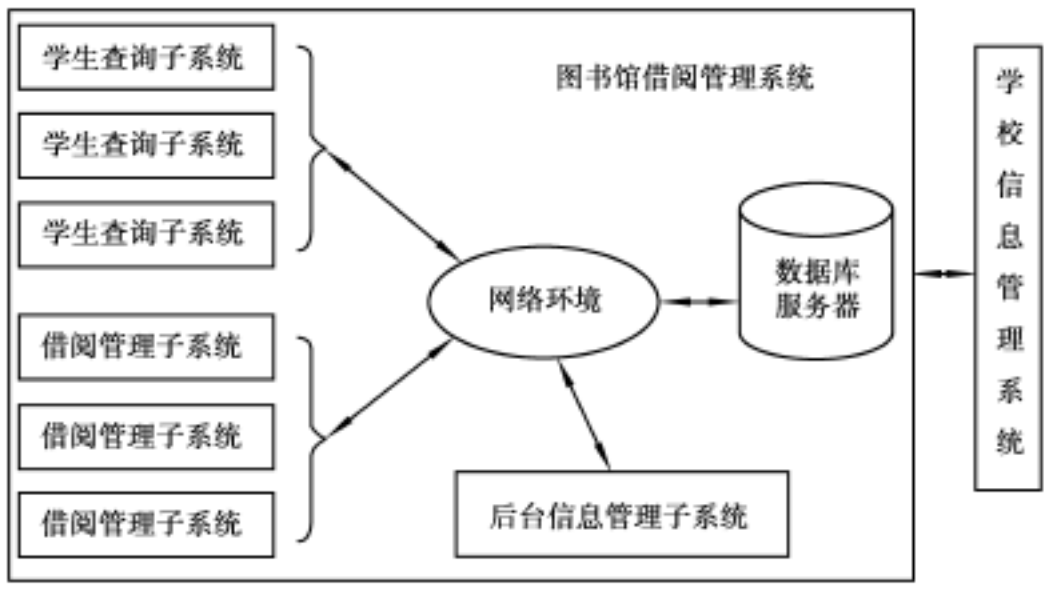


图 1.4　图书馆借阅管理系统与其他系统的关系

(3) 定义

①专门术语

SQL SERVER: 系统服务器所使用的数据库管理系统(DBMS) 。

SQL: 一种用于访问查询数据库的语言

事务流: 数据进入模块后可能有多种路径进行处理。

主键: 数据库表中的关键域。值互不相同。

外部主键: 数据库表中与其他表主键关联的域。

ROLLBACK: 数据库的错误恢复机制。

②缩写

系统: 若未特别指出, 统指本机票预定系统。

SQL: Structured Query Language(结构化查询语言) 。

(4) 参考资料

以下列出在概要设计过程中所使用到的有关资料:

图书馆借阅管理系统项目计划任务书	某某高校图书馆
图书馆借阅管理系统项目开发计划	KAVA 软件开发小组
需求规格说明书	KAVA 软件开发小组

2. 任务概述

(1) 目标

实现图书馆借阅管理的基本需求。

(2) 运行环境

系统将由三部分程序组成, 安装在各学生查询厅的每个客户机、图书馆工作人员办公室和图书馆管理人员办公室。

根据调研得知各客户端的计算机配置均在 Pentium 133 级别以上, 客户程序应能够在 Pentium 133 级别以上, Windows 环境下运行。

(3) 需求概述

计算机进行信息管理与信息管理系统的开发密切相关, 系统的开发是系统管理的前提。

本系统就是为了管理好图书馆信息而设计的。图书馆作为一种信息资源的集散地,图书和用户借阅资料繁多,包含很多的信息数据的管理,现今,有很多的图书馆都是初步开始使用,甚至尚未使用计算机进行信息管理。根据调查得知,他们以前对信息管理的主要方式是基于文本、表格等纸介质的手工处理,对于图书借阅情况(如借书天数、超过限定借书时间的天数)的统计和核实等往往采用对借书卡的人工检查进行,对借阅者的借阅权限以及借阅天数等用人工计算、手抄进行。数据信息处理工作量大,容易出错;由于数据繁多,容易丢失,且不易查找。总的来说,缺乏系统,规范的信息管理手段。尽管有的图书馆有计算机,但是尚未用于信息管理,没有发挥它的效力,资源闲置比较突出,这就是管理信息系统的开发的基本环境。使图书管理工作规范化,系统化,程序化,避免图书管理的随意性,提高信息处理的速度和准确性,能够及时、准确、有效的查询和修改图书情况。

要求系统能有效、快速、安全、可靠和无误的完成上述操作。并要求客户机的界面要简单明了,易于操作,服务器程序利于维护。

(4)条件与限制

需要相应的空间和投资,建立网络环境和购置一定数量的计算机,以及对工作人员的培训和管理意识的更改。

3. 总体设计

(1)处理流程

下面将使用(结构化设计)面向数据流的方法对图书馆借阅管理系统的处理流程进行分析。系统可分为三大部分:a. 学生查询管理模块,b. 各个图书室图书馆工作人员的图书借阅管理模块,c. 图书馆管理人员的图书信息和人员管理管理模块。以下将分别对系统的这三大部分进行流程分析:

①学生查询管理模块

学生查询模块要完成的重要功能是学生输入要查询的图书信息和学生自己的个人信息,输出的是根据学生输入的图书信息的资料在图书库中找出满足条件的信息,以及学生的个人信息并能够维护个人部分信息。其内部处理流程如图 1.5 所示:

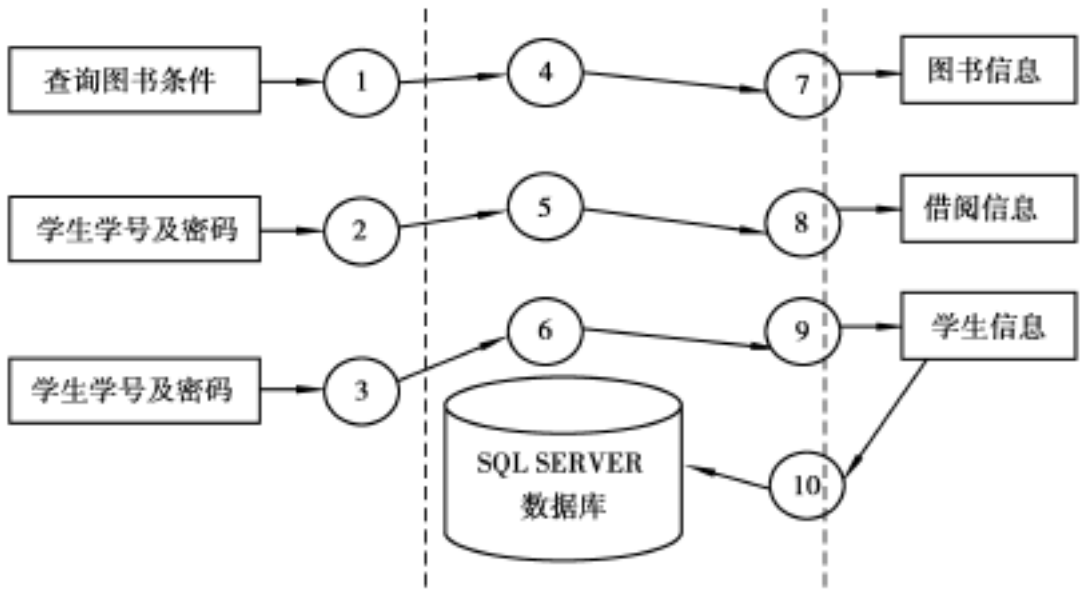


图 1.5　学生查询管理模块内部处理流程

下面对各模块(以数字表示)进行功能说明:

a. 生成查询条件:由客户端计算机生成图书的查询条件;

b. 学生登录信息: 由客户端计算机生成学生账户的登录信息, 用于查看借阅情况;

c. 学生登录信息: 由客户端计算机生成学生账户的登录信息, 用于查看和维护学生本人的部分信息;

d. 图书查询处理: 根据输入的查询条件对数据库进行查询处理;

e. 借阅查询: 根据学生的账号查询学生的借阅情况;

f. 用户信息查询: 根据学生的账号搜索出学生的基本信息;

g. 显示图书信息: 根据查询处理, 返回查询图书查询结果;

h. 显示用户借阅情况: 根据用户的账户查询借阅情况, 返回学生的借阅情况;

i. 显示用户基本信息: 根据用户信息查询返回其结果, 以便用于维护;

j. 修改用户信息: 对作出修改的信息进行处理;

②图书馆工作人员和管理人员处理模块

由于图书馆工作人员和管理人员两个子系统在功能上有很多相同的功能, 只是在某些方面由于权限分配不同而出现相对的一些偏差, 还有就是两种类型的用户在工作上也有些偏重, 但操作的功能模块是一致的, 其功能结构图如图 1.6 所示:

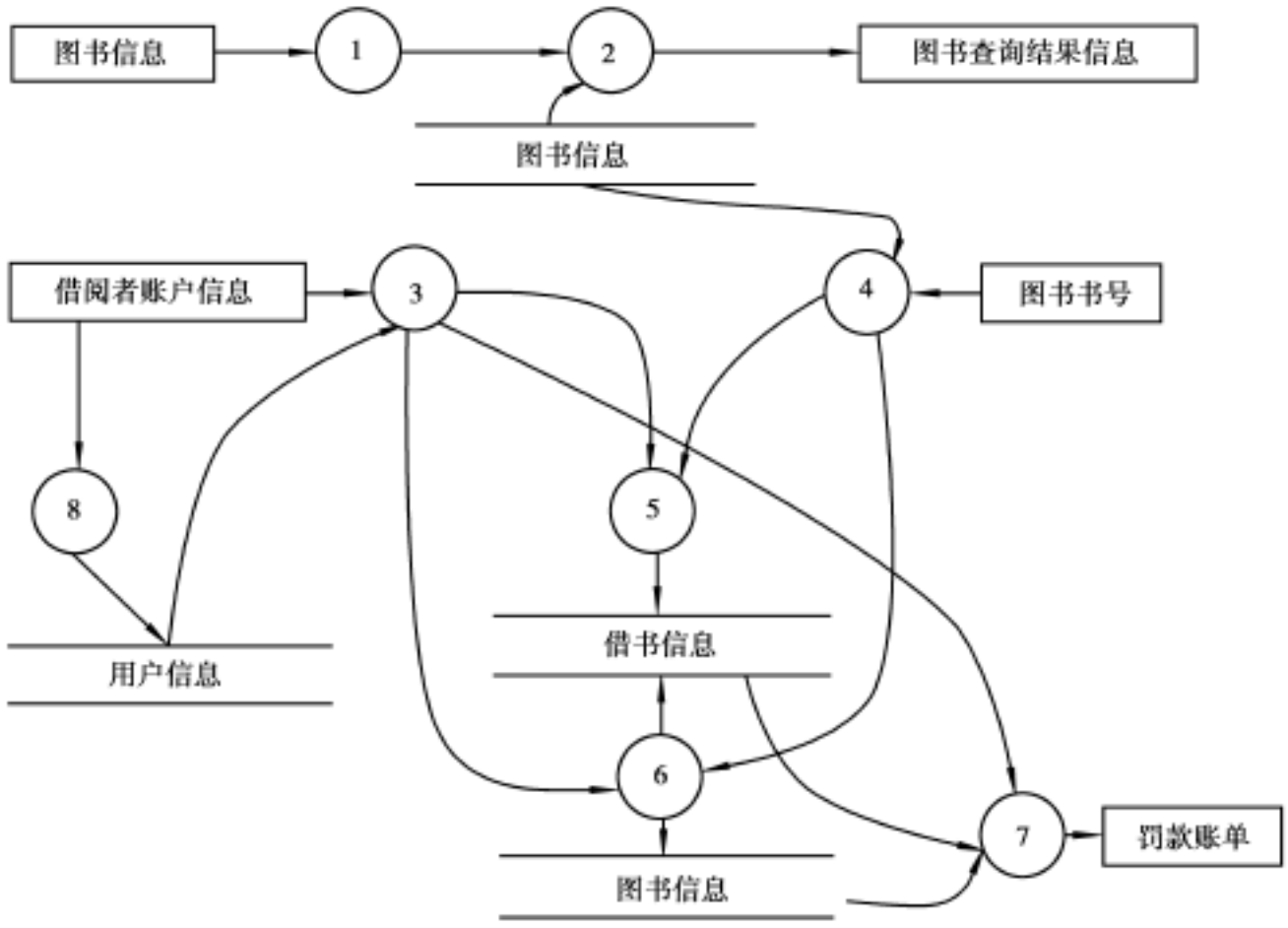


图 1.6　图书馆工作人员和管理人员处理模块功能结构图

下面对各模块(以数字表示) 进行功能说明:

a. 生成图书查询条件: 通过用户输入的查询条件信息, 形成计算机能执行的语句;

b. 检索图书信息: 由查询条件在数据库中检索出满足条件的图书信息;

c. 调出借阅者信息: 输入借阅者的账号, 从用户信息表中调出用户信息;

d. 检索图书信息: 由书号从图书信息表中检索出图书信息;

e. 生成借书账单: 由借阅者账号和图书书号生成借阅账单;

f. 生成还书账单: 由借阅者账号和图书书号生成还书账单;

g. 生成罚款账单: 由用户信息调出用户的借书状态, 并进行相应的罚款处理;

h. 维护用户信息: 维护各种用户的基本信息。

(2) 功能分配

各项模块的功能可参照 3.1 中的说明。客户机程序主要有三大块, 即: 接收数据、网络通信及输出部分。服务器程序主要具有三大功能, 即: 接收网络数据、数据库操作及发送网络数据。服务器程序需与已建立的 SQL SERVER 数据库互联, 其接口将于下面部分阐述。

4. 接口设计

(1) 外部接口

①用户界面

在用户界面部分, 根据需求分析的结果, 用户需要一个用户友善界面。在界面设计上, 应做到简单明了, 易于操作, 并且要注意到界面的布局, 应突出地显示重要以及出错信息。外观上也要做到合理化, 考虑到用户多对 WINDOW 风格较熟悉, 应尽量向这一方向靠拢。在设计语言上, 已决定使用 MS VISUAL C ++ 进行编程, 在界面上可使用 VISUAL C ++ 所提供的可视化组件, 向 WINDOWS 风格靠近。其中服务器程序界面要做到操作简单, 易于管理。在设计上采用下拉式菜单方式, 在出错显示上可调用 VISUAL C ++ 库中的错误提示函数。

总的来说, 系统的用户界面应具有可靠、简单、易学习和使用的特性。

②软件接口

服务器程序可使用 VISUAL C ++ 提供的对 SQL SERVER 的接口, 进行对数据库的所有访问。

服务器程序上可使用 SQL SERVER 的对数据库的备分命令, 以做到对数据的保存。

在网络软件接口方面, 使用一种无差错的传输协议, 采用滑动窗口方式对数据进行网络传输及接收。

③硬件接口

在输入方面, 对于键盘、鼠标的输入, 可用 VISUAL C ++ 的标准输入/输出, 对输入进行处理。

在输出方面, 打印机的连接及使用, 也可用 VISUAL C ++ 的标准输入/输出对其进行处理。

在网络传输部分, 在网络硬件部分, 为了实现高速传输, 将使用高速 ATM。

(2) 内部接口

内部接口方面, 各模块之间采用函数调用、参数传递、返回值的方式进行信息传递。具体参数的结构将在下面数据结构设计的内容中说明。接口传递的信息将是以数据结构封装了的数据, 以参数传递或返回值的形式在各模块间传输。

5. 数据结构设计

(1) 数据库数据结构设计

DBMS 的使用上系统将采用 SQL SERVER, 系统主要需要维护 4 张数据表。

其数据结构和物理结构如下:

数据项组名: 书籍信息								
特征	书号	书名	作者	出版社	出版日期	价格	页数	规格
数据类型	C	C	C	C	D	N	N	C
宽度	6	45	40	45	8	6	6	2
小数位	0	0	0	0	0	2	0	0
值约束	无	无	无	无	年月日	> 0	> 0	
空值否	否	否	否	否	否	是	是	是
值个数	1	1	1	1	1	1	1	1

数据项名: 用户信息						
特征	用户码	用户名	性别	密码	类型	备注
数据类型	C	C	C	C	C	C
宽度	7	40	1	10	8	50
小数位	0	0	0	0	0	0
值约束	无	无	男或女	无	学生或老师	
可空值否	否	否	是	是	是	是
值个数	1	1	1	1	1	1

数据项名: 借书信息					
特征	用户码	书号	拷贝号	借书日期	还书日期
数据类型	C	C	C	D	D
宽度	7	6	1	8	8
小数位	0	0	0	0	0
值约束	UUCODE	BBCODE	无	年月日	年月日
可空值否	否	否	否	否	否
值个数	1	1	1	1	1

数据项名: 状态表数据			
特征	书号	拷贝号	状态
数据类型	C	C	C
宽度	6	1	1
小数位	0	0	0
值约束	= BBCODE	无	在馆数目
空值否	否	否	否
值个数	1	1	1

(2)数据结构与程序的关系

服务器程序在对借书/还书进行操作时需对数据库数据结构,也就是数据表进行查询和修改:在借书/还书过程中都需要对数据库中的所有表,进行联合查询、修改。

物理数据结构主要用于各模块之间函数的信息传递。接口传递的信息将是以数据结构封装了的数据,以参数传递或返回值的形式在各模块间传输。出错信息将送入显示模块中,图书信息、图书库存信息和用户信息结构,各类报表结构,送入打印准备模块中准备打印。

6.运行设计

(1)运行模块的组合

客户机程序在有输入时启动接收数据模块,通过各模块之间的调用,读入并对输入进行格式化。在接收数据模块得到充分的数据时,将调用网络传输模块,将数据通过网络送到服务器,并等待接收服务器返回的信息。接收到返回信息后随即调用数据输出模块,对信息进行处理,产生相应的输出。

服务器程序的接收网络数据模块必须始终处于活动状态。接收到数据后,调用数据处理/查询模块对数据库进行访问,完成后调用网络发送模块,将信息返回客户机。

(2)运行控制

运行控制将严格按照各模块间函数调用关系来实现。在各事务中心模块中,需对运行控制进行正确的判断,选择正确的运行控制路径。

在网络传输方面,客户机在发送数据后,将等待服务器的确认收到信号,收到后,再次等待服务器发送回答数据,然后对数据进行确认。服务器在接到数据后发送确认信号,在对数据处理、访问数据库后,将返回信息送回客户机,并等待确认。

(3)运行时间

在软体的需求分析中,对运行时间的要求为必须对作出的操作有较快的反应。网络硬件对运行时间有最大的影响,当网络负载量大时,对操作反应将受到很大的影响。所以将采用高速局域网络,实现客户机与服务器之间的连接,以减少网络传输上的开销。其次是服务器的性能,这将影响对数据库访问时间即操作时间的长短,影响加大客户机操作的等待时间,所以必须使用高性能的服务器,建议使用 Pentium Ⅳ处理器。硬件对本系统的速度影响将会大于软件的影响。

7. 出错处理设计

(1)出错输出信息

程序在运行时主要会出现两种错误:a.由于输入信息,或无法满足要求时产生的错误,称为软错误。b.由于其他问题,如网络传输超时等,产生的问题,称为硬错误。

对于软错误,须在借书/还书操作成功判断及输入数据验证模块由数据进行数据分析,判断错误类型,再生成相应的错误提示语句,送到输出模块中。

对于硬错误,可在出错的相应模块中输出简单的出错语句,并将程序重置。返回输入阶段。出错信息必须给出相应的出错原因,例:“查无此图书信息”、“您输入的密码不正确”等。

(2)出错处理对策

所有的客户机及服务器都必须安装不间断电源以防止停电或电压不稳造成的数据丢失的损失。若真断电时,客户机上将不会有太大的影响,主要是服务器上:在断电后恢复过程可采用 SQL SERVER 的日志文件,对其进行 ROLLBACK 处理,对数据进行恢复。

在网络传输方面,可考虑建立一条成本较低的后备网络,以保证当主网络断路时数据的通信。

在硬件方面要选择较可靠、稳定的服务器机种,保证系统运行时的可靠性。

8.安全保密设计

由于数据的传输上需要通过网络传输,为了客户资料进行保密,需要在网络的传输过程中对数据进行加密。

这个工作主要是在准备网络包,及解开网络包这两个模块完成,它们各对数据进行加密及解密还原工作。

在加密算法选择上将使用 RSA 加密算法。具体算法可参照参考资料中《Computer Network》p. 598。

9.维护设计

维护方面主要为对服务器上的数据库数据进行维护。可使用 SQL SERVER 的数据库维护功能机制。例如,定期为数据库进行 Backup,维护管理数据库死锁问题和维护数据库内数据的一致性等。

实验 5　详细设计说明书

实验目的:由功能模块的功能需求,设计用于编程所需的程序逻辑结构图,掌握应用程序逻辑结构图的设计方法和要求,以及设计时要注意的事项。

实验要求:能够利用程序逻辑结构图的设计方法进行详细设计说明书的编写。

实验作业:编写一个详细设计说明书。

实验范例:

详细设计说明书

1.引言

(1)编写目的

在前一阶段中,已解决了实现该系统需求的程序模块设计问题。包括如何把该系统划分成若干个模块,决定各个模块之间的接口,模块之间传递的信息以及数据结构、模块结构的设计等。在以下的详细设计报告中将对在本阶段中对系统所做的所有详细设计进行说明。

在本阶段中,确定应该如何具体地实现所要求的系统,从而在编码阶段可以把这个描述直接翻译成用具体的程序语言书写的程序。主要的工作有:根据在《需求分析说明书》中所描述的数据、功能、运行、性能需求,并依照《概要设计说明书》所确定的处理流程、总体结构和模块外部设计,设计软件系统的结构设计、逐个模块的程序描述(包括各模块的功能、性能、输入、输出、算法、程序逻辑、接口等等),解决如何:

接受:图书信息和借阅者信息以及借阅者请求信息;

输出:借阅者借阅信息和催还通知单;

　　对图书信息的查询和修改;

在以下的各个阶段中,《用户操作手册》将与本阶段的工作紧密结合,努力做到让用户易

懂易学。《测试报告》和《维护报告》也将参考本说明书,检验本系统的各项性能指标,及时发现纰漏及时修补,一定要把功能强大、稳定可靠、便于维护的图书馆借阅管理系统交到用户手中。

(2)项目背景

本图书馆借阅管理系统项目主要由三部分形成:

图书信息查询和学生借阅情况查询前台客户模块;

图书馆工作人员操作模块;

图书管理员操作模块;

(3)文中特殊的定义和缩写

①定义

SQL SERVER: 系统服务器所使用的数据库管理系统(DBMS)。

SQL:一种用于访问查询数据库的语言

事务流:数据进入模块后可能有多种路径进行处理。

主键:数据库表中的关键域。值互不相同。

外部主键:数据库表中与其他表主键关联的域。

ROLLBACK:数据库的错误恢复机制。

②缩写

系统:若未特别指出,统指本图书借阅系统。

SQL: Structured Query Language(结构化查询语言)。

(4)参考资料

以下列出在概要设计过程中所使用到的有关资料:

图书馆借阅管理系统项目计划任务书	某某高校图书馆
图书馆借阅管理系统项目开发计划	KAVA 软件开发小组
需求规格说明书	KAVA 软件开发小组
概要设计说明书	KAVA 软件开发小组
用户操作手册(初稿)	KAVA 软件开发小组

2.总体设计

(1)需求概要

计算机进行信息管理与信息管理系统的开发密切相关,系统的开发是系统管理的前提。本系统就是为了管理好图书馆信息而设计的。图书馆作为一种信息资源的集散地,图书和用户借阅资料繁多,包含很多的信息数据的管理,现今,有很多的图书馆都是初步开始使用,甚至尚未使用计算机进行信息管理。根据调查得知,他们以前对信息管理的主要方式是基于文本、表格等纸介质的手工处理,对于图书借阅情况(如借书天数、超过限定借书时间的天数)的统计和核实等往往采用对借书卡的人工检查进行,对借阅者的借阅权限以及借阅天数等用人工计算、手抄进行。数据信息处理工作量大,容易出错;由于数据繁多,容易丢失,且不易查找。总的来说,缺乏系统,规范的信息管理手段。尽管有的图书馆有计算机,但是尚未用于信息管理,没有发挥它的效力,资源闲置比较突出,这就是管理信息系统的开发的基本环境。使图书管理工作规范化,系统化,程序化,避免图书管理的随意性,提高信息处理的速度和准确性,能够及时、准确、有效的查询和修改图书情况。

要求系统能有效、快速、安全、可靠和无误的完成上述操作。并要求客户机的界面要简单明了,易于操作,服务器程序利于维护。

(2)软件结构

各模块之间的关系已由概要设计给出。

①借阅者客户端模块结构图如图 1.7 所示

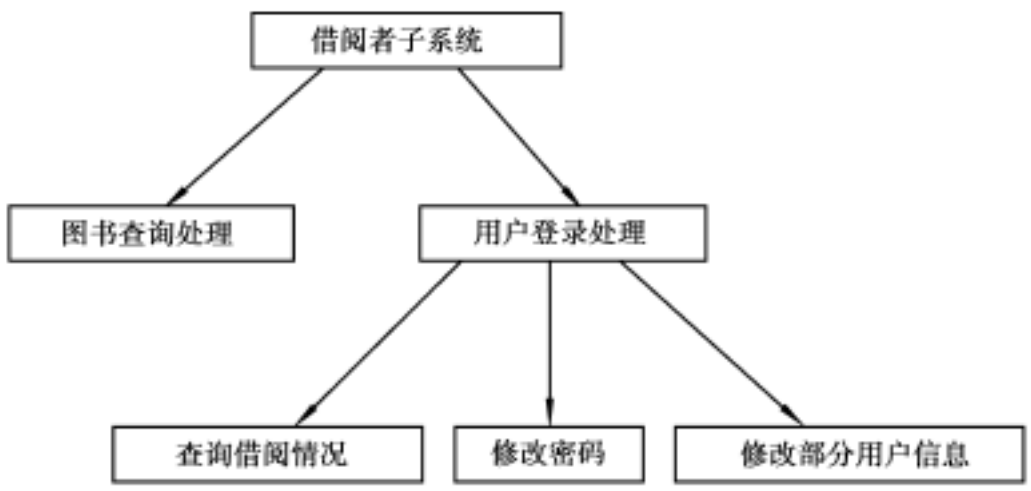


图 1.7　借阅者客户端模块结构图

②图书馆工作人员客户端模块结构图如图 1.8 所示

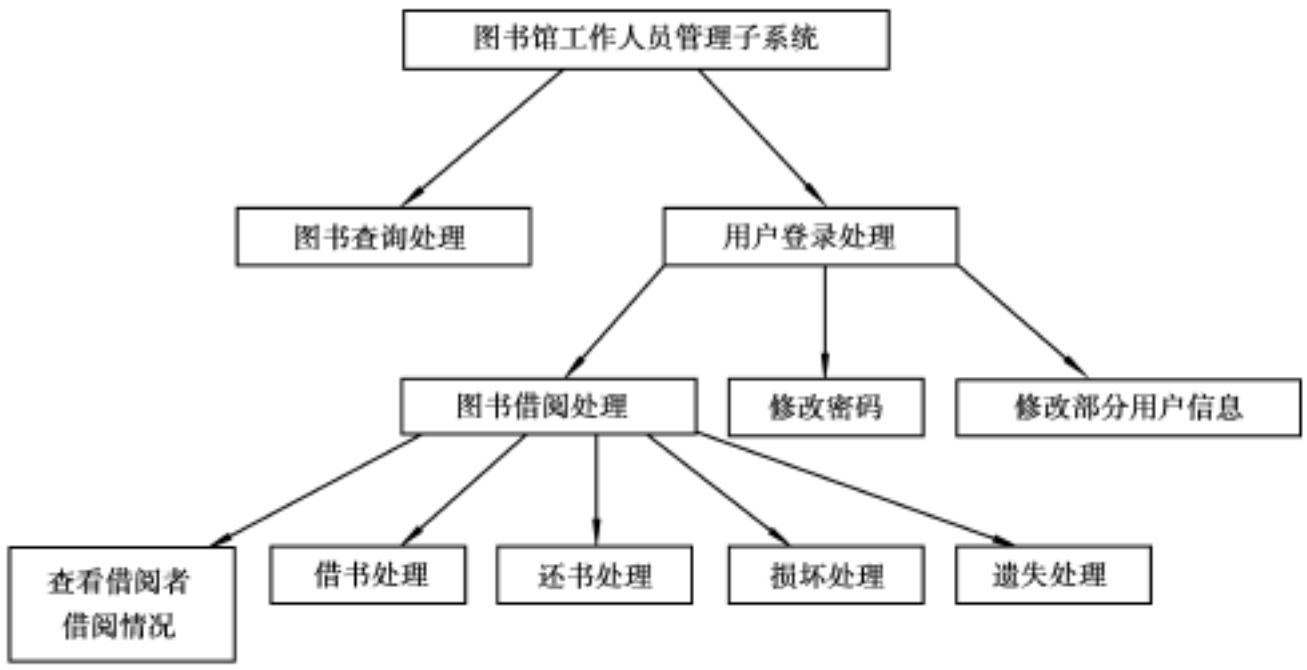


图 1.8　图书馆工作人员客户端模块结构图

③图书馆管理人员服务器端模块结构图如图 1.9 所示

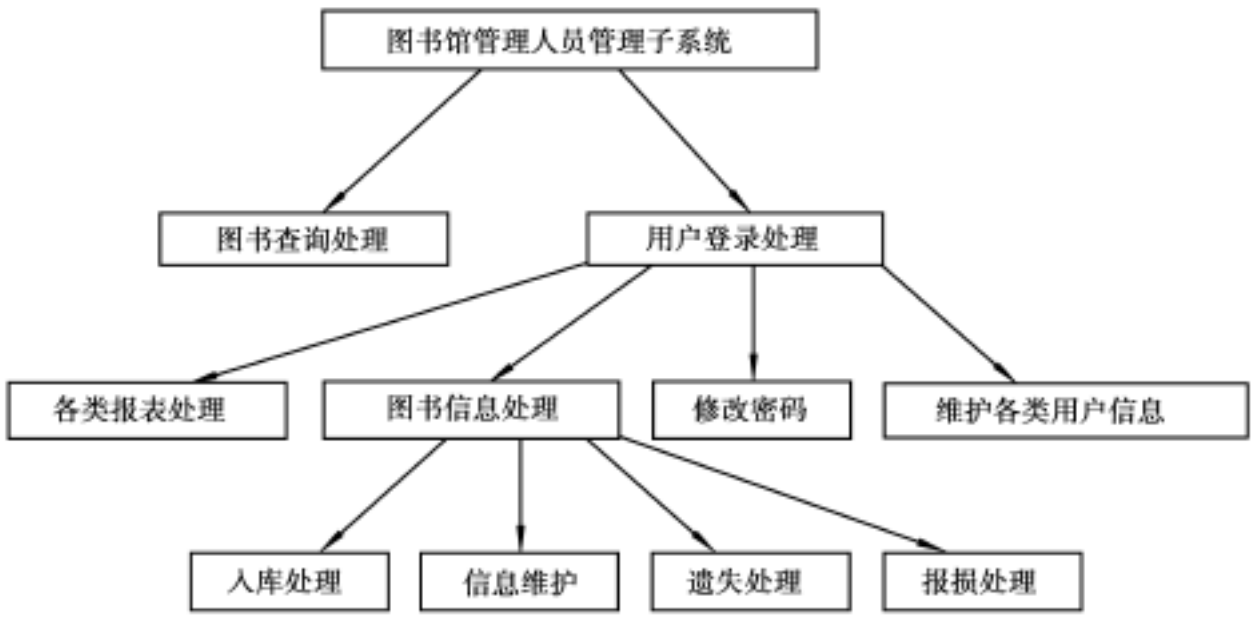


图 1.9　图书馆管理人员服务器端模块结构图

3. 程序描述(略)

(1) 借阅者客户端程序描述(略)

(2) 图书馆工作人员客户端程序描述(略)

(3) 图书馆管理人员端程序描述(略)

实验 6　用户操作手册

实验目的: 掌握如何编写用户操作使用手册。

实验要求: 能够根据软件的功能编写详细的用户操作使用手册。

实验作业: 编写一个用户操作手册。

实验范例:

用户操作手册

1. 引言

(1) 编写目的

本操作手册供本系统操作人员参考之用, 分别为学生、图书馆工作人员和图书馆管理人员说明本系统操作方法。

(2) 项目背景

本项目(图书馆借阅管理系统) 时由某某高校图书馆委托, 由《》软件开发小组负责开发。

(3) 定义

WINDOWS NT: 本系统所采用的操作系统。

SQL SERVER: 系统服务器所使用的数据库管理系统(DBMS) 。

(4) 参考资料

图书馆借阅管理系统项目计划任务书	某某高校图书馆
软件工程及其应用　周苏、王文等	天津科学技术出版社
软件工程　张海藩	清华大学出版社

项目的计划任务书	KAVA 软件开发小组
项目开发计划	KAVA 软件开发小组
需求规格说明书	KAVA 软件开发小组
概要设计说明书	KAVA 软件开发小组
详细设计说明书	KAVA 软件开发小组

2. 软件概述

(1) 目标

图书馆借阅管理系统的总目标是: 在计算机网络, 数据库和先进的开发平台上, 利用现有的软件, 配置一定的硬件, 开发一个具有开放体系结构的、易扩充的、易维护的、具有良好人机交互界面的图书借阅管理系统, 实现图书借阅管理的自动化和图书查询快速化的计算机系统, 为提高图书借阅的效率和学生查询图书的方便的信息管理系统。

(2) 功能

图书管理系统需要满足来自三方面的需求, 这三个方面分别是图书借阅者、图书馆工作人员和图书馆管理人员。图书借阅者的需求是查询图书馆所存的图书、个人借阅情况及个人信息的修改; 图书馆工作人员对图书借阅者的借阅及还书要求进行操作, 同时形成借书或还书报表给借阅者查看确认; 图书馆管理人员的功能最为复杂, 包括对工作人员、图书借阅者、图书进行管理和维护及系统状态的查看、维护并生成催还图书报表。

(3) 性能

数据精确度:

输入数据:

图书信息输入:

书号	字符;
书名	字符;
作者	字符;
出版社	字符;
出版日期	日期;
价格	数值;
页数	数值;
内容简介	字符;

用户信息输入:

用户码	字符;
用户名	字符;
性别	字符;
班级	字符;
密码	字符;
类型	字符;
备注	字符

输出数据:

图书状态信息:

书号　　字符;
拷贝号　　字符;
状态　　逻辑

用户借阅信息:

用户码　　字符;
书号　　字符;
拷贝号　　字符;
借书日期　　日期;
还书日期　　日期

时间特性:

要求数据传输时间在 0.5 秒钟以内, 服务器响应时间在 1 秒钟以内, 总响应时间为 3 秒钟。

3. 运行环境

(1) 硬件

服务器端: Pentium II 450 以上,　1 024M RAM,　36G HD

客户机端: Pentium 133 以上,　32M RAM,　2.1G HD

(2) 支持软件

系统软件: 服务器端: Windows NT Server

客户机端: Windows NT Workstation

数据库管理系统: SQL Server

4. 使用说明

(1) 安装和初始化

由于使用了安装自动生成工具, 安装变的非常简单, 只需运行光盘上的 SETUP.EXE 即可根据提示安装服务器端程序或是客户机端程序。

在安装的过程中, 系统将自动初始化, 为第一次运行作准备。

(2) 输入

数据背景

数据的来源参见需求分析说明书和概要设计说明书。

数据格式

图书信息输入:

书号　　字符;
书名　　字符;
作者　　字符;
出版社　　字符;
出版日期　　日期;
价格　　数值;
页数　　数值;
内容简介　　字符;

用户信息输入:

用户码	字符;
用户名	字符;
性别	字符;
班级	字符;
密码	字符;
类型	字符;
备注	字符

输入举例

图书信息输入:

书号	TP3001
书名	《软件工程》
作者	罗先文
出版社	重庆大学出版社
出版日期	2004.5
价格	21.5 元
页数	235
内容简介	软件工程的基础知识, 以及如何应用相应的软件工程知识开发软件。

用户信息输入:

用户码	20033101101
用户名	陈大建
性别	男
班级	03 计算机应用技术 1 班
密码	1234567
类型	学生
备注	无

(3) 输出

数据背景

数据的来源参见需求分析说明书和概要设计说明书。

数据格式

输出数据:

图书状态信息:

书号	字符;
拷贝号	字符;
状态	逻辑

用户借阅信息:

用户码	字符;
书号	字符;

拷贝号　　字符;
借书日期　　日期;
还书日期　　日期

输出举例

图书状态信息:

书号　　TP3001
拷贝号　　2
状态　　否

用户借阅信息:

用户码　　20033101101
书号　　TP3001
拷贝号　　2
借书日期　　2004-10-5
还书日期　　2004-11-5

(4) 出错和恢复

如果输入数据出错, 可立即进行取消借书和还书操作, 重新输入。

如果突然断电或系统没有响应, 则关机, 等系统重新启动时, 会根据日志文件自动 ROLL-BACK 到正确的阶段。需要等待一段时间。

(5) 求助查询

在任何时候, 按 F1 键, 即可获得详细的联机帮助。

5. 程序文件和数据文件一览表(略)

实验 7　测试计划

实验目的: 掌握如何编写制定测试计划。

实验要求: 制定软件测试计划。

实验作业: 根据范例编写制定一个软件测试计划。

实验范例:

测试计划

1.引言

(1) 编写目的

在开发大型软件的漫长过程中, 面对极其错综复杂的问题, 人的主观认识不可能完全符合客观现实, 与工程密切相关的各类人员之间的通信和配合也不可能完美无缺。因此, 在软件生命周期的每个阶段都不可避免地会产生差错。尤其对于机票预订系统这类会影响人们生活, 财产的工程软件, 必须尽量减少差错, 以免造成严重的损失。测试是“ 为了发现程序中的错误而执行程序的过程 ”。测试的目的就是在软件投入生产性运行之前, 尽可能多地发现软件中的错误。目前软件测试仍然是保证软件质量的关键步骤, 它是对软件规格说明, 设计和编码的

最后复审,也是必不可少的关键步骤。

(2)项目背景

本项目(图书馆借阅管理系统)时由某某高校图书馆委托,由软件开发小组负责开发。

(3)定义

SQL SERVER:系统服务器所使用的数据库管理系统(DBMS)。

SQL:一种用于访问查询数据库的语言

事务流:数据进入模块后可能有多种路径进行处理。

主键:数据库表中的关键域。值互不相同。

外部主键:数据库表中与其他表主键关联的域。

ROLLBACK:数据库的错误恢复机制。

(4)参考资料

图书馆借阅管理系统项目计划任务书	某某高校图书馆
《软件工程》	张海藩　清华大学出版社
项目的计划任务书	KAVA 软件开发小组
项目开发计划	KAVA 软件开发小组
需求规格说明书	KAVA 软件开发小组
概要设计说明书	KAVA 软件开发小组
详细设计说明书	KAVA 软件开发小组
用户操作手册	KAVA 软件开发小组

2.任务概述

(1)目标

测试是“为了发现程序中的错误而执行程序的过程”,测试的目的就是在软件投入生产性运行之前,尽可能多的发现软件中的错误。

(2)运行环境

由于系统将由三部分程序组成,安装在公共查询区的学生借阅信息与图书信息查询模块、每个图书室图书馆工作人员操作模块和图书馆管理人员端的服务管理模块。

①图书馆工作人员操作子系统和图书馆管理人员的服务管理子系统的运行要求:

系统软件:Window NT Server

数据库管理系统:SQL Server

硬件要求:Pentium II 450 以上,1 024M RAM,36G HD

②学生借阅信息与图书信息查询子系统的运行要求:

系统软件:Window NT Workstation

数据库管理系统:SQL Server

硬件要求:Pentium 133 以上,32M RAM,2.1G HD

(3)需求概述

计算机进行信息管理与信息管理系统的开发密切相关,系统的开发是系统管理的前提。本系统就是为了管理好图书馆信息而设计的。图书馆作为一种信息资源的集散地,图书和用户借阅资料繁多,包含很多的信息数据的管理,现今,有很多的图书馆都是初步开始使用,甚至尚未使用计算机进行信息管理。根据调查得知,他们以前对信息管理的主要方式是基于文本、

表格等纸介质的手工处理,对于图书借阅情况(如借书天数、超过限定借书时间的天数)的统计和核实等往往采用对借书卡的人工检查进行,对借阅者的借阅权限以及借阅天数等用人工计算、手抄进行。数据信息处理工作量大,容易出错;由于数据繁多,容易丢失,且不易查找。总的来说,缺乏系统,规范的信息管理手段。尽管有的图书馆有计算机,但是尚未用于信息管理,没有发挥它的效力,资源闲置比较突出,这就是管理信息系统的开发的基本环境。使图书管理工作规范化,系统化,程序化,避免图书管理的随意性,提高信息处理的速度和准确性,能够及时、准确、有效的查询和修改图书情况。

要求系统能有效、快速、安全、可靠和无误的完成上述操作。并要求客户机的界面要简单明了,易于操作,服务器程序利于维护。

(4)条件与限制

必须在保证各硬件设备、软件系统齐备的情况下,资金充足,人员齐备,各方面互相配合,齐心协力,共同完成。

3.计划

(1)测试方案

测试方案是测试阶段的关键技术问题。为了提高测试效率降低测试成本,本测试方案采用黑盒法设计基本的测试方案,再用白盒法补充一些方案。在黑盒法测试方案中,采用等价划分技术,把所有可能的输入数据(有效的和无效的)划分成几等价类,其划分类在以下的输入中再详述。

(2)测试项目

客户机接受信息模块测试

客户机输出信息模块测试

网络接受和发送模块结构测试

服务器模块(包括数据库)测试

各模块之间的接口测试

系统测试

(3)测试准备

在测试前,与各模块的主要负责人共同协商讨论,以概要设计说明书,详细设计说明书作为总的提纲,选择合适的输入输出数据,并加以意义列举说明。

(4)测试机构及人员

测试机构由信息学院学生测试工作组组成,人员有《》软件开发小组全体人员。

4.测试项目说明

(1)测试项目名称及测试内容

在测试过程中,首先需要对各子单元过程进行测试。在各子单元过程测试完毕后,再对各模块(包括各子单元过程之间的接口)进行测试,处理好各模块之间的接口,最后对系统进行测试和维护。

各子模块测试名称如下:

(2)测试用例

①输入

②输出

③步骤及操作

在测试过程中,首先需要对各子单元过程进行测试。各子单元过程的测试必须先在程序设计员调试并编译通过后才能进行。在各子单元过程测试完毕后,再对各模块(包括各子单元过程之间的接口)进行测试,处理好各模块之间的接口,最后对系统进行测试和维护。

其操作过程如下:

在客户机接受信息模块过程中,先对各子单元过程分别进行测试,然后根据白盒法按照详细设计说明书中的流程图对其进行跟踪测试。

同样,在客户机输出信息模块,网络接受和发送模块结构和服务器模块(包括数据库)过程中先对各子单元过程分别进行测试,然后根据白盒法按照详细设计说明书中的流程图对其进行跟踪测试。

然后,根据各模块之间的各种关系,对其接口进行测试。

在系统测试中,要注意对各种意外情况(例如断电、硬盘损坏等)加以处理,对数据库要注意其安全性、可靠性、健壮性、效率。网络传输更要注意其安全性。

(3)进度

由于其测试过程较长,需要对各子单元程序、各模块及它们之间的接口分别进行测试进度。一般测试过程都伴随其概要设计,详细设计过程一起进行,进度在 18 个月左右。

(4)条件

必须在保证各硬件设备、软件系统齐备的情况下,资金充足,人员齐备,各方面互相配合,齐心协力,共同完成。

(5)测试资料

测试资料主要是《》软件开发小组的各类文档及浙江航空公司公司提供的各类资料档案。

5. 评价

首先,我们要认识到测试是软件开发过程中一个非常重要的环节,一个好的软件必须经过无数次的测试。软件测试是保证软件质量的关键步骤。所以在测试过程中必须抱着不骄不躁,谦虚谨慎的态度,把好关!

实验 8　测试分析报告

实验目的:掌握如何制定测试分析报告。

实验要求:制定测试分析报告。

实验作业:根据范例制定一个详细测试分析报告。

实验范例:

测试分析报告

1. 引言

(1)编写目的

测试分析报告是在测试分析的基础上,对测试的结果以及测试的数据等加以记录和分析总结。它也是测试过程中的一个重要环节,同时,它也是对软件性能的一个总的分析和认可及

对不足之处的说明。因此,测试分析报告对于今后对软件的功能的加强,不足之处的弥补等都起着十分重要的提纲作用。另外,它还有利于今后软件开发者阅读原程序,根据测试提供的数据和结果,分析原代码,掌握各函数的功能和局限性。从而缩短软件开发者的再开发时间和所耗费的精力、资金。

从这方面上,我们不难发现,测试分析报告所指明的对象主要是针对于软件开发者。

(2)项目背景

此项目是由某某高校图书馆需求,要求开发一套实时,有效,完善,安全性好的软件系统。

(3)定义

SQL SERVER:系统服务器所使用的数据库管理系统(DBMS)。

SQL:一种用于访问查询数据库的语言。

事务流:数据进入模块后可能有多种路径进行处理。

主键:数据库表中的关键域。值互不相同。

外部主键:数据库表中与其他表主键关联的域。

ROLLBACK:数据库的错误恢复机制。

(4)参考资料

图书馆借阅管理系统项目计划任务书	某某高校图书馆
软件工程及其应用	周苏、王文等　天津科学技术出版社
软件工程	张海藩　清华大学出版社
项目的计划任务书	KAVA 软件开发小组
项目开发计划	KAVA 软件开发小组
需求规格说明书	KAVA 软件开发小组
概要设计说明书	KAVA 软件开发小组
详细设计说明书	KAVA 软件开发小组
用户操作手册	KAVA 软件开发小组
测试计划	KAVA 软件开发小组

2. 测试计划执行情况

测试项目:

客户机接受信息模块测试

客户机接受用户输入的各种数据(包括用户信息或借还账单,还包括一个借书/还书选项)然后经网络传送给服务器。

客户机输出信息模块测试

客户机输出为图书信息和确认借还书信息或出错信息。

网络接受和发送模块结构测试

接受由服务器程序经网络传送到客户机的数据包,它是程序与网络的接口。经解码后发送数据给服务器数据库。

服务器模块(包括数据库)测试

测试数据库的安全性、可靠性、健壮性、效率。

各模块之间的接口测试

对各模块之间的接口进行测试。

系统测试

用黑盒法对系统进行各类功能的测试。

测试机构和人员

测试机构——《》软件开发小组

人员主要由各程序模块的软件开发人员和航空公司的有关负责人共同主持。

测试结果

见测试计划。

软件需求测试结论

在经过对各模块的测试后,已经能够证实该软件各方面的能力都可以。只是在网络传输方面还有待加强。

3. 评价

软件能力

经测试证实该软件在各方面的综合能力都可以。

缺陷和限制

网络传输的保密性、安全性以及数据库的安全性还存在一定的缺陷。有可能会由于传输过程中数据的丢失造成软件运行的错误。

建议

建议在网络传输方面加强其保密性和安全性。

测试结论

该软件在各方面的综合能力都可以,通过!

实验 9　项目开发总结报告

实验目的:掌握如何编制项目开发总结报告。

实验要求:编制项目开发总结报告。

实验作业:编写一个项目开发总结报告。

实验范例:

总结性报告

1. 时间

这个项目从接受浙江航空公司的委托起,经过问题定义,可行性研究,需求分析,总体设计,详细设计,编码,综合测试,历时 107 个工作日。

2. 花费

设备支出:4.7 万

人员支出:15.4 万

3. 人员

系统分析员:张强。

程序员:张磊,张晓宇,林志,林柏纬,裘愉锋,吕乐明。

4. 遇到的困难

在完成问题定义,可行性研究,需求分析之后,由于用户的要求有了变化,所以进行了返工。在这一过程中,对原有文档和设计思想重新进行了改进。

实验 10　程序维护手册

实验目的:掌握如何编制程序维护手册。

实验要求:编制程序维护手册。

实验作业:编制一个项目的程序维护手册。

实验范例:

程序维护手册

1. 引言

(1)编写目的

软件维护是软件生命周期的最后一个阶段,它处于系统投入生产性运行以后的时期中,因此不属于系统开发过程。

软件维护需要的工作量非常大,虽然在不同应用领域维护成本差别很大,但是,平均说来,大型软件的维护成本高达开发成本的 4 倍左右。目前国外许多软件开发组织把 60% 以上的人力用于维护已有的软件,而且随着软件数量增多和使用寿命延长,这个百分比还在持续上升。

软件维护就是在软件已经交付使用之后,为了改正错误或者满足新的需要而修改软件的过程。

它有如下几种性质的维护:

①改正性维护

因为软件测试不可能暴露出一个大型软件系统中所有潜藏的错误,所以在使用期间,用户必然会发现程序错误,并且把他们遇到的问题报告给维护人员。我们把诊断和改正错误的过程称为改正性维护。

②适应性维护

计算机科学技术领域的各方面都在迅速进步,需要经常地修改版本。为了和变化了的环境适当地配合而进行的修改软件的活动称为适应性维护。

③完善性维护

在软件编写完成之后,投入实践,在使用软件的过程中,用户往往提出增加新功能或修改已有的功能的建议,这就需要进行完善性维护。

④预防性维护

为了改进未来的可维护性或可靠性,或为了给未来的改进奠定更好的基础而修改软件时,就需要进行预防性维护。

维护的过程本质上是修改和压缩了的软件定义和开发过程,而且事实上远在提出一项维护要求之前,与软件维护有关的工作已经开始了。

鉴于以上各点,编写维护软件的文档十分重要。它给软件维护人员提供了一份完整,清晰的说明文档,便于其快速有效地进行维护工作。

(2)开发单位

项目的提出者:某某高校图书馆

开发者:某某高校信息学院

用户:某某高校图书馆

使用场所:某某高校图书馆

(3)定义和缩写

①数据流图描绘系统的逻辑模型,图中没有任何具体的物理元素,只是描绘信息在系统中流动和处理的情况,它表示了数据和处理过程的关系。数据流图有 4 种基本符号:

正方形(或立方体)表示数据的源点或终点。

圆角矩形(或圆形)代表变换数据的处理。

处理不一定是一个程序。一个处理框可以代表一系列程序,单个程序或者程序的一个模块;它甚至可以代表一种人工处理过程。

开口矩形(或两条平行横线)代表数据存储。

数据存储可以表示一个文件,文件的一部分,数据库的元素或纪录的一部分等等。数据存储是处于静止状态的数据。

箭头代表数据流,即特定数据的流动方向。数据流是处于运动中的数据。

还有几种附加符号:

星号表示数据流之间是“与”关系

加号表示“或”关系

异或符号表示只能从中选一个

②数据字典(Data Dictionary,简称 DD)是对系统中各类数据描述的集合,是各类数据属性清单,是进行详细的数据收集和数据分析所获得的主要结果。它通常包括以下五个部分:

数据项:是数据的最小的单位。

数据结构:是若干数据项有意义的集合。

数据流:可以是数据项,也可以是数据结构,表示某一处理过程的输入或输出。

数据存储:处理过程中存取的数据。常常是手工凭证,手工文档,计算机文件,处理过程。

它们的描述内容如下:

数据项描述 = {数据项名,数据项含义说明,别名,类型,长度,取值范围,与其他数据项的逻辑关系}

取值范围,与其他数据项的逻辑关系定义了数据的完整性约束条件,是设计数据检验功能的依据。

数据结构描述 = {数据结构名,含义说明,组成:{数据结构或数据项}}

数据流 = {数据流名,说明,流出过程,流入过程,组成:{数据结构或数据项}}

流出过程,说明该数据流由什么过程来。

流入过程,说明该数据流到什么过程去。

数据存储 = {数据存储名,说明,输入数据流,输出数据流,组成:{数据结构或数据项},

数据量, 存取方式}

数据量, 说明每次存取多少数据, 每天(或每小时, 或每周) 存取几次的信息。

存取方法, 指的是批处理, 还是联机处理; 是检索还是更新; 是顺序检索还是随机检索; 尽可能详细收集并加以说明。

处理过程 = { 处理过程名, 说明, 输入: { 数据流}, 输出: { 数据流}, 处理: { 简要说明} }

简要说明中主要说明该处理过程的功能, 即" 做什么 "(不是怎么做) ; 处理频度要求, 如每小时(或每分钟) 处理多少事务, 多少数据量; 响应时间要求等。这些处理要求是后面物理设计的输入及性能评价的标准。

a. 主键: 数据库表中的关键域。值互不相同。

b. 外部主键: 数据库表中与其他表主键关联的域。

c. 系统: 若未特别指出, 统指本机票预定系统。

d. SQL: Structured Query Language(结构化查询语言) , 一种用于访问查询数据库的语言。

e. SQL SERVER: 系统服务器所使用的数据库管理系统(DBMS) 。

f. ATM: Asynchronous Transfer Mode (异步传输模式) 。

g. ROLLBACK: 数据库的错误恢复机制。

(4) 参考资料

书籍:

《软件工程导论》第三版　　张海藩　　清华大学出版社

《实用软件工程》第二版　　郑人杰　殷人昆　陶永雷　　清华大学出版社

文档:

需求规格说明书, 概要设计说明书, 详细设计说明书, 用户操作手册。

2. 系统说明

(1) 系统用途

输入: 用户信息和图书信息

输出: 图书信息和用户借阅信息

功能: 实现借阅者对图书的借书和还书处理, 以及查询各类图书的信息和库存状况。

(2) 安全保密

系统提供一定的方式让用户表示自己的身份, 系统进行核实, 通过鉴定后才提供机器使用权。常用的方法有: 用一个用户名或用户标识号来标识用户身份。

系统提供一个随机数, 用户根据预先约定好的某一过程或者函数进行计算, 系统根据用户计算结果是否正确进一步鉴定用户身份。

系统管理员还可对获得上机权的用户进行权限控制, 是不同的用户对于不同的数据对象有不同的操作权限。

(3) 总体说明

系统的总体功能: 系统对图书信息和用户信息的管理, 实现借阅者借书和还书处理, 以及对图书信息的查询, 并能够实现生成相关的报表。

系统的具体功能:

◆设计不同用户的操作权限和登陆方法

◆对所有用户开放的图书查询

◆借阅者维护借阅者个人部分信息
◆借阅者查看个人借阅情况信息
◆维护借阅者个人密码
◆根据借阅情况对数据库进行操作并生成报表
◆根据还书情况对数据库进行操作并生成报表
◆根据借阅者的对书的保护情况进行罚款处理并生成相应的报表
◆查询及统计各种借阅者和图书信息
◆维护图书信息
◆维护工作人员和管理员信息
◆维护借阅者信息
◆处理信息的完整性
◆对借阅过期的图书生成报表

(4) 程序说明(略)

3. 操作环境

设备: 共享一个数据库的若干台电脑, 台式打印机若干。

支持软件, 支持常用的数据库应用软件:

VISUAL FOXPRO 6.0 , DELPHI 6.0, POWER BUILDER 8.0

数据库

标识符: 姓名, 性别, 班级, 学号, 书号, 书名, 出版社。

静态数据: 存储在硬盘上的数据。

动态数据: 正处于处理过程中的数据。

数据库的存储媒体: 硬盘。

4. 维护过程

(1) 规则

①设计原则

a. 密切结合结构(数据) 设计和行为(处理) 设计。

b. 有机结合硬件, 软件, 技术和管理的界面。

c. 具体程序实现过程中, 对记录, 字段的引用参照相关信息。

d. 存储区的标识符也参照相关信息。

e. 在设计过程中参照瀑布模型, ER 模型, 层次图, Jackson 程序设计方法。

②设计程序变更的准则

检查可供选择的设计方案, 寻找一种与程序的原始设计原理相容的变更设计。

努力使设计简化。

能满足可变性要求的设计。

不降低程序质量。

用可测试的并具备测试方法的术语描述设计。

考虑处理时间, 存储量和操作过程方面的变化。

考虑变更对用户服务的干扰以及实施变更的代价与时间。

③修改程序代码的准则

必须要先熟悉整个程序的控制流程。

不要做不必要的修改。

不影响原始程序的风格和相容性。

记录所作过的修改。

审查软件质量是否符合标准。

更新程序文档以反映修改并保留修改前的程序代码版本。

④重新验证程序的准则

首先测试程序故障,然后测试程序的未改动部分,最后测试程序的修改部分。

不允许做修改的维护程序员成为惟一的重新验证程序的人。

鼓励终端用户参与到重新测试进程中来。

在重新验证进程中,记录出错的次数与类型,并把结果同所提供的测试功能进行比较,以便估量出程序是否退化。

(2)验证过程

每当软件被修改后,都要校验其正确性。维护员应该有选择地做些重新测试工作,不仅要证实新的逻辑的正确性,而且要校验实程序的为修改部分是否无损害,并且整个程序运行正确。若发现错误,则要马上进行修正。

出错及纠正方法

经查询还有库存,但输入借阅者信息后却发现已没有该书的库存了。发生这种情况的原因是:有多台计算机同时输入借阅同一本书的借阅者信息,在查询图书库存时,其他输入信息并未写入数据库,图书库存并未修改。此时,应该等待数秒后重新查询图书库存信息。

(3)专门维护过程

系统运行一段时间后,由于记录的不断增加,删除和修改,会使数据库的物理存储变坏。例如,逻辑上属于同一记录型或同一关系的数据被分散到了不同的文件或文件的多个碎片上。这样就会降低数据库存储空间的利用率和数据的访存效率,使数据库的性能下降。这时就要进行数据库的重组织。在重组过程中,按原设计要求重新安排记录的存储位置,调整数据区和溢出区,回收“垃圾”,减少指针链等。

5.程序清单及流程图(详见概要设计和详细设计文档)

第 2 部分
习题与参考答案

第 1 章　软件工程概述

1.1　主要内容

本章主要从软件的概念、特点和分类, 软件工程的概念, 软件的生存周期, 软件的开发模型, 软件工具及环境, 软件工程辅助工具, CASE 工具介绍等方面入手, 从整体上去了解软件工程的基本概念。

1.2　重点难点

了解软件的概念、特点和分类
了解发展和软件危机
了解软件工程的概念
了解软件生存周期
了解软件的开发模型
了解软件工具及环境

1.3　习题

1. 软件是计算机系统中与硬件相互依存的另一部分, 它是包括(A) 、(B) 及(C) 的完整集合。其中, (A) 是按事先设计的功能和性能要求执行的指令序列。(B) 是使程序能够正确操纵信息的数据结构。(C) 是与程序开发、维护和使用有关的图文材料。

供选择的答案:

A ~C. ①软件　②程序　③代码　④硬件
⑤文档　⑥外设　⑦数据　⑧图表

2. 开发软件时对提高软件开发人员工作效率至关重要的是(A) 。软件工程中描述生存周期的瀑布模型一般包括计划、(B) 、设计、编码、测试、维护等几个阶段, 其中设计阶段在管理上

又可以依次分成(C) 和(D) 两步。

供选择的答案:

A. ①程序开发环境 ②操作系统的资源管理功能
③程序人员数量 ④计算机的并行处理能力

B. ①需求分析 ②需求调查 ③可行性分析 ④问题定义

C、D. ①方案设计 ②代码设计 ③概要设计 ④数据设计
⑤运行设计 ⑥详细设计 ⑦故障处理设计 ⑧软件体系结构设计

3. 从供选择的答案中选出适当字句填入下列关于软件发展过程的叙述中的( ) 内。

有人将软件的发展过程划分为 4 个阶段:

第一阶段(1950—1950 年末) 称为“ 程序设计的原始时期 ”, 这时既没有(A) , 也没有(B) , 程序员只能用机器指令编写程序。

第二阶段(1950 年末—1960 年末) 称为“ 基本软件期 ”。出现了(A) , 并逐渐普及。随着(B) 的发展, 编译技术也有较大的发展。

第三阶段(1960 年末—1970 年中期) 称为“ 程序设计方法时代 ”。这一时期, 与硬件费用下降相反, 软件开发费急剧上升。人们提出了(C) 和(D) 等程序设计方法, 设法降低软件的开发费用。

第四阶段(1970 年中期—现在) 称为“ 软件工程时期 ”。软件开发技术不再仅仅是程序设计技术, 而是包括了与软件开发的各个阶段, 如(E) 、(F) 、编码、单元测试、综合测试、(G) 及其整体有关的各种管理技术。

供选择的答案:

A ~D: ①汇编语言 ②操作系统 ③虚拟存储器概念 ④高级语言
⑤结构式程序设计 ⑥数据库概念 ⑦固件 ⑧模块化程序设计

E ~G: ①使用和维护 ②兼容性的确认 ③完整性的确认
④设计 ⑤需求定义 ⑥图像处理

4. 软件工程过程有哪几个基本过程活动? 试说明之。

5. 试说明“ 软件生存周期 ”的概念。

6. 试论述瀑布模型软件开发方法的基本过程。

7. 软件工程是开发、运行、维护和修复软件的系统化方法, 它包含哪些要素? 试说明之。

8. 软件工程学的基本原则有哪些? 试说明之。

9. 有人说: 软件开发时, 一个错误发现得越晚, 为改正它所付出的代价就越大。对否? 请解释你的回答。

10. 软件产品的特性是什么?

11. 软件生产有几个阶段? 各有何特征?

12. 什么是软件危机? 产生原因是什么?

13. 什么是软件工程? 它目标和内容是什么?

14. 软件工程面临的问题是什么?

15. 什么是软件开发方法? 有哪些主要方法?

1.4　习题答案

1. A. ②,　B. ⑦,　C. ⑤

2. A. ①,　B. ①,　C. ③,　D. ⑥

3. A. ①,　B. ④,　C. ⑤,　D. ⑧,　E. ⑤,　F. ④,　G. ①

4. 软件工程过程的基本过程活动有 4 步:

P (Plan) : 软件规格说明。规定软件的功能及其运行的限制;

D (Do) : 软件开发。产生满足规格说明的软件;

C (Check) : 软件确认。确认软件能够完成客户提出的要求;

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。

5. 软件与任何一个事物一样, 有它的孕育、诞生、成长、成熟、衰亡的生存过程。这就是软件的生存周期。它主要分为 6 个阶段: 软件项目计划、软件需求分析和定义、软件设计、程序编码、软件测试, 以及运行维护。

(1) 软件项目计划: 在这一步要确定软件工作范围, 进行软件风险分析, 预计软件开发所需要的资源, 建立成本与进度的估算。根据有关成本与进度的限制分析项目的可行性。

(2) 软件需求分析和定义: 在这一步详细定义分配给软件的系统元素。可以用以下两种方式中的一种对需求进行分析和定义。一种是正式的信息域分析, 可用于建立信息流和信息结构的模型, 然后逐渐扩充这些模型成为软件的规格说明。另一种是软件原型化方法, 即建立软件原型, 并由用户进行评价, 从而确定软件需求。

(3) 软件设计: 软件的设计过程分两步走。第一步进行概要设计, 以结构设计和数据设计开始, 建立程序的模块结构, 定义接口并建立数据结构。此外, 要使用一些设计准则来判断软件的质量。第二步做详细设计, 考虑设计每一个模块部件的过程描述。经过评审后, 把每一个加细的过程性描述加到设计规格说明中去。

(4) 程序编码: 在设计完成之后, 用一种适当的程序设计语言或 CASE 工具生成源程序。应当就风格及清晰性对代码进行评审, 而且反过来应能直接追溯到详细设计描述。

(5) 软件测试: 单元测试检查每一单独的模块部件的功能和性能。组装测试提供了构造软件模块结构的手段, 同时测试其功能和接口。确认测试检查所有的需求是否都得到满足。在每一个测试步骤之后, 要进行调试, 以诊断和纠正软件的故障。

(6) 软件维护: 为改正错误, 适应环境变化及功能增强而进行的一系列修改活动。与软件维护相关联的那些任务依赖于所要实施的维护的类型。

6. 瀑布模型规定了各项软件工程活动, 包括: 制定软件项目计划, 进行需求分析和定义, 软件设计, 程序编码, 测试及运行维护。并且规定了它们自上而下, 相互衔接的固定次序, 如同瀑布流水, 逐级下落。然而软件开发的实践表明, 上述各项活动之间并非完全是自上而下, 呈线性图式。实际情况是, 每项开发活动均应具有以下特征:

(1) 从上一项活动接受本项活动的工作对象, 作为输入;

(2) 利用这一输入实施本项活动应完成的内容;

(3) 给出本项活动的工作成果, 作为输出传给下一项活动;

(4) 对本项活动实施的工作进行评审。若其工作得到确认, 则继续进行下一项活动, 否则返回前项, 甚至更前项的活动进行返工。

7. 软件工程包括三个要素: 方法、工具和过程。

软件工程方法为软件开发提供了“ 如何做 ”的技术。它包括了多方面的任务, 如项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法过程的设计、编码、测试以及维护等。软件工程方法常采用某一种特殊的语言或图形的表达方法及一套质量保证标准。

软件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。目前, 已经推出了许多软件工具, 已经能够支持上述的软件工程方法。特别地, 已经有人把诸多的软件工具集成起来, 使得一种工具产生的信息可以为其他的工具所使用, 这样建立起一种被称之为计算机辅助软件工程(CASE) 的软件开发支撑系统。CASE 将各种软件工具、开发机器和一个存放开发过程信息的工程数据库组合起来形成一个软件工程环境。

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。过程定义了方法使用的顺序、要求交付的文档资料、为保证质量和协调变化所需要的管理、及软件开发各个阶段完成的里程碑。

8. 在软件开发过程中必须遵循下列软件工程原则。

抽象: 采用分层次抽象, 自顶向下、逐层细化的办法进行功能分解和过程分解, 可以由抽象到具体、由复杂到简单, 逐步得到问题的解。

信息隐蔽: 遵循信息封装, 使用与实现分离的原则, 将模块设计成“ 黑箱 ”, 可以将实现的细节隐藏在模块内部, 使用者只能通过模块接口访问模块中封装的数据。

模块化: 按模块划分系统的体系结构, 使得各模块间有良好的接口。这样有助于信息隐蔽和抽象, 有助于表示复杂的系统。

局部化: 按抽象数据类型思想及问题域中的概念来建立模块, 确保模块之间低耦合, 模块内部高内聚。这有助于控制解的复杂性。

确定性: 软件开发过程中所有概念的表达应是确定的、无歧义性的、规范的。这有助于人们之间的沟通, 保证整个开发工作协调一致。

一致性: 强调软件开发过程的标准化、统一化。包括文档格式的一致, 工作流程的一致, 内、外部接口的一致, 系统规格说明与系统行为的一致等。

完备性: 软件系统不丢失任何重要成分, 可以完全实现系统所要求功能。

可验证性: 开发大型的软件系统需要对系统自顶向下、逐层分解。系统分解应遵循系统易于检查、测试、评审的原则, 以确保系统的正确性。

9. 软件开发时, 一个错误发现得越晚, 为改正它所付出的代价就越大。这个说法是对的。在 20 世纪 70 年代, GTE、TRW 和 IBM 等三家公司对此问题做了独立研究, 最后它们得到相似的结论:

阶　段	需求分析	软件设计	程序编码	单元测试	验收测试	维　护
相对修复代价	0.1 ~0.2	0.5	1	2	5	20

从表中可以看出, 在需求分析阶段检查和修复一个错误所需的代价只有编码阶段所需代价的 1/10 到 1/5, 而在维护阶段做同样的工作所付出的代价却是编码阶段的 20 倍。

10. 产品特性:

(1) 是一种逻辑产品, 与物质产品有很大的区别。

(2) 软件产品的生产主要是研制, 生产成本主要在开发和研制, 开发研制完成后, 通过复制就产生了大量软件产品。

(3) 软件产品不会用坏, 不存在磨损, 消耗。

(4) 生产主要是脑力劳动, 还未完全摆脱手工开发方式, 大部分产品是“ 定做 ”的。

(5) 开发软件的费用不断增加, 致使生产成本相当昂贵。

11. (1) 程序设计时代: 这个阶段生产方式是个体劳动, 使用的生产工具是机器语言, 汇编语言。

(2) 程序系统时代: 这个阶段生产方式是小集团合作生产, 使用的生产工具是高级语言, 开发方法仍依靠个人技巧, 但开始提出结构化方法。

(3) 软件工程时代: 这个阶段生产方式是工程化的生产, 使用数据库、开发工具、开发环境、网络、分布式、面向对象技术来开发软件。

12. 软件开发技术的进步未能满足发展的要求。在软件开发中遇到的问题找不到解决的办法, 问题积累起来, 形态尖锐的矛盾, 导致了软件危机。

产生原因: (1) 软件规模越来越大, 结构越来越复杂。

(2) 软件开发管理困难而复杂。

(3) 软件包开发费用不断增加。

(4) 软件开发技术落后。

(5) 生产方式落后, 仍采用手工方式。

(6) 开发工具落后, 生产率提高缓慢。

13. 软件工程就是用科学的知识程序和技术原理来定义, 开发, 维护软件的一门学科。

软件工程目标: 付出较低开发成本; 达到要求的功能; 取得较好的性能; 开发的软件易于移植; 只需较低的维护费用; 能按时完成开发任务, 及时交付使用; 开发的软件可靠性高。

软件工程内容: 研究内容包括开发技术和开发管理两个方面。

开发技术主要研究: 软件开发方法, 开发过程, 开发工具和环境。

开发管理主要研究: 软件管理学, 软件经济学, 软件心理学。

14. 软件工程需要解决的问题: 软件的费用, 可靠性, 可维护性, 软件生产率和软件的重用。

15. 使用早已定义好的技术集及符号表示习惯来组织软件生产的过程。通过使用成功的软件开发方法, 在规定的投资和时间内, 开发出符合用户需求的高质量的软件。软件开发方法是克服软件危机的重要方面之一, 对软件工程及软件包产业的发展起了不可估量的作用。主要有: 结构化方法, JACKSON 方法, 维也纳开发方法（ VDM ）, 面向对象开发方法。

第 2 章　可行性分析

2.1　主要内容

本章从软件开发的可行性研究的任务出发, 根据任务的要求设计可行性研究的步骤, 涉及到系统开发的流程, 再设计系统的开发进度, 从而进行成本效益分析, 设计出软件的计划说明书。

2.2 重点难点

了解可行性研究的任务
了解可行性研究的步骤
了解系统流程图(System Flow Diagram)
了解开发进度
了解成本/效益分析
了解软件计划说明书

2.3 习题

1. 可行性研究的任务是什么?
2. 可行性研究有哪些步骤?
3. 可行性研究报告有哪些内容?
4. 成本-效益分析可用哪些指标进行度量?
5. 项目开发计划有哪些内容?
6. 软件开发的早期,为什么要进行可行性研究?目标系统的可行性研究有几个方面?
7. 某航运公司为了方便旅客,拟开发一个船票预订系统。若将旅客的信息(姓名、性别、工作单位、身份证号、旅行时间、旅行目的地等) 输入该系统,则为旅客安排航班,打印出取票通知和票务账单。旅客可在航行的前一天凭取票通知和账单交款取票。系统校对无误即打印出船票给旅客。要求: (1) 提出问题定义; (2) 分析此系统的可行性; (3) 画出系统流程图和数据流图。

2.4 习题答案

1. a. 技术可行性:
考虑的因素: (1) 开发的风险; (2) 资源的有效性; (3) 技术; (4) 开发人员在主段技术可行性时,一旦估计错误,将会出现灾难性后果。
b. 经济可行性。
c. 社会可行性。
2. a. 确定项目规模和目标
b. 研究正在运行的系统
c. 建立新系统的高层逻辑模型
d. 导出和评价各种方案
e. 推荐可行的方案
f. 编写可行性研究报告
3. a. 引言
b. 可行性研究前提
c. 对现有系统的分析
d. 所建议系统的技术可行性分析
e. 所建议系统的经济可行性分析

f. 社会因素可行性分析

g. 其他可供选择方案

h. 结论意见

4. 有形效益度量:

a. 货币的时间价值

b. 投资回收期

c. 纯收入

无形效益: 主要从性质上、心理上进行衡量, 很难直接进行量的比较。通常以有形效益度量作为成本-效益分析的度量。

5. a. 项目概述

b. 实施计划

c. 人员组织及分工

d. 交付期限

6. 略

7. 略

第 3 章　软件需求分析

3.1　主要内容

本章主要讲解了系统需求分析的任务与步骤; 介绍了面向数据流的分析方法, 面向数据结构的分析方法, 原型化分析方法和系统动态分析方法。

3.2　重点难点

了解软件需求的目标和任务。

了解软件需求的获取方法。

了解可行性研究的方法和可行性研究报告的主要内容。

掌握结构化分析方法。

了解支持需求分析的原型化方法。

了解需求规格说明和需求评审的主要内容。

3.3　习题

1. 软件需求分析阶段的工作, 可以分为以下 4 个方面: 对问题的识别、分析与综合、编写需求分析文档以及(　　)。

A. 总结　　B. 阶段性报告　C. 需求分析评审　　D. 以上答案都不正确

2. 各种需求方法都有它们共同适用的(　　)。

A. 说明方法　B. 描述方式　　C. 准则　　D. 基本原则

3. 在结构化分析方法中, 用以表达系统内数据的运动情况的工具有(　　)。

A. 数据流图　B. 数据词典　C. 结构化英语　D. 判定表与判定树

4. 在结构化分析方法中用状态-迁移图表达系统或对象的行为。在状态-迁移图中, 由一个状态和一个事件所决定的下一状态可能会有(　　) 个。

A. 1　B. 2　C. 多个　D. 不确定

5. 在结构化分析方法中用实体-关系图表达系统中的对象及其关系。在实体-关系图中, 表达对象的实例之间的关联有三种类型: 一对一联系、(　　) 联系、多对多联系。

A. 多对一　B. 一对多

6. 软件需求分析的任务不应包括(A) 。进行需求分析可使用多种工具, 但(B) 是不适用的。在需求分析中, 分析员要从用户那里解决的最重要的问题是(C) 。需求规格说明书的内容不应当包括(D) 。该文档在软件开发中具有重要的作用, 但其作用不应当包括(E) 。供选择的答案:

A. ① 问题分析　②信息域分析　③结构化程序设计　④确定逻辑模型

B. ①数据流图　②判定表　③PAD 图　④数据词典

C. ①要让软件做什么　②要给该软件提供哪些信息
③要求软件工作效率如何　④要让软件具有什么样的结构

D. ①对重要功能的描述　②对算法的详细过程性描述
③软件确认准则　④软件的性能

E. ①软件设计的依据　②用户和开发人员对软件要“做什么”的共同理解
③软件验收的依据　④软件可行性分析的依据

7. 原型化方法是用户和软件开发人员之间进行的一种交互过程, 适用于(A) 系统。它从用户界面的开发入手, 首先形成(B) , 用户(C) , 并就(D) 提出意见, 它是一种(E) 型的设计过程。

供选择的答案:

A. ① 需求不确定性高的　②需求确定的
③管理信息　④决策支持

B. ①用户界面使用手册　②用户界面需求分析说明书
③系统界面原型　④完善的用户界面

C. ①改进用户界面的设计　②阅读文档资料
③模拟用户界面的运行　④运行用户界面原型

D. ①同意什么和不同意什么　②使用和不使用哪一种编程语言
③程序的结构　④执行速度是否满足要求

E. ①自外向内　②自顶向下　③自内向外　④自底向上

8. 在软件需求分析时, 首先建立当前系统的物理模型, 再根据物理模型建立当前系统的逻辑模型。试问: 什么是当前系统? 当前系统的物理模型与逻辑模型有什么差别?

9. 软件需求分析是软件工程过程中交换意见最频繁的步骤。为什么交换意见的途径会经常阻塞?

10. 你认为一个系统分析员的理想训练和基础知识是什么? 请说明理由。

11. 可行性研究主要研究哪些问题? 试说明之。

12. 信息和信息结构有什么区别? 有没有不存在信息流的系统? 有没有不存在信息结构

的系统?

13. 软件需求分析的操作性原则和需求工程的指导性原则是什么?

14. 数据流图的作用是什么? 它有哪些基本成分?

15. 数据词典的作用是什么? 它有哪些基本词条?

16. 传统的软件开发模型的缺陷是什么? 原型化方法的类型有哪些? 原型开发模型的主要优点是什么?

17. 试简述原型开发的过程和运用原型化方法的软件开发过程。

18. 软件需求分析说明书主要包括哪些内容?

19. 什么是需求分析? 需求分析阶段的基本任务是什么?

20. 什么是结构分析方法? 该方法使用什么描述工具?

21. 结构化分析方法通过哪些步骤来实现?

22. 画数据流图应注意什么事项?

23. 描述加工逻辑有哪些工具?

3.4　习题答案

1. C　　2. D　　3. A　　4. C　　5. B

6. A. ③　B. ③　C. ①　D. ②　E. ④

7. A. ①　B. ③　C. ④　D. ①　E. ①

8. 所谓当前系统可能是需要改进的某个已在计算机上运行的数据处理系统, 也可能是一个人工的数据处理过程。当前系统的物理模型客观地反映当前系统实际的工作情况。但在物理模型中有许多物理的因素, 随着分析工作的深入, 有些非本质的物理因素就成为不必要的负担, 因而需要对物理模型进行分析, 区分出本质的和非本质的因素, 去掉那些非本质的因素即可获得反映系统本质的逻辑模型。所以当前系统的逻辑模型是从当前系统的物理模型抽象出来的。

9. 软件需求分析过程中, 由于最初分析员对要解决的问题了解很少, 用户对问题的描述、对目标软件的要求也很凌乱、模糊, 再加上分析员和用户共同的知识领域不多, 导致相互间通信的需求。首先, 由于分析员和用户之间需要通信的内容相当多, 业务知识上的不足, 表达方式的不足, 可能对某些需求存在错误解释或误解的可能性, 造成需求的模糊性。其次, 用户和分析员之间经常存在无意识的“ 我们和他们 ”的界限, 不是按工作需要组成统一的精干的队伍, 而是各自定义自己的“ 版图 ”, 并通过一系列备忘录、正式的意见书、文档, 以及提问和回答来相互通信。历史已经证明, 这样会产生大量误解。忽略重要信息, 无法建立成功的工作关系。

10. 系统分析员处在用户和高级程序员之间, 负责沟通用户和开发人员的认识和见解, 起着桥梁的作用。一方面要协助用户对所开发的软件阐明要求, 另一方面还要与高级程序员交换意见, 探讨用户所提要求的合理性以及实现的可能性, 各人员之间关系见图 2.1 所示。最后还要负责编写软件需求规格说明和初步的用户手册。

为能胜任上述任务, 分析员应当具备如下的素质:

(1) 能够熟练地掌握计算机硬、软件的专业知识, 具有一定的系统开发经验。

(2) 善于进行抽象的思维和创造性的思维, 善于把握抽象的概念, 并把它们重新整理成为

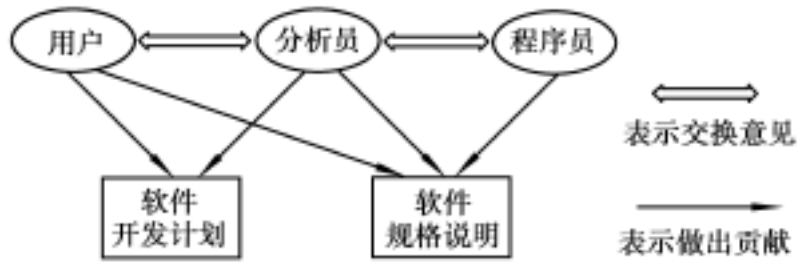


图 2.1　各人员之间关系图

各种逻辑成分,并给出简明、清晰的描述。

(3) 善于从相互冲突或混淆的原始资料中抽出恰当的条目来。

(4) 善于进行调查研究,能够很快学习用户的专业领域知识,理解用户的环境条件。

(5) 能够倾听他人的意见,注意发挥其他人员的作用。

(6) 具有良好的书面和口头交流表达能力。

11. 可行性研究主要做 4 个方面的研究:

经济可行性: 进行成本/效益分析。从经济角度判断系统开发是否“合算”。

技术可行性: 进行技术风险评价。从开发者的技术实力、以往工作基础、问题的复杂性等出发,判断系统开发在时间、费用等限制条件下成功的可能性。

法律可行性: 确定系统开发可能导致的任何侵权、妨碍和责任。

方案的选择: 评价系统或产品开发的几个可能的候选方案。最后给出结论意见。

12. 什么是信息? 广义地讲,信息就是消息。宇宙三要素(物质、能量、信息)之一。它是现实世界各种事物在人们头脑中的反映。此外,人们通过科学仪器能够认识到的也是信息。信息的特征为: 可识别、可存储、可变换、可处理、可传递、可再生、可压缩、可利用、可共享。我们通常讲的信息域就是对信息的多视角考虑。信息域包含 3 个不同的视图: 信息内容和关系、信息流和信息结构。为了完全理解信息域,必须了解每一个视图。

信息结构: 它是信息在计算机中的组织形式。一般表示了各种数据和控制对象的内部组织。数据和控制对象是被组织成 n 维表格,还是组织成有层次的树型结构? 在结构中信息与其他哪些信息相关? 所有信息是在一个信息结构中,还是在几个信息结构中? 一个结构中的信息与其他结构中的信息如何联系? 这些问题都由信息结构的分析来解决。

信息流: 表示数据和控制在系统中传递时的变化方式。输入对象首先被变换成中间信息(数据或控制),然后再变换成输出结果信息。沿着变换路径,可能从已有的数据存储(如磁盘文件或内存缓冲区)中引入附加的信息。对数据进行变换是程序中应有的功能或子功能。两个变换功能之间的数据传递就确定了功能间的接口。

所以,没有信息流的系统相当于没有功能的系统,这样的系统的存在是毫无意义的。而没有信息结构的系统是没有信息的系统,这样的系统不是计算机能够处理的系统。

13. 所有的需求分析方法都与一组操作性原则相关联:

必须理解和表示问题的信息域。

必须定义软件将完成的功能。

必须表示软件的行为(作为外部事件的结果)。

必须对描述信息、功能和行为的模型进行分解,能够以层次方式揭示其细节。

分析过程应当从要素信息转向细节的实现。

通过使用这些原则, 分析员可以系统地处理问题。首先检查信息域以便更完整地理解目标软件的功能, 再使用模型以简洁的方式表达目标软件的功能和行为, 并利用自顶向下、逐层分解的手段来降低问题的复杂性。在这些处理过程中, 因处理需求带来的逻辑约束和因其他系统元素带来的物理约束需要通过软件要素和视图的实现加以检验和确认。

除此以外, Davis 建议了一组针对“ 需求工程 ”的指导性原则:

在开始建立分析模型之前应当先理解问题。如果问题没有很好理解就急于求成, 常常会产生一个解决错误问题的完美的软件。

强力推荐使用原型。这样做可以使用户了解将如何与计算机交互, 而人们对软件质量的认识常常是基于对界面“ 友好性 ”的切身体会。

记录每一个需求的起源和原因。这是建立对用户要求的可追溯性的第一步。

使用多个视图, 建立系统的数据、功能和行为模型。这样做可帮助分析员从多方面分析和理解问题, 减少遗漏, 识别可能的不一致之处。

给需求赋予优先级。因为过短的时限会减少实现所有软件需求的可能性。因此, 对需求排定一个优先次序, 标识哪些需求先实现, 哪些需求后实现。

注意消除歧义性。因为大多数需求都是以自然语言描述, 存在叙述的歧义性问题, 造成遗漏和误解。采用正式的技术评审是发现和消除歧义性的好方法。

遵循以上原则, 就可能开发出较好的软件需求规格说明, 为软件设计奠定基础。

14. 数据流图可以用来抽象地表示系统或软件。它从信息传递和加工的角度, 以图形的方式刻画数据流从输入到输出的移动变换过程, 同时可以按自顶向下、逐步分解的方法表示内容不断增加的数据流和功能细节。因此, 数据流图既提供了功能建模的机制, 也提供了信息流建模的机制, 从而可以建立起系统或软件的功能模型。

数据流图的基本成分有 4 种, 如图 2.2 所示:

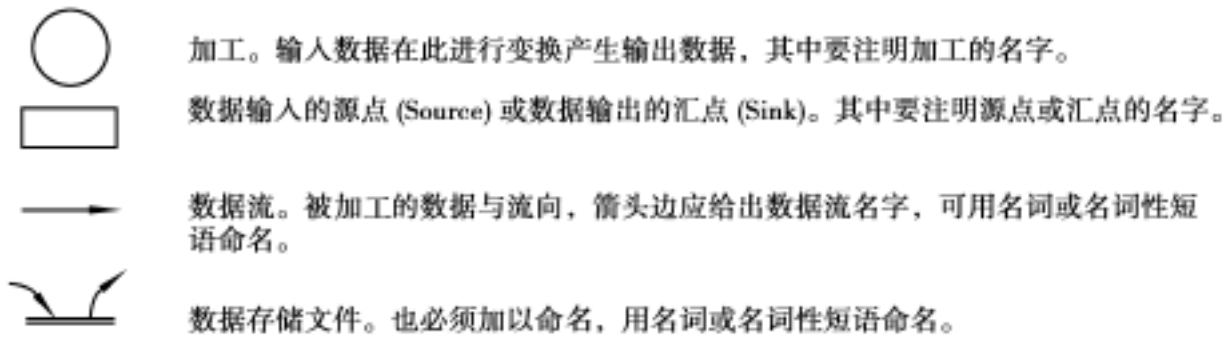


图 2.2　数据流图的基本成分

15. 分析模型中包含了对数据对象、功能和控制的表示。在每一种表示中, 数据对象和控制项都扮演一定的角色。为表示每个数据对象和控制项的特性, 建立了数据词典。数据词典精确地、严格地定义了每一个与系统相关的数据元素, 并以字典式顺序将它们组织起来, 使得用户和分析员对所有的输入、输出、存储成分和中间计算有共同的理解。

在数据词典的每一个词条中应包含以下信息:

(1) 名称: 数据对象或控制项、数据存储或外部实体的名字。

(2) 别名或编号。

(3) 分类: 数据对象? 加工? 数据流? 数据文件? 外部实体? 控制项(事件／状态) ?

(4) 描述: 描述内容或数据结构等。

(5) 何处使用: 使用该词条(数据或控制项) 的加工。

16. 传统软件生存期范型的典型代表是“ 瀑布模型 ”。这种模型的核心是将软件生存期划分为软件计划、需求分析、软件设计、编码、测试和运行维护等阶段, 根据不同阶段工作的特点, 运用不同的方法、技术和工具来完成该阶段的任务。软件开发人员遵循严格的规范, 在每一阶段工作结束时都要进行严格的阶段评审和确认, 以得到该阶段的一致、完整、正确和无歧义性的文档资料, 并以它们作为下一阶段工作的基础。

传统思想强调每一阶段的严格性, 尤其是开发初期要有良好的软件规格说明, 主要是源于过去软件开发的经验教训, 即在开发的后期或运行维护期间来修改不完善的规格说明要付出巨大的代价。但是, 要想得到一个完整准确的规格说明不是一件容易的事。特别是对于一些大型的软件项目, 在开发的早期用户往往对系统只有一个模糊的想法, 很难完全准确地表达对系统的全面要求, 软件开发人员对于所要解决的应用问题认识更是模糊不清。经过详细的讨论和分析, 也许能得到一份较好的规格说明, 但却很难期望该规格说明能将系统的各个方面都描述得完整、准确、一致, 并与实际环境相符。很难通过它在逻辑上推断出(不是在实际运行中判断评价) 系统运行的效果, 以此达到各方对系统的共同理解。随着开发工作向前推进, 用户可能会产生新的要求, 或因环境变化, 要求系统也能随之变化; 开发人员又可能在设计与实现的过程中遇到一些没有预料到的实际困难, 需要以改变需求来解脱困境。因此规格说明难以完善、需求的变更以及通信中的模糊和误解, 都会成为软件开发顺利推进的障碍。尽管在传统软件生存期管理中通过加强评审和确认, 全面测试, 甚至依靠维护阶段能够缓解上述问题, 但不能从根本上解决这些问题。

为了解决这些问题, 逐渐形成了软件系统的快速原型的概念。由于运用原型的目的和方式不同, 原型可分为以下两种不同的类型:

(1) 废弃型: 先构造一个功能简单而且质量要求不高的模型系统, 针对这个模型系统反复进行分析修改, 形成比较好的设计思想, 据此设计出更加完整、准确、一致、可靠的最终系统。系统构造完成后, 原来的模型系统就被废弃不用。

(2) 追加型或演化型: 先构造一个功能简单而且质量要求不高的模型系统, 作为最终系统的核心, 然后通过不断地扩充修改, 逐步追加新要求, 最后发展成为最终系统。

建立快速原型进行系统的分析和构造, 有以下的优点:

(1) 增进软件者和用户对系统服务需求的理解, 使比较含糊的具有不确定性的软件需求(主要是功能) 明确化。由于这种方法能在早期就明确了用户的要求, 因此可防止以后由于不能满足用户要求而造成的返工, 从而避免了不必要的经济损失, 缩短了开发周期。

(2) 软件原型化方法提供了一种有力的学习手段。通过原型演示, 用户可以亲身体验早期的开发过程, 获得关于计算机和被开发系统的专门知识。软件开发人员也可以获得用户对系统的确切要求, 学习到应用范围的专业知识。

(3) 使用原型化方法, 可以容易地确定系统的性能, 确认各项主要系统服务的可应用性, 确认系统设计的可行性, 确认系统作为产品的结果。因而它可以作为理解和确认软件需求规格说明的工具。

(4) 软件原型的最终版本, 有的可以原封不动地成为产品, 有的略加修改就可以成为最终系统的一个组成部分, 这样有利于建成最终系统。

17. 原型的开发和使用过程叫做原型生存期。下图是原型生存期的模型及其细化。

(1) 快速分析: 在分析者和用户的紧密配合下, 快速确定软件系统的基本要求。

(2) 构造原型: 根据基本规格说明, 尽快实现一个可运行的原型系统。

(3) 运行和评价原型: 用户试用原型, 考核评价原型的特性。纠正过去交互中的误解和分析中的错误, 增补新的要求, 提出全面的修改意见。

(4) 修正和改进: 根据修改意见进行修改。如果用修改原型的过程代替快速分析, 就形成了原型开发的迭代过程。在一次次的迭代过程中不断将原型完善, 以接近系统的最终要求。

(5) 判定原型完成: 经过修改或改进的原型, 达到参与者一致认可, 则原型开发的迭代过程可以结束。为此, 应判断有关应用的实质是否已经掌握, 判定的结果有两个不同的转向, 一是继续迭代验证, 一是进行详细说明。

(6) 判断原型细部是否说明: 判断组成原型的细部是否需要严格地加以说明。

(7) 原型细部的说明: 通过文件加以说明那些不能通过原型说明的项目。

(8) 判定原型效果: 考察新加入的需求信息和细部说明信息, 看其对模型有什么影响? 是否会影响模块的有效性? 如果模型受到影响, 则要进行修正和改进。

(9) 整理原型和提供文档:

快速原型方法的提出使得传统的软件生存期在思想方法上受到了影响。如果只是在局部运用原型化方法, 若将原型开发过程用于软件生存期的某一个阶段内, 那么传统的软件生存期依然不变, 只是阶段内部的软件定义或开发活动采用了新的方法。但若原型开发过程代替了传统生存期中的多个阶段, 则软件开发过程就成为一种新的形式。

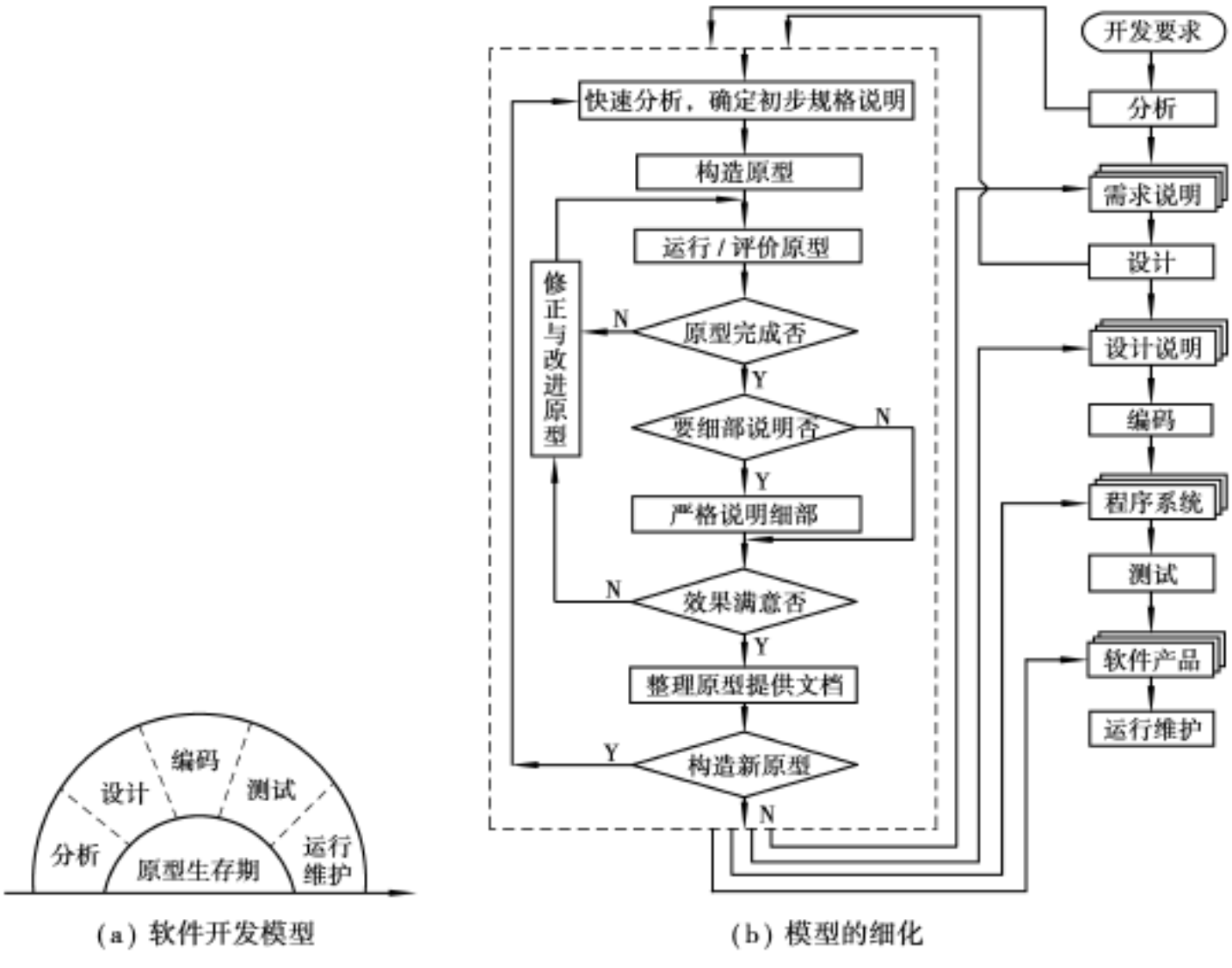


图 2.3　原型方法的软件生存期模型

图 2.3(a) 表示了使用原型方法的软件生存期模型。原型开发过程处于核心, 表示可在生存期的任何阶段中引入原型开发过程, 也可合并若干阶段, 用原型开发过程代替。图 2.3(b) 详细描述了在各个阶段可能引入原型开发过程的软件开发过程。其中, 在原型开发过程的最后加上了一个“ 是否构造新原型 ”的判断, 这是针对在系统开发的过程中有可能为不同的目的而要使用多个原型的情况而设。

(1) 辅助或代替分析阶段 : 在分析阶段利用快速原型方法可以得到良好的需求规格说明。在整体上仍然采用传统的模式, 但使用原型化方法来补充和完善需求说明以达到一致、准确、完整、无多义性地反映用户要求, 从而代替了传统的仅由复审和确认来提高需求规格说明质量的方法。并能在早期克服潜在的误解、遗漏和错误, 尽量不让潜在的问题遗留到开发的后期, 减少将来维护的代价。

(2) 辅助设计阶段: 在设计阶段引入原型, 可根据需求分析得到的规格说明进行快速分析, 得到实现方案后立即构造原型, 通过运行, 考察设计方案的可行性与合理性。在这个阶段引入原型, 可以迅速得到完善的设计规格说明。原型可能成为设计的总体框架, 也可能成为最终设计的一部分或补充的设计文档。

(3) 代替分析与设计阶段: 这时不再遵循传统的严格按阶段进行软件开发的要求, 而是把原型方法直接应用到软件开发的整体过程。在实施原型开发的过程中, 不再考虑完善的需求说明, 把分析、定义和设计交织在一起, 通过原型的构造、评价与改进的迭代过程, 逐步向最终系统的全面要求靠近。由于在分析的同时也考虑了设计与实现的要求, 能更有效地确定系统的需求和设计规格说明。

(4) 代替分析、设计和实现阶段: 在软件开发环境的支持下, 通过原型生存期的反复迭代, 直接得到软件的程序, 交付系统测试。这属于进化型的原型开发, 由初始的基本需求得到最初的原型开始, 一直进化到软件的整体系统, 并满足用户的一切可能的要求。

(5) 代替全部定义与开发阶段: 这是典型的进化型原型开发方法。完全摆脱了传统的软件生存期模式, 通过反复的原型迭代过程, 直接得到最终的软件产品。系统测试作为原型评价工作的一部分, 融入原型的开发过程。

18. 软件需求规格说明是分析任务的最终产物, 通过建立完整的信息描述、详细的功能和行为描述、性能需求和设计约束的说明、合适的验收标准, 给出对目标软件的各种需求。

软件需求规格说明的框架如下:

Ⅰ. 引言　A. 系统参考文献　B. 整体描述　C. 软件项目约束

Ⅱ. 信息描述　A. 信息内容表示　B. 信息流表示　ⅰ 数据流　ⅱ 控制流

Ⅲ. 功能描述　A. 功能划分　B. 功能描述　ⅰ 处理说明　ⅱ 限制/局限　ⅲ 性能需求　ⅳ 设计约束　ⅴ 支撑图　C. 控制描述　ⅰ 控制规格说明　ⅱ 设计约束

Ⅳ. 行为描述　A. 系统状态　B. 事件和响应

Ⅴ. 检验标准　A. 性能范围　B. 测试种类　C. 期望的软件响应 D. 特殊的考虑

Ⅵ. 参考书目

Ⅶ. 附录

19. 需求分析: 开发人员准确地理解用户的要求, 进行细致的调查分析, 将用户非形式的需求陈述转化为完整的需求定义, 再由需求定义转换到相应的需求规格说明的过程。

基本任务: ①问题识别: 双方确定对问题的综合需求, 这些需求包括功能需求, 性能需求,

环境需求,用户界面需求。②分析与综合,导出软件的逻辑模型。③编写文档:包括编写“需求规格说明书”,“初步用户使用手册”,“确认测试计划”,“修改完善软件开发计划”。

20. 结构化分析:简称 SA,面向数据流进行数据分析的方法。采用自顶向下逐层分解的分析策略。顶层抽象地描述整个系统,底层具体地画出系统工程的每个细节。中间层则是从抽象到具体的过渡。使用数据流图,数据字典,作为描述工具,使用结构化语言,判定表,判定树描述加工逻辑。

21. ①了解当前系统的工作流程,获得当前系统的物理模型。②抽象出当前系统的逻辑模型。③建立目标系统的逻辑模型。④作进一步补充和优化。

22. 命名:不能使用缺乏具体含义的名字,加工名应能反映出处理的功能。画数据流而不是控制流。数据流名称只能是名词或名词短语,整个图中不反映加工的执行顺序。一般不画物质流。

每个加工至少有一个输入数据流和一个输出数据流,反映出此加工数据的来源与加工的结果。

编号:某个加工分解成加一张数据流图时,上层图为父图,下层图为子图。子图应编号子图上的所有加工也应编号,子图的编号应与父图的编号相对应。

父图与子图的平衡:子图的输入输出数据流同父图相应加工的输入输出数据流必须一致。

局部数据存储:当某个数据流图中的数据存储不是父图中相应加工的外部接口,而只是本图中某些加工之间的数据接口,则称这些数据存储为局部数据存储。

注意数据流图的易理解性。

23. 加工逻辑的描述:一般来用结构化语言,判定表,判定树。

结构化语言是介于自然语言和形式语言之间的一种半形式化语言。它的结构可分里层和内层:

(1)外层:用来描述控制结构,采用顺序,选择,重复 3 种基本结构。

①顺序结构:是一组祈使语句,选择语句,重复语句的顺序排列。

②选择结构:一般用 IF-THEN-ELASE-ENDIF, CASE-OF-ENDCASE 等关键词。

③重复结构:一般用 DO-WHILE-ENDDO, REPEAT-UNTIL 等关键词。

(2)内层:一般采用祈使语句的自然语言短语,使用数据字典中的名词和有限的自定义词,其动词含义要具体,尽量不使用形容词和副词来修饰。

第 4 章　概要设计

4.1　主要内容

本章介绍了软件概要设计的原则和过程;程序的结构和结构图;软件设计的概念和原则;面向数据流的设计方法;面向数据结构的设计方法。

4.2 重点难点

了解软件概要设计的原则和过程。
掌握模块划分的评价准则——模块独立性的判别。
掌握结构化设计方法。
了解 Jackson 系统开发方法和 Jackson 程序设计方法。
了解数据设计和文件设计的原则。
掌握常用的详细设计的表达方法。

4.3 习题

1. 什么是软件概要设计？该阶段的基本任务是什么？
2. 软件设计的基本原理包括哪些内容？
3. 衡量模块独立性的两个标准是什么？它们各表示什么含义？
4. 模块间的耦合性有哪几种？它们各表示什么含义？
5. 模块的内聚性有哪几种？各表示什么含义？
6. 什么是软件结构？简述软件结构设计优化准则。
7. 什么是模块的影响范围？什么是模块的控制范围？它们之间应该建立什么关系？
8. 什么是"变换流"？什么是“事务流”？试将相应形式的数据流图转换成软件结构图。
9. 试述“变换分析”,“事务分析”的设计步骤。
10. 从下列有关系统结构图的叙述中选出正确的叙述。

(1) 系统结构图中反映的是程序中数据流的情况。

(2) 系统结构图是精确表达程序结构的图形表示法。因此，有时也可将系统结构当作程序流程图使用。

(3) 一个模块的多个下属模块在系统结构图中所处的左右位置是无关紧要的。

(4) 在系统结构图中，上级模块与其下属模块之间的调用关系用有向线段表示。这时，使用斜的线段和水平、垂直的线段具有相同的含义。

11. 软件的开发工作经过需求分析阶段，进入(A)以后，就开始着手解决“怎么做”的问题。常用的软件设计方法有(B)、(C)、(D)和(E)等方法。

供选择的答案:

A～B. ①程序设计　②设计阶段　③总体设计
④定义阶段　⑤ SD 方法　⑥ SP 方法

C. ① Jackson 方法　②瀑布法　③快速原型法　④回溯法

D～E. ① LCP(Wanier)方法　②递归法　③ Parnas 方法
④自下而上修正　⑤逐步求精法　⑥检测校正法

12. 请将下述有关模块独立性的各种模块之间的耦合，按其耦合度从低到高排列起来。

①内容耦合　②控制耦合　③非直接耦合　④标记耦合
⑤数据耦合　⑥外部耦合　⑦公共耦合

13. 请将下述有关模块独立性的各种模块内聚，按其内聚度(强度)从高到低排列起来。

①巧合内聚　②时间内聚　③功能内聚　④通信内聚

⑤逻辑内聚　　⑥信息内聚　　⑦过程内聚

14. 从供选择的答案中选出应该填入下列关于软件设计的叙述的(　　) 内的正确答案。

在完成软件概要设计, 并编写出相关文档之后, 应当组织对概要设计工作的评审。评审的内容包括:

分析该软件的系统结构、子系统结构, 确认该软件设计是否覆盖了所有已确定的软件需求, 软件每一成分是否可(A) 到某一项需求。分析软件各部分之间的联系, 确认该软件的内部接口与外部接口是否已经明确定义。模块是否满足(B) 和(C) 的要求。模块(D) 是否在其(E) 之内。

供选择的答案

A:	①覆盖	②演化	③追溯	④等同	⑤连接
B:	①多功能	②高内聚	③高耦合	④高效率	⑤可读性
C:	①多入口	②低内聚	③低耦合	④低复杂度	⑤低强度
D ~E:	①作用范围	②高内聚	③低内聚	④取值范围	⑤控制范围

15. 从供选择的答案中选出正确的答案填入下面的(　　) 中。

块间联系和块内联系是评价程序模块结构质量的重要标准。联系的方式、共用信息的作用、共用信息的数量和接口的(A) 等因素决定了块间联系的大小。在块内联系中, (B) 的块内联系最强。

SD 方法的总的原则是使每个模块执行(C) 功能, 模块间传送(D) 参数, 模块通过(E) 语句调用其他模块, 而且模块间传送的参数应尽量(F) 。

此外, SD 方法还提出了判定的作用范围和模块的控制范围等概念。SD 方法认为, (G) 应该是(H) 的子集。

供选择的答案:

A:	①友好性	②健壮性	③简单性	④安全性
B:	①巧合内聚	②功能内聚	③通信内聚	④信息内聚
C:	①一个	②多个		
D:	①数据型	②控制型	③混合型	
E:	①直接引用	②标准调用	③中断	④宏调用
F:	①少	②多		
G ~H:	①作用范围	②控制范围		

16. 从供选择的答案中选出正确的答案填入下列叙述中的(　　) 内。

模块内聚性用于衡量模块内部各成分之间彼此结合的紧密程度。

(1) 一组语句在程序中多处出现, 为了节省内存空间把这些语句放在一个模块中, 该模块的内聚性是(A) 的。

(2) 将几个逻辑上相似的成分放在同一个模块中, 通过模块入口处的一个判断决定执行哪一个功能。该模块的内聚性是(B) 的。

(3) 模块中所有成分引用共同的数据, 该模块的内聚性是(C) 的。

(4) 模块内的某成分的输出是另一些成分的输入, 该模块的内聚性是(D) 的。

(5) 模块中所有成分结合起来完全一项任务, 该模块的内聚性是(E) 的。它具有简明的外部界面, 由它构成的软件易于理解、测试和维护。

供选择的答案:

A～E: ①功能内聚　②信息内聚　③通信内聚　④过程内聚
⑤巧合内聚　⑥时间内聚　⑦逻辑内聚

4.4　习题答案

1. 软件概要设计: 在需求分析的基础上通过抽象和分解将系统分解成模块, 确定系统功能是实现, 即把软件需求转换为软件包表示的过程。

基本任务:

(1) 设计软件系统结构(简称软件结构)

a. 采用某种设计方法, 将一个复杂的系统按功能划分成模块(划分)

b. 确定模块的功能。　(功能)

c. 确定模块之间的调用关系。　(调用)

d. 确定模块之间的接口, 即模块之间传递的信息。　(接口)

e. 评价模块结构的质量。　(质量)

(2) 数据结构及数据库设计

a. 数据结构设计

b. 数据库设计: (概念设计、逻辑设计、物理设计)

(3) 编写概要设计文档(文档主要有: 概要设计说明书、数据库设计说明书、用户手册、修订测试计划)

(4) 评审

2. 软件设计的基本原理:

(1) 模块化(四个属性: 接口、功能、逻辑、状态)

(2) 抽象

(3) 信息隐蔽

(4) 模块独立性(两个定性的度量标准: 耦合性与内聚性)

3. 两个定性的度量标准: 耦合与内聚性耦合性: 也称块间联系。指软件系统结构中各模块间相互联系紧密程度的一种度量。模块之间联系越紧密, 其耦合性就越强, 模块的独立性则越差。

内聚性: 也称块内联系。指模块的功能强度的度量, 即一个模块内部各个元素彼此结合的紧密程度的度量。模块内元素联系越紧密, 内聚性越高。

4. 耦合性有 6 种: 无直接耦合、数据耦合、标记耦合、控制耦合、公共耦合、内容耦合、无直接耦合: 两个模块之间没有直接的关系, 它们分别从属于不同模块的控制与调用, 它们之间不传递任何信息。(无直接关系)

数据耦合: 指两个模块之间有调用关系, 传递的是简单的数据值, 相当于高级语言中的值传递。(数据值)

标记耦合: 指两个模块之间传递的是数据结构。(数据结构)

控制耦合: 指控制模块调用另一个模块时, 传递的是控制变量, 被调用块通过该控制变量的值有选择地执行块内某一功能。(控制变量)

公共耦合: 指通过一个公共数据环境相互作用的那些模块间的耦合。(一个公式数据环

境)

内容耦合: 一个模块直接使用另一个模块的内部数据, 或通过非正常入口而转入另一个模块内部。(内部数据)

5. 模块间的内聚性有 6 种: 偶然内聚、逻辑内聚、时间内聚、通信内聚、顺序内聚、功能内聚。

偶然内聚: 一个模块内的各处理元素之间没有任何联系。

逻辑内聚: 模块内执行几个逻辑上相似的功能, 通过参数确定该模块完成哪一个功能。

时间内聚: 把需要同时执行的动作组合在一起。

通信内聚: 指模块内所有处理元素都在同一个数据结构上操作, 或者指各处理使用相同的输入数据或产生相同的输出数据。

顺序内聚: 一个模块中各处理元素都密切相关于同一功能且必须顺序执行, 前一功能元素的输出是下一功能元素的输入。

功能内聚: 最强的内聚, 指模块内所有元素共同完成一个功能, 缺一不可。

6. 软件结构: 软件系统的模块层次结构, 反映了整个系统的功能实现, 即将来程序的控制体系。

软件结构设计优化准则:

a. 划分模块时, 尽量做到高内聚, 低耦合, 保持模块相对独立性, 以此为原则优化初始的软件结构。

b. 一个模块的作用范围应在其控制范围之内, 且判定所在的模块应与受其影响的模块在层次上尽量靠近

c. 软件的深度、宽度、扇入、扇出应适当。

d. 模块的大小要适中。

e. 模块的控制范围模块的接口要简单、清晰、含义明确, 便于理解, 易于实现、测试与维护。

7. 模块的影响范围: 受该模块内的一个判定影响的所有模块的集合。

模块的控制范围: 模块本身及其所有下属模块(直接或间接从属于它的模块) 的集合。

一个模块的影响范围应在其控制范围之内, 且判定所在的模块应与受其影响的模块在层次上尽量靠近。

8. 变换流由输入、变换(或处理) 、输出三部分组成。某个加工将它的输入流分离成许多发散的数据流, 形成许多加工路径, 并根据输入选择其中一个路径来执行这种特征的 DFD 称为事物流。

9. 变换分析:

a. 确定 DFD 中的变换中心。

b. 设计软件结构的顶层和第一层——变换结构。

c. 设计中下层模块。(输入模块下属模块的设计, 输出模块下属模块的设计, 变换模块下属模块的设计, 设计的优化)

事务分析:

a. 确定 DFD 中的事务中心和加工路径。

b. 设计软件结构的顶层和第一层——事务结构。(接收、发送给支)

c. 事务结构中、下层模块的设计、优化工作同变换结构。

10. (4)

11. A. ②, B. ⑤, C. ①, D. ①, E. ③。其中, D 与 E 的答案可互换。

12. ③、⑤、④、②、⑥、⑦、①

13. ③、⑥、④、⑦、②、⑤、①

14. A. ③, B. ②, C. ③, D. ①, E. ⑤

15. A. ③, B. ②, C. ①, D. ①, E. ②, F. ①, G. ①, H. ②

16. A. ⑤, B. ⑦, C. ③, D. ④, E. ①

第 5 章 详细设计

5.1 主要内容

本章介绍了系统详细设计的任务和原则, 并介绍了系统详细射击队描述工具, 以及介绍了详细设计的规格说明和评审方法。

5.2 重点难点

了解系统详细设计的任务和原则。

掌握系统详细设计的描述工具, 比如程序流程图、N-S 图、PAD 图、HIPO 图和过程设计语言 PDL。

了解详细设计规格说明与评审。

5.3 习题

1. 从供选择的答案中选出应该填入下列关于软件设计的叙述的( ) 内的正确答案。

在众多的设计方法中, SD 方法是最受人注意的, 也是最广泛应用的一种, 这种方法可以同分析阶段的(A) 方法及编程阶段的(B) 方法前后衔接, SD 方法是考虑如何建立一个结构良好的程序结构, 它提出了评价模块结构质量的两个具体标准——块间联系和块内联系。SD 方法的最终目标是(C) , 用于表示模块间调用关系的图叫(D) 。

另一种比较著名的设计方法是以信息隐蔽为原则划分模块, 这种方法叫(E) 方法。

供选择的答案:

A ~B: ①Jackson ②SA ③SC ④Parnas ⑤SP

C: ①块间联系大, 块内联系大 ②块间联系大, 块内联系小
③块间联系小, 块内联系大 ④块间联系小, 块内联系小

D: ①PAD ②HCP ③SC
④SADT ⑤HIPO ⑥ NS

E: ①Jackson ②Parnas ③Turing ④ Wirth ⑤ Dijkstra

2. 软件详细设计工具可分为三类, 即: 图示工具、设计语言和表格工具。图示工具中, (A) 简单而应用广泛; (B) 表示法中, 每一个处理过程用一个盒子表示, 盒子可以嵌套; (C) 可以纵横延伸, 图形的空间效果好; (D) 是一种设计和描述程序的语言, 它是一种面向(E) 的语言。

供选择的答案:

A ~C:	①NS 图	②流程图	③HIPO 图	④PAD 图
D:	①C	②PDL	③RPOLOG	④PASCAL
E:	①人	②机器	③数据结构	④对象

3. 举例说明你对概要设计与详细设计的理解。有不需要概要设计的情况吗?

4. 递归模块(即自己调用自己的模块) 的概念如何能够与本章所介绍的设计原理与方法相适应?

5. 如何用 PDL 语言来实施逐步求精的设计原理?

5.4　习题答案

1. A. ②,　B. ⑤,　C. ③,　D. ③,　E. ②

结构化设计方法(SD) 是一种应用非常广泛的软件设计方法, 它以结构化分析方法(SA) 得到的数据流图和数据词典为依据, 建立软件的模块结构, 然后对每一个模块用结构化程序设计(SP) 方法设计它的内部逻辑。这几种方法是前后衔接的。用 SD 方法建立的模块结构用模块间的耦合(块间联系) 和模块的内聚(块内联系) 来度量, 要求一个好的模块结构应满足高内聚, 低耦合。在 SD 方法中表示模块间调用关系的图叫做系统结构图(SC) 。

另一种著名的设计方法是以信息隐蔽为原则划分模块, 这种方法是 Parnas 提出来的, 叫做 Parnas 方法。

2. A. ②,　B. ①,　C. ④,　D. ②,　E. ①

3. 软件设计是一个把软件需求变换成软件表示的过程。最初这种表示只是描绘出软件的总的框架, 然后进一步细化, 在此框架中填入细节, 把它加工成在程序细节上非常接近于源程序的软件表示。正因为如此, 所以从工程管理的角度来看, 软件设计分两步完成。首先做概要设计, 将软件需求转化为数据结构和软件的系统结构。然后是详细设计, 即过程设计。通过对结构表示进行细化, 得到软件的详细的数据结构和算法。

由于概要设计建立起整个系统的体系结构框架, 并给出了系统中的全局数据结构和数据库接口, 人机接口, 与其他硬、软件的接口。此外还从系统全局的角度, 考虑处理方式、运行方式、容错方式以及系统维护等方面的问题, 并给出了度量和评价软件质量的方法, 所以它奠定了整个系统实现的基础。没有概要设计, 直接考虑程序设计, 就不能从全局把握软件系统的结构和质量, 实现活动处于一种无序状态, 程序结构划分不合理, 导致系统处于一种不稳定的状态, 稍一做改动就会失败。所以, 不能没有概要设计。

4. 递归过程在求解复杂的大型问题时非常有效。常用的求解问题的方法, 一种叫做“ 分而治之 ”的策略和“ 回溯 ”的策略, 都可以用递归方法来解决。所谓“ 分而治之 ”的方法即是把大而复杂的问题化为规模稍小的子问题, 用同样方法求解。如果分解后的子问题能够直接解决, 就直接解出, 然后再回推得到原来问题的解。所谓“ 回溯 ”方法就是如果一个大的问题在求解过程中从某一步出发有可选的多种解决方案, 先选择一种解决方案, 试探求解。如果求解失败, 撤销原来的选择, 再选一种解决方案, 试探求解, ……。如果用某一方案求解成功, 则退回上一步并报告这一步求解成功; 如果所有可选方案都试过, 都求解失败, 则退回上一步并报告求解失败。

软件设计过程中的“ 自顶向下, 逐层分解 ”的做法与上述求解问题的策略是一致的。如果

分解出来的子问题(子功能、子模块) 相互独立性比较强, 这种分解可以降低模块的复杂性, 做到模块化。所以, 只要分解出来的子问题与原来问题满足递归的情况, 用递归方法建立模块结构也是可行的。

5. 使用 PDL 语言, 可以做到逐步求精: 从比较概括和抽象的 PDL 程序起, 逐步写出更详细的更精确的描述。下面举一个例子。

```
PROCEDURE   spellcheck IS                          查找错拼的单词
    BEGIN
    split document into single words               把整个文档分离成单词
    lood up words in dictionary                    在字典中查这些单词
    display words which are not in dictionary      显示字典中查不到的单词
        create a new dictionary                    造一新字典
      END spellcheck
```

这个例子只是搭起一个处理问题的框架。为进一步表明查找拼错的单词的 4 个步骤如何实现, 可以对它每一步进行细化:

```
PROCEDURE spellcheck
    BEGIN
        - - *  split document into single words
        LOOP get next word
          add word to word list in sortorder
          EXIT WHEN all words processed
        END LOOP
        - - *  look up words in dictionary
        LOOP get word from word list
        IF word not in dictionary THEN
          - - *  display words not in dictionary
          display word, prompt on user terminal
          IF user response says word OK THEN
            add word to good word list
          ELSE
            add word to bad word list
          ENDIF
        ENDIF
        EXIT WHEN all words processed
    END LOOP
    - - *  create a new words dictionary
    dictionary: = merge dictionary and good word list
END spellcheck
```

第 6 章　程序编码

6.1　主要内容

本章主要通过对各种不同分类的软件以及相应的特点，如何去选取相应的程序设计语言；通过学习软件的编程风格，形成良好的编程风格；并掌握程序的编程效率和了解程序的复杂度的度量方法；程序的结构化程序设计方法的介绍。

6.2　重点难点

了解程序设计语言的分类、特点和如何选择程序设计语言。
掌握程序的编程风格，形成比较优秀的编程风格。
理解程序编程效率，提高自己的编程效率。
了解程序复杂度的度量方法，比如代码行度量法、McCabe 度量法、Halstead 的软件科学。
了解程序的结构化程序设计方法。

6.3　习题

1. 从下列关于模块化程序设计的叙述中选出 5 条正确的叙述。

①程序设计比较方便，但比较难以维护。

②便于由多个人分工编制大型程序。

③软件的功能便于扩充。

④程序易于理解，也便于排错。

⑤在主存储器能够容纳得下的前提下，应使模块尽可能大，以便减少模块的个数。

⑥模块之间的接口叫做数据文件。

⑦只要模块之间的接口关系不变，各模块内部实现细节的修改将不会影响别的模块。

⑧模块间的单向调用关系叫做模块的层次结构。

⑨模块越小，模块化的优点越明显。一般来说，模块的大小都在 10 行以下。

2. 结构化程序设计有时被错误地称为“无 GOTO 语句”的程序设计。请说明为什么会出现这样的说法，并讨论环绕着这个问题的一些争论。

3. 从下面关于程序编制的叙述中，选出三条正确的叙述。

①在编制程序之前，首先必须仔细阅读给定的程序说明书。然后，必须如实地依照说明书编写程序。说明书中常会有含糊不清或难以理解的地方。程序员在作业时应该对这些地方作出适当的解释。

②在着手编制程序时，重要的是采用既能使程序正确地按设计说明书进行处理，又易于出错的编写方法。

③在编制程序时，首先应该对程序的结构充分考虑，不要急于开始编码，而要像写软件文档那样，很好地琢磨程序具有什么样的功能，这些功能如何安排等等。

④考虑到以后的程序变更，为程序编写完整的说明书是一项很重要的工作。只要有了完

整的程序说明书,即使程序的编写形式难以让他人看懂也没有什么关系。

⑤编制程序时不可缺少的条件是,程序的输入和输出数据的格式都应确定。其他各项规定都是附带的,无足轻重。

⑥作为一个好的程序,不仅处理速度要快,而且易读易修改等等也都是重要的条件。为了能得到这样的程序,不仅要熟悉程序设计语言的语法,还要注意采用适当的规程和单纯的表现方法,注意使整个程序的结构简洁。

4.从下列叙述中选出 5 条符合程序设计风格指导原则的叙述。

① 嵌套的重数应加以限制。

② 尽量多使用临时变量。

③ 不滥用语言特色。

④ 不用可以省略的括号。

⑤ 使用有意义的变量名。

⑥ 应尽可能把程序编得短些。

⑦ 把常见的局部优化工作留给编译程序去做。

⑧ 注解越少越好。

⑨ 程序的格式应有助于读者理解程序。

⑩ 应尽可能多用 GOTO 语句。

5.从供选择的答案中选出应该填入下面(　　)中的正确答案。

A. 允许用户建立、修改、存储正文的计算机程序是(　　)。

①BOOtstrap　②Editor　③Loader　④Textformatter

B. 程序语言的编译系统和解释系统相比,从用户程序的运行效率来看(　　)。

①前者运行效率高　②两者大致相同

③后者运行效率高　④不能确定

C. FORTRAN 语言的源程序是(　　)结构。

①块状　②分程序嵌套

③既是块状,又是嵌套　④既不是块状,又不是嵌套的

D. 国际上最广泛使用的商用及行政管理语言是(　　)。

①COBOL　②BASIC　③FORTRAN　④PL/1

E. 国际上最流行的数值计算的程序设计语言是(　　)。

①BASIC　②ALGOL　③FORTRAN　④C

F. 美国国防部主持开发了高级程序设计语言 Ada,在它研制开始时,经反复比较,确定以高级语言(　　)作为 Ada 研究的出发点。

①LISP　②ALGOL　③ALGOL68　④PL/1

G. 在人工智能领域,目前最广泛使用的高级语言是(　　)。

①Ada　②FORTRAN　③COBOL　④LISP

6.从供选择的答案中选出应该填入下面(　　)中的正确答案。

A. 汇编程序是指(　　)。

①用汇编语言写的程序　②符号程序　③汇编语言的处理程序

B. 为了实现递归子程序的正确调用,人们必须用(　　)来保存(　　)及有关信息。

①堆栈　②线性表　③队列　④树
⑤入口点　⑥返回地址　⑦断点

C. UNIX 操作系统是（　　）研制的，它是用程序语言（　　）书写实现的。

①Bell 实验室　②DEC 公司　③IBM 公司　④ PASCAL
⑤并发 PASCAL　⑥MODULA　⑦ C

7. 下面给出一个求实函数方程 F(x) 在自变量区间 [a, b] 中的全部实根的算法。首先阅读此程序，然后

(1) 画出消去全部 goto 语句的结构化程序流程图。

(2) 将它改成 N_S 图。

(3) 计算该程序的 McCabe 复杂性度量。

在算法中，a 与 b 是区间[a, b] 的两端点值; eps1 与 eps2 是用户要求的求解精度。如果区间中点的函数值的绝对值小于 eps1 或新的小区间的长度小于 eps2，就认为这个中点为根。

```
float BinRoot ( float a, float b, float eps1, float eps2 )
  float low = a, high = b, mid, fmid;
   float flow = Func( low) , fhigh: = Func( high) ;
  label L1, L2, L3;                    //标号说明，给定某些程序地址
  if ( flow *  fhigh > 0.0 ) { BinRoot = 0;   goto L3; }        //无实根
L1:   mid = ( low + high) / 2;   fmid = Func( mid) ;
      if ( abs ( fmid ) <= eps1 ) {
L2:   BinRoot = mid; goto L3;
    }
    else if ( high - mid <= eps2 ) goto L2;
      else if ( flow *  fmid > 0.0 ) { low = mid; flow = fmid; goto L1; }
        else{ high = mid; goto L1 };
L3:
    }
```

8. 从供选择的答案中选出适当的字句填入下面关于程序生产率的描述中的（　　）内。

(1) 1960 年底 Dijkstra 提倡的 (A) 是一种有效的提高程序设计效率的方法。

(2) Dijkstra 为了使程序结构易于理解，把基本控制结构限于顺序、(B) 、(C) 3 种，应避免使用 (D) 。

(3) (A) 不仅提高程序设计的生产率，同时也容易进行程序的 (E) 。

供选择的答案:

A.　①标准化程序设计　② 模块化程序设计　③多道程序设计
　　④宏语言　⑤结构化程序设计　⑥汇编语言　⑦表格处理语言

B, C.　①分支　②选择　③重复　④计算　⑤输入输出

D.　①GOTO 语句　②DO 语句　③IF 语句　④REPEAT 语句

E.　①设计　②调试　③维护　④编码

9. 用某种软件复杂性度量算法来度量不同类型的程序时，得出的度量值是否真正反映了它们的复杂性? 如果对同类型的程序进行度量，其结果是否就比较有价值?

10. 软件复杂性有哪几类? 软件复杂性度量模型应遵循哪些基本原则?

6.4 习题答案

1. 正确的叙述有②、③、④、⑦、⑧。如果程序结构的模块化满足评价的标准(高内聚,低耦合),这样的结构是容易编码,容易测试,容易理解,容易修改,容易维护的。程序的功能也容易扩充。特别适合于大型程序编制时,多人分工合作,协同完成任务的情形。因为是采用自顶向下,逐层分解来划分模块结构的,所以模块之间的调用关系是分层次的模块结构,就叫做模块的层次结构。模块之间的信息传递叫做模块的接口,模块之间传递信息可以通过参数表、全局变量或全局数据结构、数据文件、专门的通信模块,不是专指数据文件。划分模块时,模块大小要适中。模块太大,控制路径数目多、涉及的范围广、变量的数目多、总体复杂性高,可理解性、可修改性、可靠性就会变差。模块太小,模块个数增大,调用的系统开销就会增大。所以要有一个权衡。

2. 早在 1963 年,针对当时流行的 ALGOL 语言,Peter Naur 指出,在程序中大量地,没有节制地使用 GOTO 语句,会使程序结构变得非常混乱。但是很多人还不太注意这一问题。以致许多人写出来的程序仍然是纷乱如麻的。

1965 年,E. W. Dijkstra 在一次会议上提出,应当把 GOTO 语句从高级语言中取消。并指出,程序的质量与程序中包含的 GOTO 语句的数量成反比。在这种思想的影响下,当时新开发的几种高级程序设计语言,例如 LISP、ISWIM、BLISS 等,都把 GOTO 语句取消了。

1966 年,Bohm 与 Jacopini 证明了任何单入口单出口的没有“死循环”的程序都能由三种最基本的控制结构构造出来。这三种基本控制结构就是“顺序结构”、“选择 IF—THEN—ELSE 结构”、“重复 DO—WHILE 或 DO—UNTIL 结构”。

1968 年,Dijkstra 在写给 < ACM > (美国计算机协会通讯) 杂志编辑部的信中再次建议从一切高级语言中取消 GOTO 语句,只使用三种基本控制结构编写程序。他的建议引起了激烈的争论。争论集中在如何看待 GOTO 语句的问题上。赞成取消 GOTO 语句的一方认为,GOTO 语句对程序清晰性有很大破坏作用,凡是使用 GOTO 语句多的程序,其控制流时而 GOTO 向前,时而 GOTO 向后,常使程序变得很难理解,从而增加查错和维护的困难,降低程序的可维护性。但以 D. E. Knuth 为代表的另一方认为,GOTO 语句虽然存在着破坏程序清晰性的问题,但不应完全禁止。因为 GOTO 语句概念简单,使用方便,在某些情况下,保留 GOTO 语句反能使写出的程序更加简洁,并且 GOTO 语句可直接得到硬件指令的支持。经过争论,人们认识到,不是简单地去掉 GOTO 语句的问题,而是要创立一种新的程序设计思想、方法和风格,以显著提高软件生产率和软件质量,降低软件维护的成本。

20 世纪 70 年代初 N. Wirth 在设计 Pascal 语言时对 GOTO 语句的处理可被当作对 GOTO 语句争论的结论。在 Pascal 语言中设置了支持上述三种基本控制结构的语句;另一方面,GOTO 语句仍然保留在该语言中。不过,N. Wirth 解释说,通常使用所提供的几种基本控制结构已经足够,习惯于这样做的人不会感到 GOTO 语句的必要。也就是说,在一般情况下,可以完全不使用 GOTO 语句。如果在特殊情况下,由于特定的要求,偶然使用 GOTO 语句能解决问题,那也未尝不可,只是不应大量使用罢了。

事实上,大量采用 GOTO 语句实现控制路径,会使程序路径变得复杂而且混乱,从而使程序变得不易阅读,给程序的测试和维护造成困难,还会增加出错的机会,降低程序的可靠性。

因此要控制 GOTO 语句的使用。但有时完全不用 GOTO 语句进行程序编码,比用 GOTO 语句编出的程序可读性差。例如,在查找结束时,文件访问结束时,出现错误情况要从循环中转出时,使用布尔变量和条件结构来实现就不如用 GOTO 语句来得简洁易懂。

3. ①、④、⑥。编制程序的过程实际上是根据设计的结果,用某种机器能够识别的程序设计语言,将设计翻译成机器代码的过程。因此,必须如实地按照设计说明书编写程序。至于设计说明书中含糊不清的地方,应当在编程时与分析人员或设计人员协商,对这些地方做出适当的解释。另外,考虑到将来的程序修改,必须为程序编写完整的说明书,同时程序必须编写得容易让别人看得懂,这样程序才容易修改,修改时不容易出错,而且容易验证修改后的结果。还有,编写程序的人不需重新考虑程序要完成什么功能,这些已经在软件分析与设计过程中充分考虑过了。

4. ①、③、⑤、⑦、⑨是正确的。因为①条件语句和循环语句嵌套得过多会增加程序的复杂性,从而增加程序的出错率。③虽然国际以至国内已经发表了编程语言的标准,但各个计算机厂商在推出自己的计算机系统的同时,也推出了针对自己机器特色的程序设计语言的非标准版本,如果利用这些语言的非标准特性编写程序,就会给将来程序的移植带来困难。为了提高程序的可移植性,应当只使用语言的标准版本,不要滥用语言的非标准特色。⑤给在程序中使用的变量赋予与实际含义相符的名字,可以提高程序的可读性,从而提高程序的可维护性。⑦程序优化的工作最好交给编译程序来做,程序员应把主要注意力放在提高程序的可读性、清晰性、简洁性、正确性、一致性等方面,从而保证软件的可靠性和可维护性。⑨程序的可读性是至关重要的。所以程序的格式应有助于读者理解程序。

5. A. ②　B. ①　C. ①　D. ①　E. ③　F. ②　G. ④

除以上几种论述外,其他的叙述都不对。例如,程序中加入临时变量,可能会改变程序执行中的时序关系,造成程序出错。在表达式中加入括号,可以明确标明表达式的运算优先关系,避免因语言方面的原因可能潜藏的错误。程序模块的大小要适中,不是编得越短越好。注解加多少,由问题得难度来决定,但绝不是可有可无的。最后要限制 GOTO 语句的使用,因为它可能会造成思路混乱、极易出错。

A. 计算机用户通常是使用“编辑程序(Editor) ”对源程序文本进行录入、编辑、存储的,不用自举程序(Bootstrap) 、连接程序(Loader) 或文本格式化程序(Textformatter) 。

B. 解释系统是边解释源程序边执行该源程序,编译程序是先编译出源程序的对应目标代码,再执行这些目标代码。所以编译程序编出的目标代码运行效率高。

C. FORTRAN 程序是以 SUBROUTINE 为单元的块状结构,对每一个 SUBROUTINE 进行编译后通过连接形成整个程序系统。它不是嵌套的。

D. 国际上最流行的商业和行政管理语言是 COBOL 语言。

E. 国际上最流行的用于数值计算的语言是 FORTRAN 语言。

F. 美国国防部主持开发高级程序设计语言 Ada 时,曾确定以 ALGOL 语言作为 Ada 研究的出发点。所以, Ada、ALGOL、Pascal、BASIC 和 C 都是 ALGOL 系的一些程序语言。

G. 在人工智能领域,目前最广泛使用的高级语言是 Lisp。

6. A. ③　B. ①⑥　C. ①⑦

A. 汇编程序实际是指汇编语言的处理程序。而用汇编语言写成的源程序一般称为汇编语言程序。

B. 为了实现递归子程序的正确调用,一般使用堆栈来保存每次调用后返回到上一层程序的返回地址、本次递归调用时的形式参数、局部变量等。

C. UNIX 操作系统是 Bell 实验室研制的,用 C 语言写出来的。

7.(1) 结构化的程序流程图如图 2.4 所示:

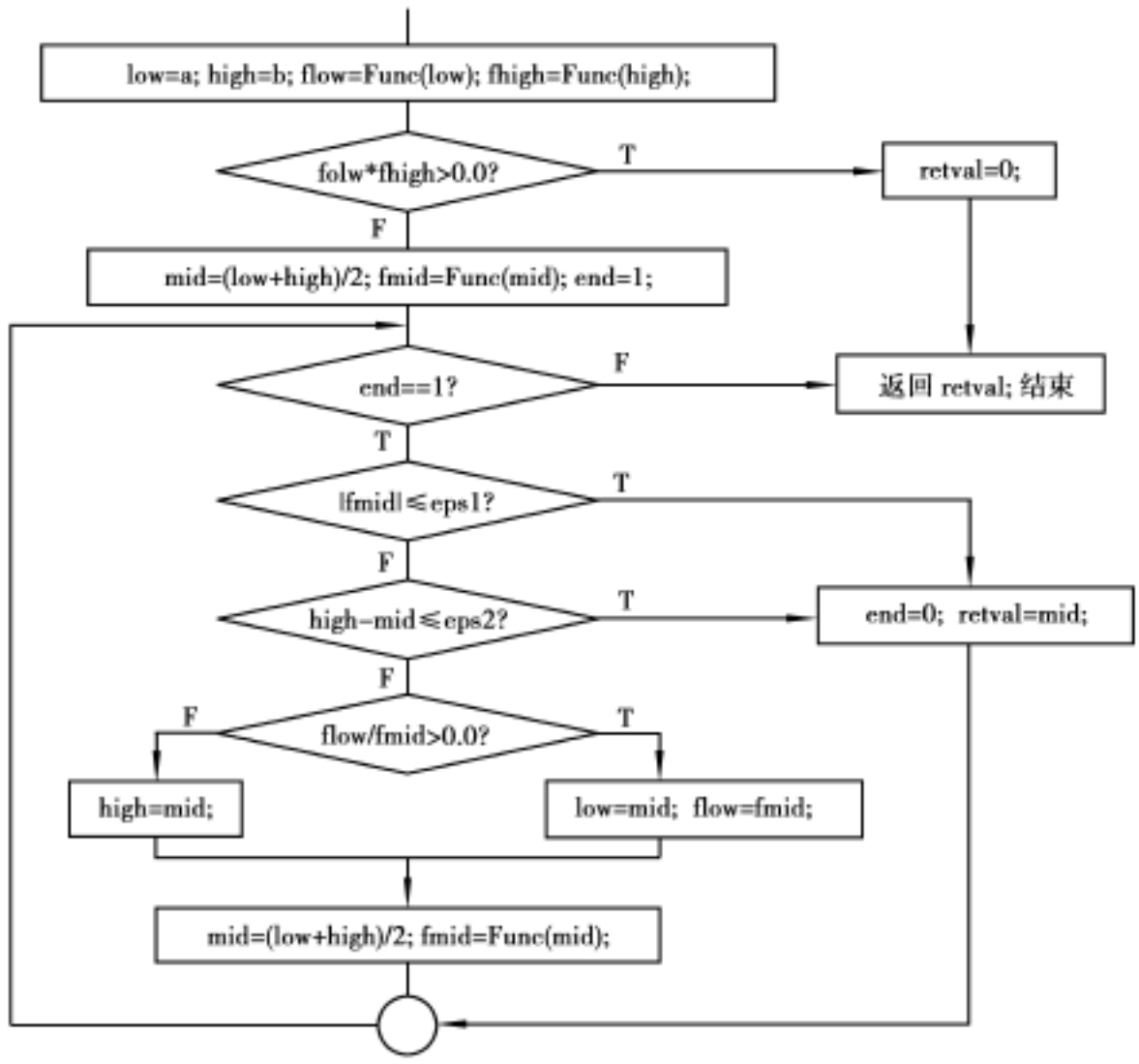


图 2.4 结构化的程序流程图

(2) N-S 图如图 2.5 所示:

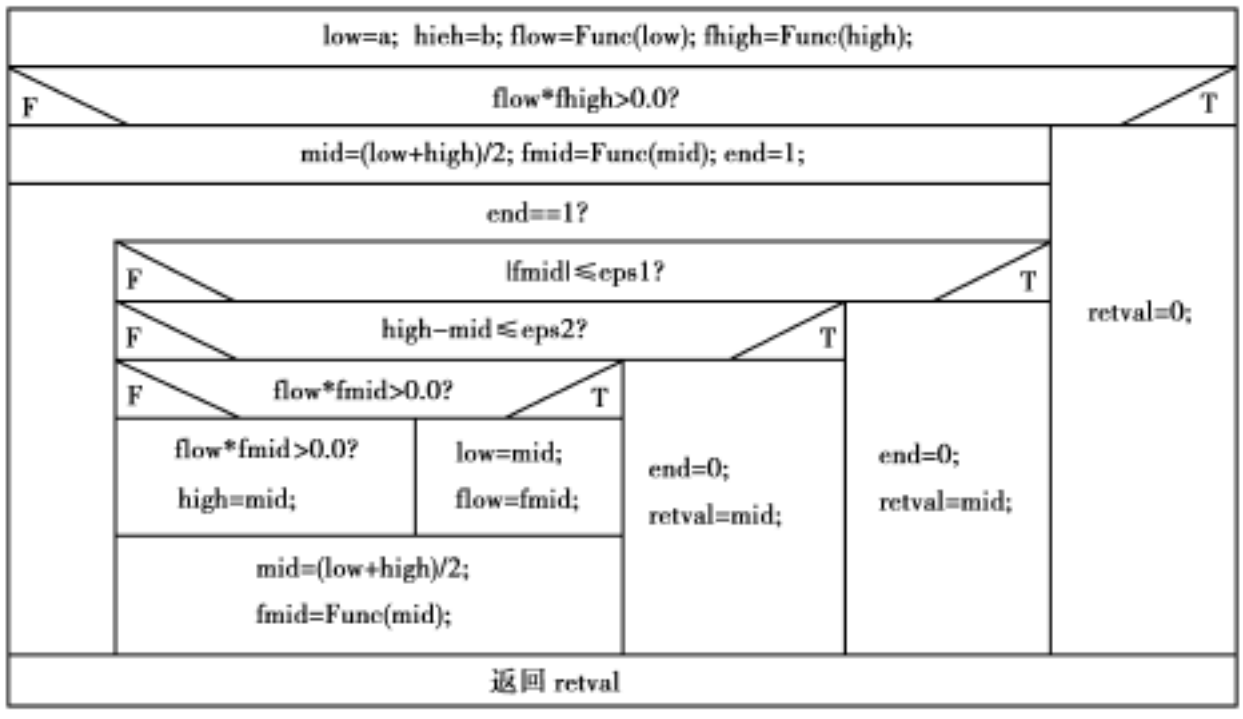


图 2.5 N-S 图

(3) 环路复杂性度量 V(G) = 6

8. A. ⑤,　B. ②,　C. ③,　D. ①,　E. ③

1960 年 Dijkstra 提倡的结构化程序设计方法, 反对(后来改为尽量避免) 在程序中使用 GOTO 语句, 认为使用顺序、选择、重复等三种基本控制结构进行组合、嵌套, 就可以构造出整个程序。这种程序设计方法可以提高程序的生产率, 还有利于将来程序的维护。

9. 开发规模相同, 但复杂性不同的软件, 花费的成本和时间会有很大的差异。因此到目前为止, 还没有一个软件复杂性度量的方法能够全面、系统地度量任一软件的复杂性, 某一种度量方法只偏重于某一方面。所以, 用某一种软件复杂性来度量不同类型的程序, 所得到的度量值不一定真正反映它们的复杂性。但对同一类型的程序, 按某种视点来度量它们的复杂性, 其结果还是比较有价值的。

10. K. Magel 从 6 个方面描述软件复杂性:

①理解程序的难度;

②改错及维护程序的难度;

③向他人解释程序的难度;

④按指定方法修改程序的难度;

⑤根据设计文档编写程序的工作量;

⑥执行程序时需要资源的程度。

软件复杂性度量模型应遵循的基本原则:

(1) 软件复杂性与程序大小的关系不是线性的;

(2) 控制结构复杂的程序较复杂;

(3) 数据结构复杂的程序较复杂;

(4) 转向语句使用不当的程序较复杂;

(5) 循环结构比选择结构复杂, 选择结构又比顺序结构复杂;

(6) 语句、数据、子程序和模块在程序中的次序对软件复杂性都有影响;

(7) 全程变量、非局部变量较多时程序较复杂;

(8) 参数按地址传递比按值传递更复杂;

(9) 函数副作用比显式参数传递更难以琢磨;

(10) 具有不同作用的变量共用一个名字时较难理解;

(11) 模块间或过程间联系密切的程序较复杂;

(12) 嵌套深度越深程序越复杂。

最典型的两种程序复杂性度量的方法中, McCabe 环路复杂性度量就是针对基本原则(2) 制定的度量模型; Halstead 软件科学则是针对程序中操作符和操作数的出现频度而制定的度量模型。

第 7 章　面向对象的分析与设计方法

7.1　本章主要内容

本章主要介绍了面向对象的基本概念,包括对象、类、继承、消息等,面向对象的特征。面向对象开发模型,用面向对象方法构造软件的开发过程,包括应用生存期和类生存期的概念。面向对象的分析和设计方法,例如:OTM 方法和 Booch 方法。此外,还介绍了面向对象设计的一种全新方法——UML 方法。

7.2　重点难点

理解掌握面向对象的概念:对象、类、继承、消息等。
用面向对象方法构造软件的开发过程。
理解面向对象分析方法。掌握 OO 分析方法的 5 个层次。
理解掌握面向对象设计方法。
掌握 OO 设计原则和高层设计模型,掌握有影响的 OMT 方法和 Booch 方法的基本思想。
了解 UML 方法的基本思想。

7.3　习题

1. 面向对象的程序设计语言具有数据抽象、信息隐蔽、(A) 等特征。作为运算单位的对象应具有下列特性: (B) 、(C) 、(D) 。(E) 是面向对象的语言。

供选择的答案:

A:　① 对象调用　②对象变换　③ 非过程性　④ 信息继承　⑤ 并发性

B ~D:①对象把数据和处理数据的操作结合为一体
②在程序运行时对象都处于活动状态
③对象在计算中可向其他对象发送消息
④接受消息的对象必须给消息发送者以回答
⑤对象的内部状态只根据外部送来的消息才操作

E:　①C ++, SMALLTALK, objectC　②C, Ada, Modula2
③PASCAL, C ++, APL　④Ada, objectC, C

2. 由 RumBaugh 等人提出的一种面向对象方法叫做对象模型化技术(OMT) ,即三视点技术,它要求把分析时收集的信息建立在三个模型中。第一个模型是(A) ,它的作用是描述系统的静态结构,包括构成系统的对象和类,它们的属性和操作,以及它们之间的联系。第二个模型是(B) ,它描述系统的控制逻辑,主要涉及系统中各个对象和类的时序及变化状况。(B) 包括两种图,即(C) 和(D) 。(C) 描述每一类对象的行为,(D) 描述发生于系统执行过程中的某一特定场景。第三个模型是(E) ,它着重于描述系统内部数据的传送与处理,它由多个数据流图组成。供选择的答案:

A, B, E: ①数据模型　②功能模型　③行为模型　④信息模型

⑤原型　⑥动态模型　⑦对象模型　⑧逻辑模型
⑨控制模型　⑩仿真模型

C, D: ① 对象图　②概念模型图　③状态迁移图　④数据流程图
⑤时序图　⑥事件追踪图　⑦控制流程图　⑧逻辑模拟图
⑨仿真图　⑩行为图

3. 在面向对象软件设计过程中, 应按如下要求进行类的设计: 只有类的共有界面的成员才能成为使用类的操作, 这就是软件设计的(A) 原则。当且仅当一个操作对类的实例的用户有用时, 它才是类公共界面的一个成员, 这是软件设计的(B) 原则。由同属一个类的操作负担存取或加工类的数据, 这是软件设计的(C) 原则。两个类之间的交互应当仅涉及参数表, 这是软件设计的(D) 原则。每个派生类应该当作基类的特殊化来开发, 而基类所具有的公共界面成为派生类的共有界面的一个子集, 这是软件设计的(E) 原则。供选择的答案:

A: ①过程抽象　②功能抽象　③信息隐蔽　④共享性　⑤连通性
B: ①标准调用　②最小界面　③高耦合　④高效率　⑤可读性
C: ①数据抽象　②低内聚　③高内聚　④低复杂度　⑤低强度
D: ①显式信息传递　②高内聚　③低内聚　④相互操作性　⑤连接性
E: ①动态联编　②异质表　③信息隐蔽　④多态性　⑤继承性

4. 对象是面向对象范型的(A) 。每个对象可用它自己的一组(B) 和它可以执行的一组(C) 来表征。应用执行对象的(C) 可以改变该对象的(B) 。它的应用必须通过(D) 的传递。可以认为, 这种(D) 的传递大致等价于过程性范型中的函数调用。某些语言提供了特殊功能, 允许对象引用自己。若一个对象没有显式地被引用, 则可让该对象(E) 。供选择的答案:

A. ①基本单位　②最小单位　③最大单位　④语法单位
B ~C. ①行为　②功能　③操作　④数据　⑤属性
D. ①接口　②消息　③信息　④操作　⑤过程
E. ①撤销　②歇着　③缺省　④隐式引用　⑤引用自己

5. 在面向对象软件开发过程中特别重视复用。软件构件应独立于当初开发它们的应用而存在。在以后的应用开发中, 可以调整这些独立构件以适应新问题的需要。因此, 应使得类成为一个(A) 的单元。这样就有一个(B) 生存期问题。(B) 生存期有自己的步骤, 与任一特定应用的开发(C) 。按照这些步骤, 可以完整地描述一个基本(D) 。而不仅仅考虑当前正在开发的系统。系统开发的各个阶段都可能会标识新的类。随着各个新类的标识, (B) 生存期引导开发工作逐个阶段循序渐进。

在设计与实现类时, 应尽可能利用既存类提供为当前应用所需要的功能, 利用既存类的三个可能途径是: (E) 复用既存类; 对既存类进行(F) 以得到满足要求的类; 重新开始进行开发。供选择的答案:

A. ①可复用　②可测试　③可适用　④可靠
B. ①应用　②寿命　③类　④软件
C. ①相关　②密切相关　③负相关　④ 无关
D. ①概念　②实体　③事件　④事情
E, F. ①修改　②更新　③照原样　④演化

6. 类常常被看作是一个抽象数据类型的实现, 更合适的是把类看作是某种(A) 的一个模

型。事实上,类是单个的(B) 语义单元。类的用户能够操纵的操作叫做类的(C) 。类定义的其余部分给出数据定义和辅助功能定义,包括类的实现。

类的实现常常包括了其他类的实例,这些实例(D) 被其他对象存取,包括同一个类的其他实例。类的实现可能还包括某些私有方法,实现它们的类可以使用,而其他任何对象都不能使用。

类,就它是一个数据值的聚合的意义上来看,与 Pascal 中的记录或 C 中的结构类似,但又有差别。类扩展了通常的记录语义,可提供各种级别的(E) 。类不同于记录,因为它们包括了操作的定义,这些操作与类中声明的数据值有相同的地位。供选择的答案:

A.	①功能	②概念	③结构	④数据
B.	①语法	②词法	③语义	④上下文环境
C.	①界面	②操作	③行为	④活动
D.	①可自由地	②可有控制地	③可通过继承	④不应受保护
E.	①可移植性	②可重复性	③可访问性	④继承性

7. 请说明下面有关范型的叙述的正确答案。

(1) 问题的解决是基于规则的,它把有关问题的知识分解成一组具体规则,用语言的 if_then 等结构来表示这些规则。

(2) 问题的解决把软件视为由一系列步骤构成的算法。每一步骤都是带有预定输入和特定输出的一个过程,连贯起来产生合理的稳定的贯通于整个程序的控制流。

(3) 把一个问题分解成独立执行的模块。让不止一个程序(进程) 同时运行。这些进程互相配合,解决问题。

(4) 把标识和模型化问题论域中的主要实体作为系统开发的起点,主要考虑对象的行为而不是必须执行的一系列动作。

供选择的答案:

①面向存取	②面向对象	③过程性	④逻辑性
⑤函数型	⑥面向进程	⑦说明型	⑧原型

8. 有一种非形式的技术,对于捕获信息有时很有用,它就是 CRC 卡片。CRC 是(A) 、(B) 和(C) 的缩写。它可以用来组织在每一个子系统中的类。以 CRC 卡片为辅助工具的设计有以下几个步骤: 识别(A) 和(B) , 分配(B) , 找寻(C) , 细化。CRC 的作者强调模拟在执行每个基本功能时系统内部出现的(D) , 以此推动细化工作的进行。在这个过程中,CRC 卡片是十分重要的一个工具。

用 CRC 卡片来进行设计,既不是传统的“ 自上而下 ”,也不是“ 自下而上 ”,而是从已知到未知的逐步(E) 的过程。

供选择的答案:

A ~C.	①控制	②协作者	③可靠性	④类
	⑤计算	⑥职责	⑦比较	⑧上下文环境
D.	①场景	②算法	③进程	④变换
E.	①演化	②进展	③展开	④认识

9. 论域分析的(A) 和对应用分析和高层设计的(B) 就构成问题论域的模型。已有许多建立这种模型的技术,一种特别适用的技术就是语义数据模型。

语义数据模型来源于 Codd 的(C) 数据模型和实体——联系模型, 并对这类模型进行了扩充和一般化。语义数据模型可以表达问题论域的内涵, 还可以表示复杂对象和对象之间的联系。语义数据模型与(C) 数据模型本来都是在(D) 设计时使用的, 但它们的范围已经扩展到系统的开发。作为(D) 结构标准的 ANSI/SPARC 建议提出了三层模型: 外部模型、概念模型和(E) 模型。这三层可以被映象到面向对象设计的三个层次上去。外部模型与概念模型层相当于高层设计阶段。

供选择的答案:

A, B.	①控制	②输出	③输入	④处理
	⑤计算	⑥解释	⑦比较	⑧创建
C.	①变换	②关系	③抽象	④事务
D.	①网络	②程序	③算法	④数据库
E.	①低层	②底层	③内部	④存储

10. 从分析到设计的过程流如图所示。一旦已经开发完成一个合理完整的(A) 模型后, 就要着手(B) 的设计。这需要描述(B) 的特征, 以准确表达待实现的用户需求, 以及实现需求所必需的支持环境。一旦定义了各个(B) , 就开始(C) 设计, 这时, 可利用 CRC 卡片, 将属性转换为(D) , 将关系转换为(E) 。

供选择的答案:

A ~C.	①分析	②系统设计	③模块设计	④子系统设计
	⑤对象设计	⑥数据设计	⑦操作设计	⑧行为设计
D ~E.	①对象	②数据结构	③算法	④消息传递
	⑤控制	⑥并发处理	⑦进程	⑧过程

11. 什么叫面向对象? 为什么要用面向对象方法开发软件?

12. 面向对象方法的特点是什么?

13. 分别说明什么是对象、类、消息?

14. 分别说明对象模型、动态模型和功能模型的特征, 说明三种分析模型的关系。

15. 简述对象建模、动态建模和功能建模的基本步骤。

16. 说明面向对象设计的步骤。

17. 面向对象开发方法与面向数据流的结构化开发方法有什么不同? 使用面向对象开发方法的优点在什么地方?

18. 在类的通过复用的设计中, 主要的继承关系有哪几种? 试举例说明。

19. 在类的设计中需要遵循的方针是什么? 三个主要的设计准则: 抽象、信息隐蔽和模块化如何才能做到?

20. 建立分析和设计模型的一种重要方法是 UML。试问 UML 是一种什么样的建模方法? 它如何表示一个系统?

7.4　习题答案

1. A. ④　B. ①　C. ③　D. ④　E. ①　其中, B、C、D 的答案可互换。

面向对象的程序设计语言应具备面向对象方法所要求的 4 个成分: 类、对象、继承和消息通信。类与对象由数据抽象和信息隐蔽得到, 此外语言应具有信息继承的机制。对象由一组

属性和它可以执行的一组操作来定义。面向对象的软件系统通过对象间的消息通信和对象执行消息所要求的服务完成系统预定的功能。所以,对象在计算中可向其他对象发送消息,接受消息的对象必须通过响应消息/执行服务,给消息发送者以回答。

C++, Smalltalk, object C 是面向对象的程序设计语言, Ada、Modula2 是基于对象的程序设计语言,因为它缺少继承的机制,而 Pascal, C, APL 等都不是面向对象或基于对象的程序设计语言。

2. A. ⑦ B. ⑥ C. ③ D. ⑥ E. ②

在 OMT 中,把分析时收集的信息建立在三个模型中。第一个模型是对象模型,它的作用是描述系统的静态结构,包括构成系统的对象和类,它们的属性和操作,以及它们之间的联系。第二个模型是动态模型,它描述系统的控制逻辑,主要涉及系统中各个对象和类的时序及变化状况。动态模型包括两种图,即状态迁移图和事件追踪图。状态迁移图描述每一类对象的行为,事件追踪图描述发生于系统执行过程中的某一特定场景。第三个模型是功能模型,它着重于描述系统内部数据的传送与处理,它由多个数据流图组成。

3. A. ③ B. ② C. ③ D. ① E. ⑤

在面向对象软件设计过程中,应按如下要求进行类的设计:只有类的共有界面的成员才能成为使用类的操作,这就是软件设计的信息隐蔽原则。当且仅当一个操作对类的实例的用户有用时,它才是类公共界面的一个成员,这是软件设计的最小界面原则。由同属一个类的操作负担存取或加工类的数据,这是软件设计的高内聚原则。两个类之间的交互应当仅涉及参数表,这是软件设计的显式信息传递原则。每个派生类应该当作基类的特殊化来开发,而基类所具有的公共界面成为派生类的共有界面的一个子集,这是软件设计的继承性原则。

4. A. ①, B. ⑤, C. ③, D. ②, E. ③

5. A. ①, B. ③, C. ④, D. ②, E. ③, F. ④

6. A. ②, B. ③, C. ①, D. ④, E. ③

7. (1) ④ (2) ③ (3) ⑥ (4) ②

8. A. ④, B. ⑥, C. ②, D. ①, E. ③

9. A. ③, B. ②, C. ②, D. ④, E. ③

10. A. ①, B. ④, C. ⑤, D. ②, E. ④

11. 关于“面向对象”,有许多不同的看法。Coad 和 Yourdon 给出了一个定义:“面向对象=对象+类+继承+消息通信”。如果一个软件系统是使用这样 4 个概念设计和实现的,则认为这个软件系统是面向对象的。

使用面向对象方法开发软件的好处是:

开发方法的惟一性,开发阶段的高度连续性,表示方式的一致性;

问题空间实体的自然表示,减轻了设计者的负担,在设计系统之初不必考虑一个很完整的解决方案。

建立稳定的系统结构,可促进复用性,易于维护,易于修改,可合理利用共同性,减少复杂性。

12. 面向对象方法的特点是:

方法的惟一性,即方法是对软件开发过程所有阶段进行综合考虑而得到的。

从生存期的一个阶段到下一个阶段的高度连续性,即生存期后一阶段的成果只是在前

一阶段成果的补充和修改。

把面向对象分析(OOA) 、面向对象设计(OOD) 和面向对象程序设计(OOP) 集成到生存期的相应阶段。

13. 对象的定义: 对象是面向对象开发模式的基本成分, 是现实世界中个体或事物的抽象表示。每个对象可由一组属性和它可以执行的一组操作来定义。

类的定义: 具有相同或相似性质的对象的抽象就是类。类成为某些对象的模板, 抽象地描述了属于该类的全部对象的属性和操作。属于某个类的对象叫做该类的实例。因此, 对象的抽象就是类, 类的具体化就是对象, 也可以说类的实例是对象。对象的状态则包含在它的实例变量, 即实例的属性中。类定义了各个实例所共有的结构, 类的每一个实例都可以使用类中定义的操作。实例的当前状态是由实例所执行的操作定义的。

消息: 对象之间进行通信的构造叫做消息。在对象的操作中, 当一个消息发送给某个对象时, 消息包含接收对象去执行某种操作的信息。接收消息的对象经过解释, 然后给予响应。这种通信机制称为消息传递。发送一条消息的格式是“ 对象名. 方法名(参数) ”。

14. 对象模型表示了静态的、结构化的系统数据性质, 描述了系统的静态结构, 它是从客观世界实体的对象关系角度来描述。表现了对象的相互关系。该模型的特征是用对象图来表现对象的结构、属性和操作, 它是分析阶段三个模型的核心, 也是其他两个模型的框架。

动态模型是与时间和变化有关的系统性质, 该模型描述了系统的控制结构, 它表示了瞬时的、行为化的系统控制性质, 它关心的是系统的控制, 操作的执行顺序, 它从对象的事件和状态的角度出发, 表现了相互行为。

功能模型描述了系统的所有计算。功能模型指出发生了什么, 动态模型确定什么时候发生, 而对象模型确定发生的客体。功能模型表明一个计算如何从输入值得到输出值, 它不考虑所计算的次序。功能模型由多张数据流图组成。数据流图说明数据流是如何从外部输入、经过操作和内部存储输出到外部的。功能模型也包括对象模型中值的约束条件。

三种分析模型之间关系是这样的: 功能模型指出发生了什么, 动态模型确定什么时候发生, 而对象模型确定发生的客体。

15. 略

16. 对象设计要确定实现用到的类、关联的完整定义, 接口的形式以及实现操作方法的算法, 可以增加实现必需的内部对象, 对数据结构和算法进行优化。

(1) 获得操作

(2) 确定操作的目标对象

(3) 算法设计

(4) 优化设计

(5) 控制的实现

(6) 调整继承

(7) 关联的设计

17. 结构化开发方法是使用最广泛、历史最长的过程化开发方法。结构化开发方法产生过程的抽象, 这些抽象把软件视为处理流, 定义构成一系列步骤的算法, 每一步骤都是带有预定义输入和特定输出的一个过程, 把这些步骤串联在一起可产生合理的稳定的贯通于整个程序的控制流。这将最终导致一个很简单的具有静态结构的体系结构。

在结构化开发方法中,数据结构是应算法步骤的要求而开发的。数据结构贯穿于过程,提供过程需要传送给它的操作的信息。系统的状态是一组全局变量,这组全局变量保持了状态的值,把它们从一个过程传送到另一个过程。

结构化开发方法是一种成熟的应用开发过程。对这种方法已经存在许多支持。然而,在大型系统的开发上和在面向用户系统的构造上存在一些问题。改进大型系统开发的技术主要集中在开发数据抽象。日益增多的考虑是使用抽象数据类型,把过程化系统开发过程包括到数据驱动的方法中。随着大型系统的开发,接踵而来的问题就是要把过程抽象与数据抽象方法组合起来,这种需要导致了面向对象开发方法的诞生。

面向对象开发方法是我们分解问题所使用方法演化的结果。在结构化开发方法中过程抽象是优先的,而面向对象开发方法中优先的是实体,即问题论域的对象。在面向对象开发方法中,把标识和模型化问题论域中的主要实体作为系统开发的起点,主要考虑对象的行为而不是必须执行的一系列动作。

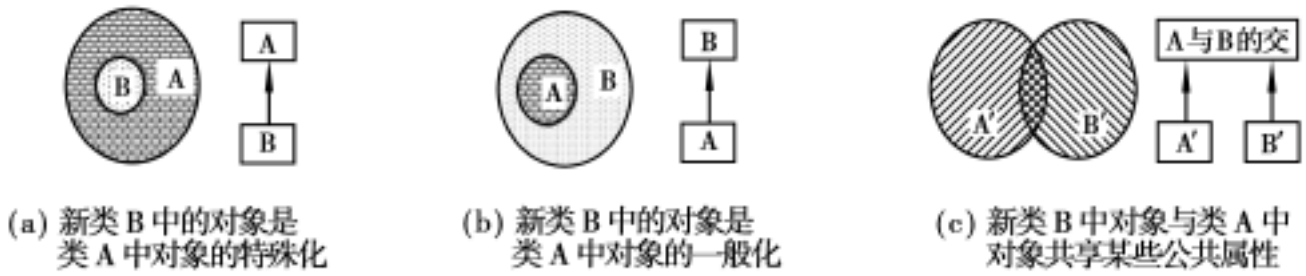
面向对象系统中的对象是数据抽象与过程抽象的一个混合体。表示这些实体的数据抽象是面向对象设计过程的主要产品,系统的状态保存在各个数据抽象的核心所定义的数据存储中。控制流被分成块,并被包括在各个在数据抽象上的各个操作里面。不像在结构化开发方法里那样,把数据从一个过程传送到另一个过程,而是控制流从一个数据抽象被传送到另一个数据抽象。完成的系统体系结构更复杂但也更灵活。在块中分离的控制流允许把复杂的动作视为局部的相互影响。

18. 在类的通过复用的设计中,主要的继承关系有两大类:

①配置:利用既存类来设计类,可能会要求由既存类的实例提供类的某些特性。通过把相应类的实例声明为新类的属性来配置新类。例如,一种仿真服务器可能要求使用一个计时器来跟踪服务时间。设计者不必开发在这个行为中所需的数据和操作,而是应当找到计时器类,并在服务器类的定义中声明它。

但如果使用既存类的内部表示来做为新类的内部表示的一部分,这是一种“针对实现”的继承方式,这种继承方式不好。例如,考虑使用继承来实现一个 Circle 类。Point 类可支持 Circle 类的一部分实现。为了定义一个圆,我们只需要定义一个点和一个值,作为圆的圆心和半径。把 Point 当作子类,Circle 类不但能得到由 x 和 y 提供的圆心,而且还能得到一个操作,让圆能够自由移动。但这样做,我们失去了抽象。

②演变:要开发的新类可能与一个既存类非常类似,但不完全相同。此时可以从一个既存类演变成一个新类,可以利用继承机制来表示一般化—特殊化的关系。特殊化处理有三种可能的方式,如图 2.6 所示。



(a) 新类 B 中的对象是类 A 中对象的特殊化　(b) 新类 B 中的对象是类 A 中对象的一般化　(c) 新类 B 中对象与类 A 中对象共享某些公共属性

图 2.6　演变的三种方式

如果新的概念是一个既存类所表示概念的一个特殊情况,特殊化运算可以从该既存类

的定义产生新类的初始构造, 这是典型的类继承的使用。既存类 A 的数据结构和操作可以成为新类 B 的一部分, 如图 2. 6(a) 所示。既存类 A 的公共操作成为新类 B 的共有界面部分。

如果新类比软件库中那些既存类更一般, 则新类 B 不具有既存类 A 的全部特性, 一般化运算把两个类中共同的特性移到新的更高层的类中, 高一层的类是 B, 我们将要设计它。原来的类 A 成为新类 B 的子类。如图 2. 6(b) 所示。

一个既存类 A 与我们设计的新类 B 共享概念的某一个部分, 则两个概念的共同部分形成新类的基础, 且既存类与新类两者成为子类, 如图 2. 6(c) 所示。

19. 在设计类时需要遵循的方针是:

信息隐蔽: 通过信息隐蔽可保护类的存储表示不被其他类的实例直接存取。

消息限制: 该类实例的用户应当只能使用界面提供的操作。

狭窄界面: 只有对其他类的实例是必要的操作才放到界面上。

强内聚: 模块内部各个部分之间应有较强的关系, 它们不能分别标识。

弱耦合: 一个单独模块应尽量不依赖于其他模块。

显式信息传递: 两个类之间的交互应当仅涉及显式信息传递。

派生类当作派生类型: 每个派生类应该当作基类的特殊化来开发, 而基类所具有的公共界面成为派生类的共有界面的一个子集。

抽象类: 某些语言提供了一个类, 用它作为继承结构的开始点, 所有用户定义的类都直接或间接以这个类为基类。

为了在类的设计中做到抽象、信息隐蔽和模块化:

以类作为系统的基本模块单元, 通过一般化-特殊化关系和整体-部分关系, 搭建整个系统的类层次结构, 实现数据抽象和过程抽象;

将数据和相关的操作封装在类内部, 建立共有、私有和子类型等存取级别, 将数据表示定义成为类的私有成员, 实现信息隐蔽。

通过建立类属性(类模板) , 将某些有可复用要求的类设计成在数据类型上通用的可复用的软件构件, 这样有助于实现模块化。

20. UML 叫做统一的建模语言, 它把 Booch、Rumbaugh 和 Jacobson 等各自独立的 OOA 和 OOD 方法中最优秀的特色组合成一个统一的方法。UML 允许软件工程师使用由一组语法的语义的实用的规则支配的符号来表示分析模型。

在 UML 中用 5 种不同的视图来表示一个系统, 这些视图从不同的侧面描述系统。每一个视图由一组图形来定义。这些视图概述如下:

用户模型视图: 这个视图从用户(在 UML 中叫做参与者) 角度来表示系统。它用使用实例(use case) 来建立模型, 并用它来描述来自终端用户方面的可用的场景。

结构模型视图: 从系统内部来看数据和功能性。即对静态结构(类、对象和关系) 模型化。

行为模型视图: 这种视图表示了系统动态和行为。它还描述了在用户模型视图和结构模型视图中所描述的各种结构元素之间的交互和协作。

实现模型视图: 将系统的结构和行为表达成为易于转换为实现的方式。

环境模型视图: 表示系统实现环境的结构和行为。

通常, UML 分析建模的注意力放在系统的用户模型和结构模型视图, 而 UML 设计建模则

定位在行为模型、实现模型和环境模型。

第 8 章　软件质量

8.1　主要内容

本章主要介绍了关于软件质量的定义及其属性,说明了影响软件质量的因素,软件质量保证策略,软件质量保证活动,软件质量保证标准和软件的评审。

8.2　重点难点

理解软件质量的定义及其属性。
了解影响软件质量的因素。
理解什么软件质量保证及其主要功能。
了解软件质量保证标准。
理解进行软件的评审原因和软件设计质量评审的内容。

8.3　习题

1.软件质量必须在设计和实现过程中加以保证。为了确保每个开发过程的质量,防止把差错传播到下一个过程,必须进行(A) 。(A) 是质量保证活动的一个重要部分,其目的有两个:其一是切实搞好开发阶段的管理;其二是(B) 软件差错给用户造成损失。(A) 的实施有两种方式:(C) 和(D) 。(C) 即白盒测试和黑盒测试。(A) 的类型有(E) 、(F) 、(G) 和(H) 。

供选择的答案:

A.	①质量保证	②差错检测	③质量检验	④质量管理
B.	①预先防止	②完全避免	③减少	④消除
C, D.	①鉴定	②测试	③实际运行管理	④实际运行检验
E ~H.	①供货检验	②集中测试	③中间检验/阶段评审	
	④产品检验	⑤用户检测	⑥程序检测	⑦验收检验

2.为了开发高质量的软件,从计划阶段开始,不但要明确软件的功能,还要明确软件应当达到什么样的质量标准,即指定软件的质量指标。软件质量度量和保证的条件通常有以下几项:(A) ,即必须制定能够适应用户要求、软件类型和规模的质量标准;(B) ,即人人都容易掌握;(C) ,即对于同一软件评价的结果必须一致;(D) ,即各阶段确立质量目标并落实;(E) ,即从不同角度加以评价;(F) ,软件的质量评价标准分为 3 级:(G) 、(H) 和(I) 。(G) 的着眼点是“是否满足用户的要求”;(H) 的着眼点是“开发者在设计实现时是否按照软件需求保证了质量”;(I) 是为定量度量软件质量而规定的一些检查项目。它们一级比一级(J) ,一级比一级易于(K) 。

供选择的答案:

A ~F.	①可靠性	②易学习性	③适应性	④针对性
	⑤客观性	⑥经济性	⑦主观性	⑧多样性

G ~I. ①子质量特性　②质量度量准则　③质量特性
　　④质量维护准则　⑤质量管理准则
J, K. ①抽象　②具体　③检验　④定量评价

3. 从技术上改进软件的开发过程, 提高软件产品的质量无非是两个方面: 一是提高(A) , 二是改进(B) 。在发现错误和排除错误方面更重要也是更困难的是(C) 。由于软件测试技术方面没有多少新的突破, 人们只能用加强阶段评审或是检查作为辅助手段。这是一个由同行人员小组(D) 所开发的阶段产品的验证方法。至于改进(B) 的新技术, 是采用面向对象的开发技术或是建立(E) 。一个诱人的说法是采用(F) 技术, 其基本思想在于(G) 开发过程, 使得差错或缺陷不可能有机可乘混入开发过程。

供选择的答案:

A, B. ①测试效率　②开发速度　③开发过程　④维护过程
　　⑤测试方法　⑥开发工具
C.　①排除错误　②发现错误
D.　①机器检查　②人工检查　③集成测试　④单元测试
E, F. ①智能　②软件原型　③“ 净室 ”软件开发　④基于构件的复用
G.　① 简化　②结构化　③净化　④强化

4. 软件配置项是软件配置管理的对象, 指的是软件工程过程中所产生的(A) 。不仅如此, 而且在不同的时期, 出于不同的要求, 可进行各种组合, 如针对不同的硬件环境和(B) 的组合, 这就是软件配置的概念。实施软件配置管理要做的事情有: 制定(C) ; 实施(D) ; 实施版本管理和发行管理。制定适当的命名规则是(E) 的重要工作, 命名要求: (F) , 目的在于避免出现重名, 以免造成混乱; (G) , 以反映命名对象之间的关系。

供选择的答案:

A, B. ①接口　②软件环境　③信息项　④“ 版本 ”
C ~E. ①配置标识　②配置管理　③配置管理计划　④变更管理
　　⑤版本变化　⑥配置审核
F, G. ①多态性　②惟一性　③独立性　④可追溯性

5. 在变更管理中,“ 检出 ”和“ 登录 ”处理实现了两个重要的变更控制要素, 即(A) 和(B) 。(A) 控制和管理各个软件技术人员存取或修改一个特定软件配置对象的权限; (B) 可用来确保由不同的人所执行的并发的变更不会产生混乱。请选择适合的答案完成图 2. 7 所示的存取和控制图。

供选择的答案:

A, B. ①异步控制　②同步控制　③存取控制　④基线控制
C, D. ①登入　②检出　③管理　④填写变更请求
E.　①审查人员　②配置人员　③质量保证人员　④软件工程人员

6. 软件工程标准的类型是多方面的。它可能包括(A) 标准, 如方法、技术、度量等; (B) 标准, 如需求、设计、部件、描述、计划、报告等; (C) 标准, 如职别、道德准则、认证、特许、课程等; (D) 标准, 如术语、表示法、语言等。根据中国国家标准 GB/T 15538—1995(软件工程标准分类法) 规定, 软件工程标准可用一张(E) 来表示。

供选择的答案:

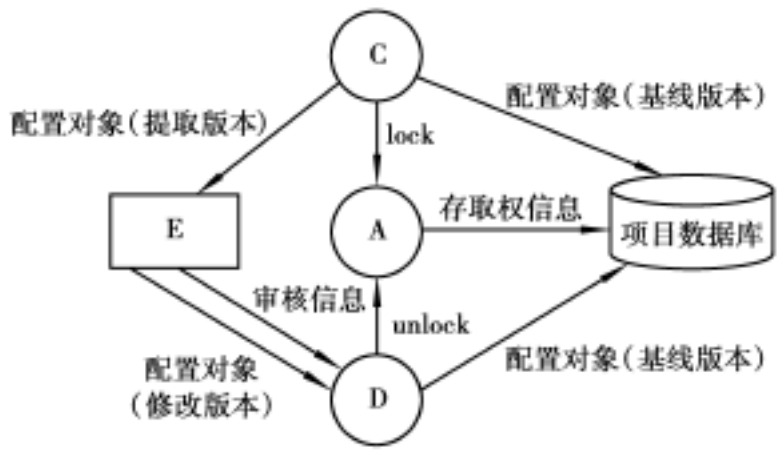


图 2.7　存取和控制图

A～D.①专业　②产品　③记法　④过程　⑤管理

E.　①矩阵图　②一维的表格　③二维的表格　④帕列特图

7.根据软件工程标准制定的机构和标准的适用范围有所不同,它可分为 5 个级别,即(A)、(B)、(C)、(D)和(E)。(A)由国际联合机构制定和公布,通常冠有 ISO 字样;而 GB、ANSI、JIS 等属于(B);IEEE、GJB 等属于(C);(D)的实例是(F);(E)是专用的软件工程规范,如(G)。

供选择的答案:

A～E.①国家标准　②国际标准　③企业规范　④项目规范

⑤行业标准

F, G.　①DOD-STD　②FIPS(NBS)　③BS　④MIL-S

⑤IBM 的《程序设计开发指南》　⑥CIMS 的《软件工程规范》

8.下面关于标准和文档的叙述中正确的叙述为(　、　、　、　、　)。

①国家标准是由政府或国家级机构制定或批准,适用于全国的标准。这些标准都是强制性的,相关产品必须严格执行标准。

②ISO 9001 是设计/开发、生产、安装和服务中的质量保证模式,ISO 9000—3 是使 ISO 9001 适合于软件的质量保证指南。

③软件工程标准化可提高软件的生产率。

④软件质量保证体系是贯穿于整个软件生存期集成化过程体系,而不仅仅体现在最后产品的检验上。

⑤ISO 9000—3 与具体的开发模式有关。它将软件全过程工序从管理角度、合同角度和工程角度划分为三大类。

⑥软件测试计划始于需求分析阶段,完成于软件设计阶段。

⑦任何一个文档都应是完整的、独立的,它应自成体系。

⑧在新文档取代旧文档后,管理人员应删去旧文档。

⑨软件开发机构应保存一份完整的主文档,并允许开发人员可以保存主文档中的部分主文档,有自己的活动空间。

⑩软件需求分析报告是给开发人员使用的,不是给其他人员,如维护人员、用户等使用的。

9.从供选择的答案中选出同下列各条叙述关系最密切的字句。

(1)在软件开发中以下几方面的内容应分别在哪个文档中得到阐明:

A) 软件总体结构　　B) 运行环境　　C) 出错处理设计

(2) 以下两个文档应分别在哪两个阶段中开发:

D) 初步的用户手册　　E) 确认测试计划

供选择的答案:

A ～C. ①可行性研究报告　②项目开发计划　③软件需求规格说明
④数据要求规格说明　⑤概要设计规格说明　⑥详细设计规格说明
⑦测试计划　⑧测试报告　⑨ 用户手册

D, E: ①可行性研究与计划　②需求分析　③概要设计
④详细设计　⑤测试　⑥ 维护

10. 从下列关于文档编制的叙述中选出五条正确的叙述。

①可行性研究报告应评述为了合理地达到开发目标而可能选择的各种方案, 以便用户抉择。因此, 编写者不必提出结论。

②操作手册的编写工作应该在软件测试阶段之前完成。

③软件的开发单位应该建立本单位文档的标识方法, 使文档的每一页都具有明确的标识。

④为了使得文档便于修改保持一致性, 各文档的内容不应有相互重复的地方。

⑤用户手册要使用专门术语, 并充分地描述该软件系统的结构及使用方法。

⑥详细设计说明书中可以使用判定表及必要的说明来表示程序的逻辑。

⑦概要设计说明书中可以使用 IPO 图来说明接口设计。

⑧测试分析报告应把每个模块实际测试的结果, 与软件需求规格说明书和概要设计说明书中规定的要求进行对照并作出结论。

⑨软件需求规格说明书中可以对软件的操作人员和维护人员的教育水平和技术专长提出要求。

⑩项目开发计划除去规定项目开发所需的资源、开发的进度等以外, 还可以包括用户培训计划。

11. 保证软件过程质量应包含的工作有: (A) 过程标准, (B) 开发过程, (C) 软件过程。近年来, 软件工程界的专家对过程评估和过程改进表现出浓厚的兴趣。其中有卡内基-梅隆大学 SEI 的(D) ; ISO 的(E) 。(D) 的主题是(F) , (E) 的主题是(G) 。Bell 的(I) 的主题是(H) 。

供选择的答案:

A ～C. ①监控　②报告　③定义　④执行

D, E, I. ①ISO 9001, ISO 9000—3　②CMM
③ISO/IEC TR15504 SPICE　④MIL-STD—498
⑤TRILLTUM　⑥ V-Model　⑦BooTSTRAP

F ～H. ①软件过程改进和能力评估　②建立和维持质量体系
③软件过程能力评估　④软件过程评估(基于 CMM)
⑤软件过程定义

12. 人们用于开发和维护软件及其相关产品的一系列活动称为(A) ; 表示(开发组织或项目组) 遵循其软件过程所获得的实际结果称为(B) ; 而描述通过遵循其软件过程能够实现预期结果的程度称为(C) ; 一个特定软件过程被明确和有效地定义、管理、测量和控制的程度称为(D) ; 表征(D) 的平台称为(E) ; 对软件机构进化阶段的描述, 随着软件机构定义、实践、测量、

控制和改进其软件过程,软件机构的能力经过这些阶段逐步前进,称为(F);互为关联的若干软件实践活动和有关基础设施的一个集合称为(G);对(G)的实施起关键作用的方针、规程、措施、活动以及相关基础设施的建立称为(H)。

供选择的答案:

A～H. ①关键过程域　②软件能力成熟度模型　③关键实践
④软件过程　⑤软件过程能力　⑥软件过程成熟度
⑦软件过程性能　⑧软件能力成熟度等级

13. 质量体系是一种(A)。质量体系是通过若干(B)来实现的。(B)是将(C)转化为(D)的一组彼此相关的资源和活动。(B)本身具有(E)的效果,是一种有效益的行为。(F)是通过质量体系来实现的。一个组织机构建立自己的质量体系,并使之有效地运行,使达到(F)目标的重要手段。请选择合适的答案,完成图2.8的质量体系文件。

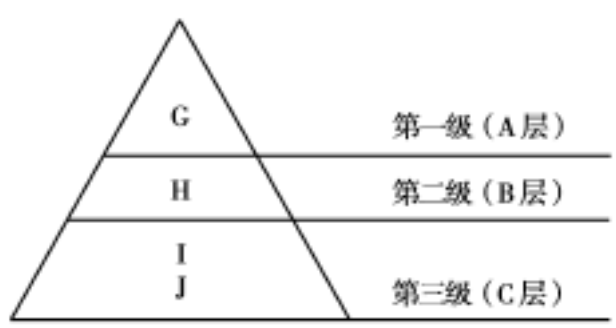


图2.8　质量体系文件

供选择的答案:

A.　①体系结构　②管理手段　③质量改进过程　④质量管理制度

B～E. ①输出　②输入　③接口　④减少
⑤增值　⑥过程　⑦扩充

F.　①质量保证　②质量管理　③质量检验　④质量改进

G～J. ①质量记录　②程序文件　③质量手册　④作业指导书

14. 软件质量与软件质量保证的含义是什么?

15. 影响软件质量的因素有哪些?

16. 什么是软件质量保证策略?软件质量保证的主要任务是什么?

17. 说明容错软件的定义与容错的一般方法。

18. 为什么要进行软件评审?软件设计质量评审与程序质量评审都有哪些内容?

8.4　习题答案

1. A. ③,　B. ①,　C. ④,　D. ①,　E. ①,　F. ③,　G. ④,　H. ④。其中,E、F、G、H的答案顺序可互换。

2. A. ③,　B. ②,　C. ①,　D. ④,　E. ⑤,　F. ⑥,　G. ③,　H. ①,　I. ②,　J. ②,　K. ④

3. A. ①,　B. ③,　C. ②,　D. ②,　E. ②,　F. ③,　G. ③

4. A. ③,　B. ②,　C. ③,　D. ④,　E. ①,　F. ②,　G. ④

5. A. ③,　B. ②,　C. ②,　D. ①,　E. ④

6. A. ④,　B. ②,　C. ①,　D. ③,　E. ③

7. A. ②,　B. ①,　C. ⑤,　D. ③,　E. ④,　F. ⑤,　G. ⑥

8. ①错　②对　③对　④对　⑤错　⑥对　⑦对　⑧错　⑨错　⑩错

9. A. ⑤,　B. ③,　C. ⑤,　D. ②,　E. ②

10. ①错　②对　③对　④错　⑤错　⑥对　⑦对　⑧错　⑨错　⑩对

11. A. ③,　B. ①,　C. ②,　D. ②,　E. ①,　F. ③,　G. ②,　H. ④,　I. ⑤

12. A. ④,　B. ⑦,　C. ⑤,　D. ⑥,　E. ⑧,　F. ②,　G. ①,　H. ③

13. A. ④,　B. ⑥,　C. ②,　D. ①,　E. ⑤,　F. ②,　G. ③,　H. ②,　I. ①,　J. ④。其中, I 和 J 的答案顺序可互换。

14. 从实际应用来说, 软件质量定义包括 3 个一致性:

(1) 与所确定的功能和性能需求的一致性;

(2) 与所成文的开发标准的一致性;

(3) 与所有专业开发的软件所期望的隐含特性的一致性。

软件质量保证是指确定、达到和维护所需要的软件质量而进行的所有有计划、有系统的管理活动。

15. 影响软件质量的因素分为可以直接度量的因素和只能间接度量的因素。

(1) 正确性和精确性;

(2) 性能与效率;

(3) 易用性;

(4) 可理解性与简洁性

(5) 可复用性与可扩充性

16. 软件质量保证策略是指软件质量保证工作的过程和侧重点。

质量保证的主要任务包括以下几点:

(1) 正确定义用户的要求。

(2) 技术方法的应用。

(3) 提高软件开发的工程能力。

(4) 软件的复用。

(5) 发挥每个开发者的能力。

(6) 组织外部力量协作。

(7) 排除无效劳动。

(8) 提高计划和管理质量。

17. 容错软件的定义有 4 种, 指规定功能的软件。

(1) 在一定程度上对自身错误的作用具有屏蔽能力的软件。

(2) 在一定程度上能从错误状态自动恢复到正常状态的软件。

(3) 在因错误而发生错误时, 仍然能在一定程度上完成预期的功能的软件。

(4) 在一定程度上具有容错能力的软件。

实现容错技术的主要手段是冗余。冗余通常分为 4 类:

(1) 结构冗余, 又分为静态、动态和混合冗余 3 种。

(2) 信息冗余

(3) 时间冗余

(4)冗余附加技术

因为软件生存期每个阶段的工作都有可能引入人为错误,如果某一阶段的错误不及时纠正,就会传播到开发的后结阶段,引出更多错误,因此,进行软件评审是必要的,评审可以揭露软件中的缺陷然后加以改正。

18. 设计质量评审的对象是在需求分析阶段产生的软件需求规格说明、数据需求规格说明,在软件概要设计阶段产生的软件概要设计说明书等。主要内容有:

(1)评价软件的规格说明是否合乎用户的要求。

(2)评审可靠性。

(3)评审保密措施实现情况。

(4)评审操作特性实施情况。

(5)评审性能实现情况。

(6)评审软件是否具有可修改性、或扩充性、可互换性和可移植性。

(7)评审软件是否具有可测试性。

(8)评审软件是否具有复用性。

程序质量评审的重点在于软件本身的结构、与运行环境的接口、变更带来的影响而进行的评审活动。

第9章　软件测试

9.1　主要内容

本章主要介绍了软件测试的目的和原则。软件测试的技术,主要介绍了黑盒测试和白盒测试的方法。软件测试的过程和策略。软件测试的工具。面向对象的软件测试和软件测试计划与测试分析报告。

9.2　重点难点

理解掌握软件测试的目的和原则。

理解掌握软件测试的技术,主要介绍了黑盒测试和白盒测试的方法。

理解掌握软件测试的过程:单元测试,组装测试,确认测试和系统测试。

理解掌握软件测试的策略。

会使用软件测试的工具。掌握程序静态测试和动态测试的方法。

掌握面向对象的软件测试的技术。

掌握软件测试计划与测试分析报告的书写。

9.3　习题

1. 为了把握软件开发各个环节的正确性和协调性,人们需要进行(A)和(B)工作。(A)的目的是想证实在一给定的外部环境中软件的逻辑正确性。它包括(C)和(D),(B)则试图证明在软件生存期各个阶段,以及阶段间的逻辑(E)、(F)和正确性。

供选择的答案:

A, B. ① 操作　②确认　③验证　④测试　⑤调试

C, D. ①用户的确认　②需求规格说明的确认　③程序的确认　④测试的确认

E, F. ①可靠性　②独立性　③协调性　④完备性　⑤扩充性

2. 测试过程需要三类输入: (A) 、(B) 和(C) 。请选择正确的答案填入图 2. 9 中以完成测试信息处理的全过程。

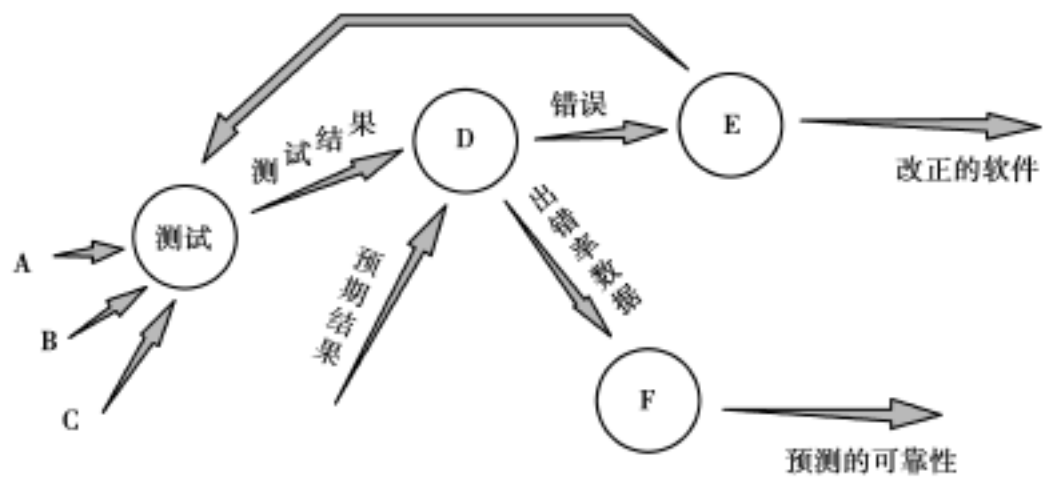


图 2. 9　测试信息处理

供选择的答案:

A ~C: ①接口选择　②软件配置　③硬件配置
④测试配置　⑤测试环境　⑥测试工具

D ~F: ①排错　②可靠性分析　③结果分析　④数据分类

3. 从供选择的答案中选出应填入下面有关软件测试的叙述的(　　) 内的正确答案。

软件测试方法可分为黑盒测试法和白盒测试法两种。

黑盒测试法是通过分析程序的(A) 来设计测试用例的方法。除了测试程序外, 它还适用于对(B) 阶段的软件文档进行测试。

白盒测试法是根据程序的(C) 来设计测试用例的方法。除了测试程序外, 它也适用于对(D) 阶段的软件文档进行测试。

白盒法测试程序时常按照给定的覆盖条件选取测试用例。(E) 覆盖比(F) 覆盖严格, 它使得每一个判定的每一条分支至少经历一次。(G) 覆盖既是判定覆盖, 又是条件覆盖, 但它并不保证使各种条件都能取到所有可能的值。(H) 覆盖比其他条件都要严格, 但它不能保证覆盖程序中的每一条路径。单元测试一般以(I) 为主, 测试的依据是(J) 。供选择的答案:

A, C:　①应用范围　②内部逻辑　③功能　④输入数据

B, D:　①编码　②软件详细设计　③ 软件总体设计　④需求分析

E, F, G, H: ①语句　②判定　③条件　④判定/条件
⑤多重条件　⑥路径

I:　①白盒法　②黑盒法

J:　①模块功能规格说明　② 系统模块结构图　③ 系统需求规格说明

4. 软件测试是软件质量保证的主要手段之一, 测试的费用已超过(A) 的 30% 以上。因此, 提高测试的有效性十分重要。“ 高产 ”的测试是指(B) 。根据国家标准 GB 8566—88《计算机软件开发规范》的规定, 软件的开发和维护划分为 8 个阶段, 其中, 单元测试是在(C) 阶段完

成的，集成测试的计划是在(D) 阶段制定的，确认测试的计划是在(E) 阶段制定的。

供选择的答案:

A. ①软件开发费用　②软件维护费用　③软件开发和维护费用
④软件研制费用　⑤软件生存期全部

B. ①用适量的测试用例运行程序，证明被测程序正确无误
②用适量的测试用例运行程序，证明被测程序符合相应的要求
③用少量的测试用例运行程序，发现被测程序尽可能多的错误
④用少量的测试用例运行程序，纠正被测程序尽可能多的错误

C ～E. ①可行性研究和计划　②需求分析　③概要设计
④详细设计　⑤实现　⑥集成测试
⑦确认测试　⑧使用和维护

5. 请从供选择的答案中选出应填入下列(　　) 中的字句。

程序的 3 种基本控制结构是(A) 。它们的共同点是(B) 。结构化程序设计的一种基本方法是(C) 。软件测试的目的是(D) 。软件调试的目的是(E) 。

供选择的答案:

A. ①过程，子程序，分程序　②顺序，条件，循环
③递归，堆栈，队列　④调用，返回，转移

B. ①不能嵌套使用　②只能用来写简单的程序
③已经用硬件实现　④只有一个入口和一个出口

C. ①筛选法　②递归法　③归纳法　④逐步求精法

D. ①证明程序中没有错误　②发现程序中的错误
③测量程序的动态特性　④检查程序中的语法错误

E. ①找出错误所在并改正之　②排除存在错误的可能性
③对错误性质进行分类　④统计出错的次数

6. 从下列关于软件测试的叙述中，选出 5 条正确的叙述。

(1) 用黑盒法测试时，测试用例是根据程序内部逻辑设计的。

(2) 尽量用公共过程或子程序去代替重复的代码段。

(3) 测试是为了验证该软件已正确地实现了用户的要求。

(4) 对于连锁型分支结构，若有 n 个判定语句，则有 2n 条路径。

(5) 尽量采用复合的条件测试，以避免嵌套的分支结构。

(6) GOTO 语句概念简单，使用方便，在某些情况下，保留 GOTO 语句反而能使写出的程序更加简洁。

(7) 发现错误多的程序模块，残留在模块中的错误也多。

(8) 黑盒测试方法中最有效的是因果图法。

(9) 在做程序的单元测试时，桩(存根) 模块比驱动模块容易编写。

(10) 程序效率的提高主要应通过选择高效的算法来实现。

7. 从供选择的答案中选出同下列关于软件测试的各条叙述关系最密切的字句。

(1) 对可靠性要求很高的软件，例如操作系统，由第三者对源代码进行逐行检查。

(2) 已有的软件被改版时，由于受到变更的影响，改版前正常的功能可能发生异常，性能

也可能下降。因此,对变更的软件进行测试是必要的。

(3)在意识到被测试模块的内部结构或算法的情况下进行测试。

(4)为了确认用户的需求,先做出系统的主要部分,提交给用户试用。

(5)在测试具有层次结构的大型软件时,有一种方法是从上层模块开始,由上到下进行测试。此时,有必要用一些模块替代尚未测试过的下层模块。

供选择的答案:

A～E:①仿真器　②代码审查　③模拟器　④桩　⑤驱动器

⑥域测试　⑦黑盒测试　⑧原型　⑨白盒测试　⑩退化测试

8.软件测试的目的是什么?软件测试中,应注意哪些原则?

9.什么是白盒测试法?有哪些覆盖标准?试对他们的检错能力进行比较?

10.什么是黑盒测试法?采用黑盒技术测试用例有哪几种方法?这些方法各有什么特点?

11.软件测试要经过哪些步骤?这些测试与软件开发各阶段之间有什么关系?

12.单元测试有哪些内容?测试中采用什么方法?

13.什么是集成测试?非渐增式测试与渐增式测试有什么区别?渐增式测试如何组装模块?

14.什么是确认测试?该阶段有哪些工作?

15.调试的目的是什么?调试有哪些技术手段?

16.根据下面给出的规格说明,利用等价类划分的方法,给出足够的测试用例。

“一个程序读入三个整数。把此三个数值看成是一个三角形的三个边。这个程序要打印出信息,说明这个三角形是三边不等的、是等腰的、还是等边的。”

17.有一个处理单价为5角钱的饮料的自动售货机软件测试用例的设计。其规格说明如下:“若投入5角钱或1元钱的硬币,押下 橙汁 或 啤酒 的按钮,则相应的饮料就送出来。若售货机没有零钱找,则一个显示 零钱找完 的红灯亮,这时在投入1元硬币并押下按钮后,饮料不送出来而且1元硬币也退出来;若有零钱找,则显示 零钱找完 的红灯灭,在送出饮料的同时退还5角硬币。”

18.在任何情况下单元测试都是可能的吗?都是需要的吗?

19.对小的程序进行穷举测试是可能的,用穷举测试能否保证程序是百分之百正确呢?

20.应该由谁来进行确认测试?是软件开发者还是软件用户?为什么?

9.4　习题答案

1. A.②　B.③　C.②　D.③　E.③　F.④

2. A.②　B.④　C.⑥　D.③　E.①　F.②

3. A.③　B.④　C.②　D.②　E.②　F.①　G.④　H.⑤　I.①　J.①

4. A.①　B.③　C.⑤　D.③　E.②

5. A.②　B.④　C.④　D.②　E.①

6.正确的叙述有(4)、(5)、(6)、(7)、(10)。

7.(1)②　(2)⑩　(3)⑨　(4)⑧　(5)④

8.软件测试的目的是为了发现软件的错误。

软件测试中应注意的原则有:

(1) 测试用例应由输入数据和预期的输出数据两部分组成。这样便于对照检查, 做到有的放矢。

(2) 测试用例不仅选用合理输入数据, 还要选择不合理的输入数据。这样能更多地发现错误, 提高程序的可靠性。对于不合理的输入数据, 程序应拒绝接受, 并给出相应的提示。

(3) 除了检查程序是否做了它应该做的事, 还应该检查程序是否做了它不应该做的事。

(4) 应制定测试计划并严格执行, 排除随意性。

(5) 长期保留测试用例。

(6) 对发现错误较多的程序段, 应进行更深入的测试。

(7) 程序员应避免测试自己的程序。测试是一种“挑剔性”的行为, 心理状态是测试自己程序的障碍。

9. 白盒法测试法把测试对象看作一个打开的盒子, 测试人员需了解程序内部结构和处理过程, 以检查处理过程的细节为基础, 对程序中尽可能多的逻辑路径进行测试, 检验内部控制结构和数据结构是否有错, 实际的运行状态与预期的状态是否一致。

白盒法有下列几种覆盖标准:

语句覆盖
判定覆盖
条件覆盖
判定/条件覆盖
条件组合覆盖
路径覆盖

从上到下的覆盖标准其检错能力也从弱到强, 其中条件组合发现错误的能力较强, 凡满足其标准的测试用例, 也必然满足前 4 种覆盖标准。在实际的逻辑测试中, 一般以条件组合覆盖为主设计测试用例, 然后再补充部分用例来达到路径覆盖的测试标准。

10. 黑盒测试法把被测试对象看成是一个黑盒子, 测试人员完全不考虑程序的内部结构和处理过程, 只在软件接口处进行测试, 依据需求规格说明书, 检查程序是否满足功能要求。

采用黑盒技术测试用例的方法有: 等价类的划分、边界值分析、错误推测和因果图。

等价类的划分, 是将输入数据按有效的或无效的(也称合理的或不合理的) 划分成若干个等价类, 测试每个等价类的代表值就等于对该类其他值的测试。这样就把漫无边迹的随机测试改为有针对性的等价类测试, 用少量有代表性的例子代替大量测试目的相同的例子, 能有效地提高测试效率。但这个方法的缺点是没有注意选择某些高效的、能够发现更多错误的测试用例。

边界值分析法一般与等价类划分结合起来。但它不是从一个等价类中任选一个例子做代表, 而是将测试边界情况作为重点目标, 选取正好等于、刚刚大于和刚刚小于边界值的测试数据。(边界情况是指输入等价类和输入等价类边界上的情况。) 这种方法可以查出更多的错误, 因为在程序中往往在处理边界情况时易发生错误。

错误推测法是在测试程序时, 人们根据经验或直觉推测程序中可能存在的错误, 从而有针对性地编写检查这些错误的测试用例。

因果图能够有效地检测输入条件的各种组合可能会引起的错误。它的基本原理是通过画因果图,把用自然语言描述的功能说明转换为判定表,最后为判定表的每一列设计一个测试用例。

这几种方法都不能提供一组完整的测试用例,在实际测试中应把各种方法结合起来使用。

综合策略:就是联合使用上述几种测试方法,尽可能多地发现程序中的错误。

11. 软件测试要经过的步骤是:单元测试→集成测试→确认测试→系统测试。

单元测试对源程序中每一个程序单元进行测试,检查各个模块是否正确实现规定的功能,从而发现模块在编码中或算法中的错误。该阶段涉及编码和详细设计文档。

集成测试是为了检查与设计相关的软件体系结构的有关问题,也就是检查概要设计是否合理有效。

确认测试主要是检查已实现的软件是否满足需求规格说明书中确定了的各种需求。

系统测试是把已确认的软件与其他系统元素(如硬件、其他支持软件、数据、人工等)结合在一起进行测试。以确定软件是否可以支付使用。

12. 单元测试主要针对模块的以下 5 个基本特征进行测试:

(1) 模块接口

(2) 局部数据结构

(3) 重要的执行路径

(4) 错误处理

(5) 边界条件

测试的方法是为被测试模块编写驱动模块和桩模块来实现被测试单元的可运行。通过驱动模块来模拟被测试模块的上级调用模块,以上级模块调用被测模块的格式驱动被测模块,接收被测模块的测试结构并输出。桩模块则用来代替被测试模块所调用的模块。它的作用是返回被测模块所需的信息。

13. 集成测试是指在单元测试的基础上,将所有模块按照设计要求组装成一个完整的系统进行的测试。

非渐增式测试是指首先对每个模块分别进行单元测试,再把所有模块组装成一个完整的系统进行的测试。而渐增式测试就是逐个把未经测试的模块组装到已经过测试的模块上去进行集成测试,每加入一个新模块进行一次集成测试,重复此过程直到程序组装完毕。渐增式测试有两种不同的组装方法:自顶向下和自底向上结合。

两者区别是:

(1) 非渐增式方法把单元测试和集成测试分成两个不同的阶段,前一阶段完成模块的单元测试,后一阶段完成集成测试。而渐增式测试往往把单元测试和集成测试合在一起,同时完成。

(2) 非渐增式需要更多的工作量,因为每个模块都需要驱动模块和桩模块,而渐增式利用已测试过的模块作为驱动模块或桩模块,因此工作量少。

(3) 渐增式可以较早地发现接口之间的错误,非渐增式最后组装时才发现。

(4) 渐增式有利于排错,发生错误往往和最近新加入的模块有关,而非渐增式发现接口错误推迟到最后,很难判断是哪一部分接口出错。

(5) 渐增式比较彻底,已测试的模块和新的模块再测试。

(6) 渐增式占用时间较多, 但非渐增式所需更多的驱动模块和桩模块也占用一些时间。

(7) 非渐增式开始可并行测试所有模块, 能充分利用人力, 对测试大型软件很有意义。

14. 确认测试又称有效性测试。它的任务是检查软件的功能与性能是否与需求规格说明书中确定的指标相符合。因而需求说明是确认测试的基础。确认测试阶段有两项工作: 进行确认测试与软件配置审查。

15. 调试则是在进行了成功的测试之后才开始的工作。调试的目的是确定错误的原因和位置, 并改正错误, 因此调试也称为纠错(Debug) 。调试的技术手段有简单的调试方法、归纳法、演绎法和回溯法等。

16. 设三角形的三条边分别为 A, B, C。如果它们能够构成三角形的三条边, 必需满足:

A > 0, B > 0, C > 0, 且 A + B > C, B + C > A, A + C > B。

如果是等腰的, 还要判断是否 A = B, 或 B = C, 或 A = C。

对于等边的, 则需判断是否 A = B, 且 B = C, 且 A = C。

列出等价类表:

输入条件	有效等价类	无效等价类
是否三角形的三条边	(A > 0) (1) , (B > 0) (2) , (C > 0) (3) , (A + B > C) , (4) (B + C > A) (5) , (A + C > B) (6)	A ≤0(7) , B≤0(8) , C≤0 (9) , A + B≤C(10) , A + C≤B(11) , B + C≤A(12)
是否等腰三角形	(A = B) (13) , (B = C) (14) , (A = C) (15)	(A≠B) and(B≠C) and(A≠C) (16)
是否等边三角形	(A = B) and(B = C) and(A = C) (17)	(A≠B) (18) , (B≠C) (19) , (A≠C) (20)

设计测试用例: 输入顺序是 A, B, C

3, 4, 5 覆盖等价类(1) , (2) , (3) , (4) , (5) , (6) 。满足即为一般三角形。

0, 1, 2 覆盖等价类(7) 。不能构成三角形。

1, 0, 2 覆盖等价类(8) 。同上。

1, 2, 0 覆盖等价类(9) 。同上。

} 若不考虑特定 A, B, C, 三者取一即可

1, 2, 3 覆盖等价类(10) 。同上。

1, 3, 2 覆盖等价类(11) 。同上。

3, 1, 2 覆盖等价类(12) 。同上。

} 若不考虑特定 A, B, C, 三者取一即可

3, 3, 4 覆盖等价类(1) , (2) , (3) , (4) , (5) , (6) , (13) 。满足即为等腰三角形

3, 4, 4 覆盖等价类(1) , (2) , (3) , (4) , (5) , (6) , (14) 。若不考虑特定 A, B, C

3, 4, 3 覆盖等价类(1) , (2) , (3) , (4) , (5) , (6) , (15) 。三者取一即可

3, 4, 5 覆盖等价类(1) , (2) , (3) , (4) , (5) , (6) , (16) 。不是等腰三角形

3, 3, 3 覆盖等价类(1) , (2) , (3) , (4) , (5) , (6) , (17) 。是等边三角形

3, 4, 4 覆盖等价类(1) , (2) , (3) , (4) , (5) , (6) , (14) , (18) 。不是等边三角形

3, 4, 3 覆盖等价类(1) , (2) , (3) , (4) , (5) , (6) , (15) , (19) 。若不考虑特定 A, B, C

3, 3, 4 覆盖等价类(1) , (2) , (3) , (4) , (5) , (6) , (13) , (20) 。三者取一即可

17.　分析这一段说明,列出原因和结果

原因:　1. 售货机有零钱找　　3. 投入 5 角硬币　　5. 按下啤酒按钮
　　2. 投入 1 元硬币　　4. 按下橙汁按钮

结果:　21. 售货机 零钱找完 灯亮　24. 送出橙汁饮料
　　22. 退还 1 元硬币　　25. 送出啤酒饮料
　　23. 退还 5 角硬币

画出因果图,如图 2.10 所示。所有原因结点列在左边,所有结果结点列在右边。建立两个中间结点,表示处理的中间状态。

中间结点:　11. 投入 1 元硬币且按下饮料按钮
　　12. 按下 橙汁 或 啤酒 的按钮
　　13. 应当找 5 角零钱并且售货机有零钱找
　　14. 钱已付清

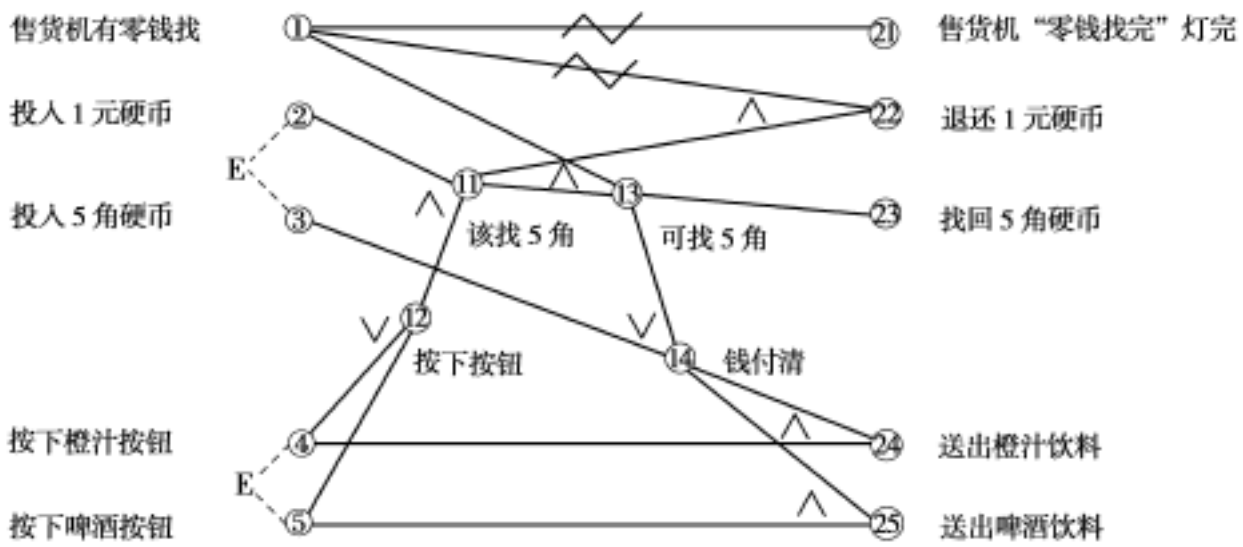


图 2.10　因果图

由于 2 与 3 ,4 与 5 不能同时发生,分别加上约束条件 E。

转换成判定表:

在判定表中,阴影部分表示因违反约束条件的不可能出现的情况,删去。第 16 列与第 32 列因什么动作也没做,也删去。最后可根据剩下的 16 列作为确定测试用例的依据。

因果图方法是一个非常有效的黑盒测试方法,它能够生成没有重复性的且发现错误能力强的测试用例,而且对输入、输出同时进行了分析。

18. 单元测试又称模块测试,是针对软件设计的最小单位——程序模块,进行正确性检验的测试工作。其目的在于发现各模块内部可能存在的各种差错。单元测试需要从程序的内部结构出发设计测试用例。多个模块可以平行地独立进行单元测试。

单元测试是在编码阶段完成的,每编写出一个程序模块,就开始做这个模块的单元测试,所以只要采用模块化方法开发软件,单元测试都是必需的。它可由编写程序的人来完成。因为它需要根据程序的内部结构设计测试用例,对于那些不了解程序内部细节的人,这种测试无法进行。

19. 对小程序进行穷举测试,不一定能保证程序百分之百正确。所谓穷举测试是拿所有可能的输入数据来作为测试用例(黑盒测试) ,或覆盖程序中所有可能的路径(白盒测试) 。对于

序号		1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1	2
条件	①	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	②	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
	③	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
	④	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
	⑤	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
中间结果	⑪						1	1	0		0	0	0		0	0	0						1	1	0		0	0	0		0	0	0
	⑫						1	1	0		1	1	0		1	1	0						1	1	0		1	1	0		1	1	0
	⑬						1	1	0		0	0	0		0	0	0						0	0	0		0	0	0		0	0	0
	⑭						1	1	0		1	1	1		0	0	0						0	0	0		1	1	1		0	0	0
结果	㉑						0	0	0		0	0	0		0	0	0						1	1	1		1	1	1		1	1	1
	㉒						0	0	0		0	0	0		0	0	0						1	1	0		0	0	0		0	0	0
	㉓						1	1	0		0	0	0		0	0	0						0	0	0		0	0	0		0	0	0
	㉔						1	0	0		1	0	0		0	0	0						0	0	0		1	0	0		0	0	0
	㉕						0	1	0		0	1	0		0	0	0						0	0	0		0	1	0		0	0	0
测试用例							Y	Y	Y		Y	Y	Y		Y	Y							Y	Y	Y		Y	Y	Y		Y	Y	

图 2.11　由因果图得到的判定表

小程序来说,实际上并不能真正做到穷举测试。例如前面讲过,一个小程序 P 只有两个输入 X 和 Y 及输出 Z,在字长为 32 位的计算机上运行。如果 X、Y 只取整数,考虑把所有的 X、Y 值都作为测试数据,按黑盒方法进行穷举测试,这样做可能采用的测试数据组(X_i, Y_i) ,基数(radix) i 的最大可能数目为:$2^{32} \times 2^{32} = 2^{64}$。如果程序 P 测试一组 X、Y 数据需要 1 毫秒,而且假定一天工作 24 小时,一年工作 365 天,要完成 2^{64} 组测试,需要 5 亿年。

20. 在对照需求做有效性测试和软件配置审查时,是由软件开发者在开发环境下进行的测试。而接下来做验收测试时则以用户为主。软件开发人员和 QA(质量保证) 人员也应参加。由用户参加设计测试用例,使用用户界面输入测试数据,并分析测试的输出结果。一般使用生产中的实际数据进行测试。

如果软件是为多个客户开发的,则需要进行 α测试和 β测试。α测试是由一个用户在开发环境下进行的测试,也可以是公司内部的用户在模拟实际操作环境下进行的测试。软件在一个自然设置状态下使用。开发者坐在用户旁边,随时记下错误情况和使用中的问题。这是在受控制的环境下进行的测试。

β测试是由软件的多个用户在一个或多个用户的实际使用环境下进行的测试。这些用户是与公司签订了支持产品预发行合同的外部客户,他们要求使用该产品,并愿意返回有关错位错误信息给开发者。与 α测试不同的是,开发者通常不在测试现场。因而,β测试是在开发者无法控制的环境下进行的软件现场应用。

第 10 章　软件维护

10.1　主要内容

本章主要介绍了软件维护的定义、分类和特点。软件维护的过程及组织。可维护性的概念和可维护性的度量,维护的副作用。软件逆向工程与再工程的概念。

10.2　重点难点

理解软件维护定义。
了解软件维护的类型与特点。
理解软件维护的过程。
掌握可维护性的概念。
理解软件逆向工程与再工程的概念。

10.3　习题

1. 对于软件产品来说, 有 4 个方面影响着产品的质量, 即(A) 、(B) 、(C) 及成本、时间和进度等条件。重视软件过程的质量是近年来质量管理理论和实践的新发展。重视软件过程质量的控制, 其部分原因可能是: 相对于产品质量的控制来说, 过程质量的控制是(D) 、(E) 、(F) , 而产品质量的控制是(G) 、(H) 、(I) 。

供选择的答案:

A ~C.	①开发时间	②开发技术	③过程质量	④风险控制
	⑤质量控制	⑥人员素质	⑦项目管理	⑧配置管理
D ~I.	①主动的	②被动的	③整体的	④系统的
	⑤先期的	⑥事后的	⑦个别的	⑧部分的

2. 在软件维护的实施过程中, 为了正确、有效地修改, 需要经历以下 3 个步骤: (A) 、(B) 、(C) 。(A) 是决定维护成败和质量好坏的关键。(C) 包括(D) 确认、计算机确认和维护后的(E) 。

供选择的答案:

A ~C.	①修改程序	②建立目标程序	③分析和理解程序	
	④重新验证程序	⑤验收程序		
D.	①动态	②静态	③人工	④自动
E.	①验证	②验收	③检验	④存档

3. 从供选择的答案中选出同下列各叙述关系最密切的字句。

A. 软件从一个计算机系统或环境转移到另一个计算系统或环境的容易程度。
B. 软件在需要它投入使用时能实现其指定的功能的概率。
C. 软件使不同的系统约束条件和用户需求得到满足的容易程度。
D. 在规定的条件下和规定的一段期间内, 实现所指定的功能的概率。
E. 尽管有不合法的输入, 软件仍能继续正常工作的能力。

供选择的答案:

①可测试性	②可理解性	③可靠性	④可移植性
⑤可使用性	⑥兼容性	⑦容错性	⑧可修改性
⑨可接近性	⑩一致性		

4. 从下列叙述中选出 5 条与提高软件的可移植性有关的叙述。

①把程序中与计算机硬件特性有关的部分集成在一起。
②选择时间效率和空间效率高的算法。

③使用结构化的程序设计方法。

④尽量用高级语言编写程序中对效率要求不高的部分。

⑤尽可能减少注释。

⑥采用表格控制方式。

⑦文档资料详尽、正确。

⑧在有虚拟存储器的计算机系统上开发软件。

⑨减少程序中对文件的读写次数。

⑩充分利用宿主计算机的硬件特性。

5. 软件可移植性是用来衡量软件的(A) 的重要尺度之一。为了提高软件的可移植性, 应注意提高软件的(B) 。采用(C) 有助于提高(B) 。为了提高可移植性, 还应(D) 。使用(E) 语言开发的系统软件具有较好的可移植性。

供选择的答案:

A.	①通用性	②效率	③质量	④人机界面
B.	①使用的方便性	②简洁性	③可靠性	④设备独立性
C.	①优化算法	②专用设备	③表格驱动方式	④树型文件目录
D.	①有完备的文件资料		② 选择好的宿主计算机	
	③ 减少输入输出次数		④ 选择好的操作系统	
E.	①COBOL	②APL	③C	④SQL

6. 下面有关软件维护的叙述有些是不准确的, 请将它们列举出来。

供选择的答案:

①要维护一个软件, 必须先理解这个软件。

②阅读别人写的程序并不困难。

③如果文档不齐全也可以维护一个软件。

④谁写的软件就得由谁来维护这个软件。

⑤设计软件时就应考虑到将来的可修改性。

⑥维护软件是一件很吸引人的创造性工作。

⑦维护软件就是改正软件中的错误。

⑧维护好一个软件是一件很难的事情。

7. 软件再工程是一类软件工程活动, 它能够使我们: ⅰ) 增进对软件的理解; ⅱ) 准备或直接提高软件自身的(A) 、(B) 或演化性。第ⅱ部分旨在改善软件的(C) , 使得软件更容易为人们服务。纯粹是出于改善性能的代码优化(D) 软件再工程。逆向工程属于上述软件再工程的第(E) 部分。

供选择的答案:

A, B.	①可靠性	②灵活性	③可维护性	④可复用性	⑤可修改性
C.	①静态质量	②动态质量	③性能	④功能	
D.	①属于	②不属于			
E.	①ⅱ	② ⅰ			

8. 一个软件产品开发完成投入使用后, 常常由于各种原因需要对它做适当的变更。在软件的使用过程中, 软件原来的(A) 可能不再适应用户的要求, 需要进行变更; 软件的工作环境

也可能发生变化,最常见的是配合软件工作的(B) 有变动;还有一种情况是在软件使用过程中发现错误,需要进行修正。通常把软件交付使用后做的变更称为(C) 。软件投入使用后的另一项工作是(D) ,针对这类软件实施的软件工程活动,主要是对其重新实现,使其具有更好的(E) ,包括软件重构、重写文档等。(D) 和新的软件开发工作的主要差别在于(H) 。我们把常规的软件开发称为(F) ,而(G) 是从代码开始推导出设计或是规格说明来的。

供选择的答案:

A, B.	①环境	②软件	③硬件	④功能和性能	⑤要求
C, D, F, G.	①逆向工程	②正向工程	③软件再工程	④维护	⑤设计
E.	①可靠性	②可维护性	③可移植性	④可修改性	
H.	①使用的工具不同		②开发的过程不同		
	③开发的起点不同		④要求不同		

9. 软件可维护性是指纠正软件系统出现的错误和缺陷,以及为满足新的要求进行修改,(A) 的容易程度。目前广泛使用 7 个特性来衡量软件的可维护性,其中就有(B) 、(C) 、(D) 。其中,(B) 和(D) 主要在改正性维护中侧重应用,(C) 主要在适应性维护和(E) 维护中侧重应用。

供选择的答案:

A.	①维护	②扩充与压缩	③调整	④再工程
B ~D.	①安全性	②可靠性	③完整性	④适应性
	⑤可理解性	⑥可使用性	⑦一致性	⑧数据无关性
E.	①预防性	②完善性	③改正性	④容错性

10. McCall 提出了表明软件质量的 11 个质量特性。它们是(A) 、(B) 、(C) 、(D) 、(E) 、(F) 、(G) 、(H) 、效率、可测试性和互连性。我们把这 11 个特性分为 3 组,使其分别隶属于产品修正、产品转移和产品运行等 3 个方面,如图 2. 12 所示。

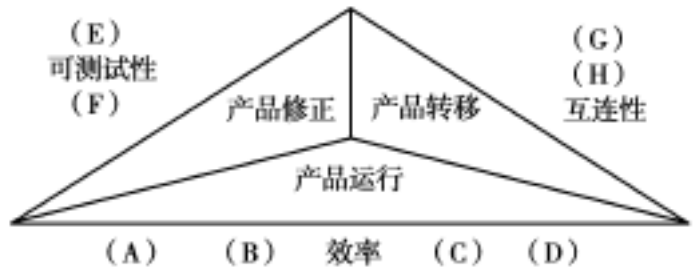


图 2. 12

供选择的答案:

A ~H.	①可读性	②正确性	③功能性	④完整性
	⑤可靠性	⑥可移植性	⑦可复用性	⑧灵活性
	⑨可维护性	⑩可使用性		

11. 关于软件再工程的定义有这样两种说法。ⅰ) 软件再工程是变更系统(或程序) 的(A) ,或是系统(或程序) 的(B) ,而不变更其(C) 的一种工程活动。ⅱ) 检查并改进对象系统,按新的模式对系统进行(D) ,进而实现其新的模式。

多数软件再工程工具可按图示的自动进行再工程的模式工作。请选择合适的答案完成图 2. 13。

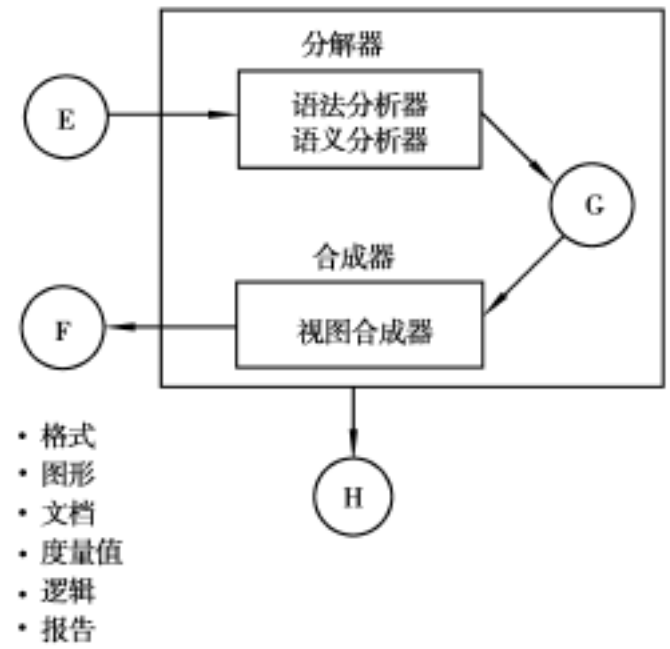


图 2.13

供选择的答案:

A ~D. ①外部环境　②接口　③内部机制　④流程图
⑤重构　⑥数据结构　⑦功能性　⑧层次性

E ~H. ①其他产品　②软件工具　③信息库　④软件
⑤软件的新视图

12. (A) 是软件的一种表达形式, 或是有关软件的一种报告。(A) 可分为 4 类: 软件的规格说明属于(B), PDL 或 DFD 属于(C), 规格说明文本的图索引属于(D), 程序和程序段属于(E)。(A) 得到工具支持时, 就成为(F), 可用其支持加入、变更或浏览信息。

供选择的答案:

A, F. ①软件说明　②概要设计　③软件视图　④信息库
⑤视图编辑器　⑥软件设计

B ~E. ①A 类视图　②1 类视图(非过程性描述和/或元描述)
③2 类视图(伪过程性描述和/或面向体系结构的描述)
④3 类视图(纯过程性描述或直接导出的信息)

13. 为什么软件需要维护? 维护有哪几种类型? 简述它们的维护过程。

14. 改错性维护与“排错”是否是一回事? 为什么?

15. 什么是程序修改的副作用? 程序修改的副作用有哪几种? 试举例说明?

16. 在软件计划中是否应该把维护费用计划在内? 实际情况如何?

17. 软件维护困难主要表现在什么方面?

10.4 习题答案

1. A. ②,　B. ③,　C. ⑥,　D. ⑤,　E. ①,　F. ④,　G. ⑥,　H. ②,　I. ⑦

其中, A、B、C 答案顺序可互换, D、E、F 答案顺序可互换, G、H、I 答案顺序可互换。

2. A. ③,　B. ①,　C. ④,　D. ②,　E. ②

3. A. ④,　B. ⑤,　C. ⑥,　D. ③,　E. ⑦

论述 A 是指可移植性。可移植性的定义是: 将一个软件系统从一个计算机系统或环境移植到另一个计算机系统或环境中运行时所需工作量的大小。

论述 B 是指可使用性。可使用性的定义是: 程序方便、实用及易于使用的程度。用户一有请求, 就能对每一个操作方式作出解释, 始终如一地按照用户的要求运行。计算其按用户请求实现指定功能的概率, 是一种度量准则。

论述 C 是指兼容性。有两类基本的兼容性: 向下兼容和交错兼容。向下兼容是软件新版本保留它早期版本的功能的情况; 交错兼容是共同存在的两个相关但不同的产品之间的兼容性。软件可以在不同系统约束和不同用户需求下完成指定的工作。

论述 D 是指可靠性。可靠性的定义是: 一个程序按照用户的要求和设计目标, 在给定的一段时间内正确执行的概率。

论述 E 是指容错性。容错性的定义是: 系统出错(机器临时发生故障或数据输入不合理) 时, 能以某种预定方式, 作出适当处理, 得以继续执行和恢复系统的能力。

4. 正确的叙述有　①、③、④、⑥、⑦。

为了提高软件的可移植性, 应当尽可能用高级语言编写源程序代码。对于与硬件或操作系统有关的部分, 或对效率要求很高的部分, 应当为它们建立专门的模块, 将用汇编语言写的程序封装在这些模块中, 与程序中其他部分以事先约定的标准方式接口。这样, 一旦硬件环境或操作系统环境发生变化, 只需修改个别模块即可。

采用表格控制方式, 将所有的外部设备接口或与其他系统的接口, 包括信息传递、驱动程序入口等都用表格控制, 即使将来硬件、相关软件发生的变化, 只需修改表格中的登记项, 原来的程序一律可以不改。

为了将来修改方便, 不至于引入新的错误, 相关文档一定要齐全、正确, 程序中必须有必要的注释, 并使用如结构化程序设计方法这样的良好的程序设计方法来编写程序。至于算法选择, 与效率有关, 与可移植性无关。其他叙述, 如⑧、⑨、⑩, 都不利于可移植性。

5. A. ③,　B. ④,　C. ③,　D. ①,　E. ③

6. 软件维护人员通常不是改软件的开发人员, 这给软件维护带来很大的困难。特别是有些软件在开发时没有遵循软件开发的准则, 没有开发方法的支持, 维护这样的软件就更困难。下面列举一些与软件维护有关的问题。

①要维护一个软件, 首先必须要理解它。而理解一个别人编写的程序通常是很困难的, 尤其是对软件配置(指各种相关的文档) 不齐全的软件, 理解起来就更加困难。

②需要维护的软件往往缺少合格的文档, 或者文档资料不齐全, 甚至根本没有文档。在软维护中, 合格的文档十分重要, 它有助于理解被维护的软件。合格的文档不仅要完整正确地反映开发过程各阶段的工作成果, 而且应当容易理解并应与程序源代码一致。而错误的文档会把对软件的理解引入歧途。

③在软件维护时, 不要指望得到原来开发该软件的人员的帮助。开发人员开发完一个软件后, 往往会从事另一软件的开发, 甚至已离开原开发单位。即使原来的开发人员还在, 也可能时间太久而忘却了实现的细节。

④多数软件在设计时没有考虑到将来的修改, 这给软件的修改造成了困难。而且在修改软件时很可能引入新的差错。

⑤软件维护通常不是一件吸引人的工作。从事维护工作常使维护人员缺乏成就感, 这也

严重影响维护工作,从而影响了维护质量的提高。

7. A. ③, B. ④, C. ①, D. ②, E. ②。其中,A、B 答案的顺序可互换。

8. A. ④, B. ③, C. ④, D. ③, E. ②, F. ②, G. ①, H. ③

9. A. ②, B. ②, C. ⑥, D. ⑤, E. ②。其中,B、D 的答案顺序可互换。

10. A. ②, B. ⑤, C. ⑩, D. ④, E. ⑨, F. ⑧, G. ⑥, H. ⑦

其中,A、B、C、D 答案顺序可互换,E、F 答案顺序可互换,G、H 答案顺序可互换。

11. A. ③, B. ⑥, C. ⑦, D. ⑤, E. ④, F. ⑤, G. ③, H. ①。其中,A、B 的答案顺序可互换;

12. A. ③, B. ②, C. ③, D. ①, E. ④, F. ⑤

13. 在软件开发完成交付用户使用后,为了保证软件在一个相当长的时期能够正常运行,就需要对软件进行维护。

软件维护的类型有 4 种:改正性维护、适应性维护、完善性维护和预防性维护。其中,改正性维护是要改正在特定的使用条件下暴露出来的一些潜在程序错误或设计缺陷;适应性维护是要在软件使用过程中数据环境发生变化或处理环境发生变化时修改软件以适应这种变化;完善性维护是在用户和数据处理人员使用软件过程中提出改进现有功能,增加新的功能,以及改善总体性能的要求后,修改软件以把这些要求纳入到软件之中。

由这些原因引起的维护活动可以归为以下几类:预防性维护是为了提高软件的可维护性、可靠性等,事先采用先进的软件工程方法对需要维护的软件或软件中的某一部分(重新)进行设计、编制和测试,为以后进一步改进软件打下良好基础。

软件维护的过程如图 2.14 所示。第一步是先确认维护要求。这需要维护人员与用户反复协商,弄清错误概况以及对业务的影响大小,以及用户希望做什么样的修改,并把这些情况存入故障数据库。然后由维护组织管理员确认维护类型。

对于改正性维护申请,从评价错误的严重性开始。如果存在严重的错误,则必须安排人员,在系统监督员的指导下,进行问题分析,寻找错误发生的原因,进行“救火”性的紧急维护;对于不严重的错误,可根据任务、机时情况、视轻重缓急,进行排队,统一安排时间。对于适应性维护和完善性维护申请,需要先确定每项申请的优先次序。若某项申请的优先级非常高,就可立即开始维护工作,否则,维护申请和其他的开发工作一样,进行排队,统一安排时间。并不是所有的完善性维护申请都必须承担,因为进行完善性维护等于是做二次开发,工作量很大,所以需要根据商业需要、可利用资源的情况、目前和将来软件的发展方向以及其他的考虑,决定是否承担。

尽管维护申请的类型不同,但都要进行同样的技术工作。这些工作有:修改软件需求说明、修改软件设计、设计评审、对源程序做必要的修改、单元测试、集成测试(回归测试)、确认测试、软件配置评审等。在每次软件维护任务完成后,最好进行一次情况评审,对以下问题做一总结:

在目前情况下,设计、编码、测试中的哪一方面可以改进?

哪些维护资源应该有但没有?

工作中主要的或次要的障碍是什么?

从维护申请的类型来看是否应当有预防性维护?

情况评审对将来的维护工作如何进行会产生重要的影响,并可为软件机构的有效管理提

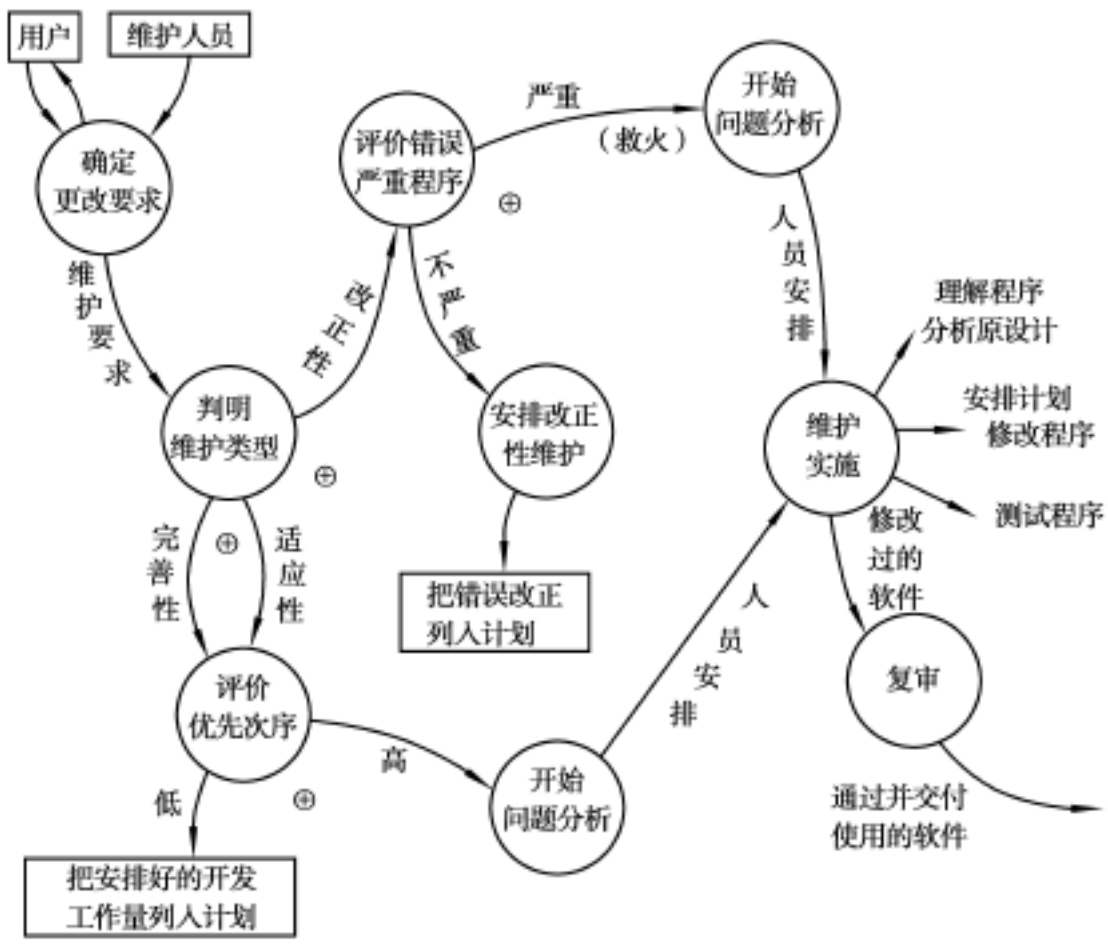


图 2.14　软件维护的过程

供重要的反馈信息。

14. 改错性维护与“排错(调试)”不是一个概念。调试是作为测试的后继工作而出现的，是当测试发现软件中的错误后，进一步诊断和改正程序中潜在的错误的活动。而改正性维护是指在软件交付使用后，由于开发时测试的不彻底、不完全，必然会有一部分隐藏的错误被带到运行阶段来，这些隐藏下来的错误在某些特定的使用环境下就会暴露出来。为了识别和纠正软件错误、改正软件性能上的缺陷、排除实施中的误使用所进行的诊断和改正错误的过程。调试在程序编码阶段、测试阶段、运行和维护阶段都可以发挥作用，它实际上是一种工具或手段。在软件交付运行之后，用户实际充当了测试员的角色，一旦发现软件运行中的错误或缺陷，就会将问题报告通报软件销售商，申请软件维护。其后软件维护人员可以利用调试手段来诊断和改正软件中存在的错误。这时可能涉及的范围不只包括程序，还有文档和数据，不仅可能修改程序代码，而且可能需要修改设计，甚至需求。所以改正性维护是在更大范围中做工作。

15. 所谓副作用是指因修改软件而造成的错误或其他不希望发生的情况，有以下 3 种副作用：

①修改代码的副作用：在使用程序设计语言修改源代码时，都可能引入错误。例如，删除或修改一个子程序、删除或修改一个标号、删除或修改一个标识符、改变程序代码的时序关系、改变占用存储的大小、改变逻辑运算符、修改文件的打开或关闭、改进程序的执行效率以及把设计上的改变翻译成代码的改变、为边界条件的逻辑测试做出改变时，都容易引入错误。

②修改数据的副作用：在修改数据结构时，有可能造成软件设计与数据结构不匹配，因而导致软件出错。数据副作用就是修改软件信息结构导致的结果。例如，在重新定义局部的或全局的常量、重新定义记录或文件的格式、增大或减小一个数组或高层数据结构的大小、修改

全局或公共数据、重新初始化控制标志或指针、重新排列输入/输出或子程序的参数时,容易导致设计与数据不相容的错误。

③文档的副作用:对数据流、软件结构、模块逻辑或任何其他有关特性进行修改时,必须对相关技术文档进行相应修改。否则会导致文档与程序功能不匹配,缺省条件改变,新错误信息不正确等错误。使得软件文档不能反映软件的当前状态。如果对可执行软件的修改不反映在文档里,就会产生文档的副作用。例如,对交互输入的顺序或格式进行修改,如果没有正确地记入文档中,就可能引起重大的问题。过时的文档内容、索引和文本可能造成冲突,引起用户的失败和不满。

16. 针对不同的工作目标,软件计划的可以有以下多种类型:

①项目实施计划(或称为软件开发计划)——这是软件开发的综合性计划,通常应包括任务、进度、人力、环境、资源、组织等多个方面。

②质量保证计划——把软件开发的质量要求具体规定为每个开发阶段可以检查的质量保证活动。

③软件测试计划——规定测试活动的任务、测试方法、进度、资源、人员职责等。

④文档编制计划——规定所开发项目应编制的文档种类、内容、进度、人员职责等。

⑤用户培训计划——规定对用户进行培训的目标、要求、进度、人员职责等。

⑥综合支持计划——规定软件开发过程中所需要的支持,以及如何获取和利用这些支持。

⑦软件分发计划——软件开发项目完成后,如何提供给用户。

并未专门考虑软件的维护费用问题。但实际上,为了提高软件的可维护性,在软件开发的具体操作时,必须考虑将来的维护,采取必要的措施,以降低将来维护的费用。

17. (1)一般来讲,维护人员对开发人员写的程序及文档,理解都比较困难,对维护工作不会喜欢;

(2)维护持续时间都很长,在开发人员不在现场的轻快下,维护软件通常是很困难的;

(3)绝大多数软件在设计时对将来的软件修改都没有考虑或考虑不多,尤其未能在设计中强调并认真解决好模块的独立性,使软件的修改既困难又易发生差错。

第11章　软件项目管理

11.1　主要内容

本章主要介绍了软件项目管理的特点和功能。软件项目的工作要求。软件和硬件资源的确定,人员的计划和组织。软件成本估算的概念,COCOMO成本估算方法。进度计划,软件配置管理,软件知识产权。

11.2　重点难点

掌握软件项目的概念、特点和功能。

理解软件项目的工作要求。

了解软件和硬件资源的确定。

理解掌握人员的计划和组织。
理解软件成本估算的概念,掌握 COCOMO 成本估算方法。
理解掌握进度计划安排,如何进行软件配置管理。
理解软件知识产权的概念。

11.3　习题

1.软件项目管理的主要职能包括:(A) ,建立组织,配备人员,(B) 和(C) 。由于软件项目的特有性质,使得项目管理存在一定困难。第一、(D) ,软件工程过程充满了大量高强度的脑力劳动;第二、(E) ,在特定机型上,利用特定的硬件配置,由特定的系统软件和支撑软件支持,形成了特定的开发环境;第三、(F) ,软件项目经历的各个阶段都深透了大量的手工劳动,远未达到自动化的程度;第四、(G) ,用户要经过专门的培训,才能掌握操作步骤,且需要配备专职维护人员进行售后服务;第五、(H) ,为高质量地完成软件项目,充分发掘人员的智力才能和创造精神。

在总结和分析足够数量失误的软件项目之后可知,造成软件失误的原因大多与(I) 工作有关。在软件项目开始执行时,执行的过程中及项目进行的最后阶段都会遇到种种问题。

供选择的答案:

A ~C. ①编码　②制定计划　③开发　④指导
⑤测试　⑥检验

D ~H. ①软件工作渗透了人的因素　②智力密集,可见性差
③单件生产　④使用方法繁琐,维护困难
⑤劳动密集,自动化程度低

I. ①设计　②维护　③测试　④管理
⑤实践　⑥指导　⑦审核　⑧分析

2.软件开发小组的目的是发挥集体的力量进行软件研制。因此,小组从培养(A) 的观点出发进行程序设计消除软件的(B) 的性质。通常,程序设计小组的组织形式有三种,如下图 2.15所示的 a 属于(C) ,b 属于(D) ,c 属于(E) 。

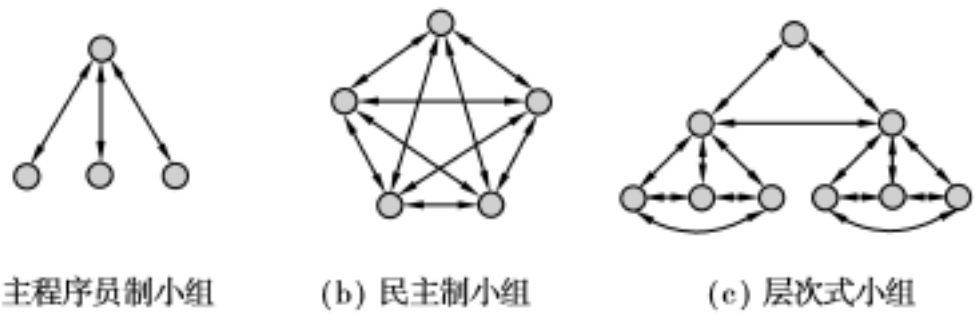


图 2.15

供选择的答案:

A, B. ①“局部”　②“全局”　③“集体”　④“个人”

C ~E. ①层次式小组　②民主制小组　③主程序员制小组

3.对于一个小型的软件开发项目,一个人就可以完成需求分析、设计、编码和测试工作。但随着软件项目规模增大,需要有多人共同参与同一软件项目的工作。当几个人共同承担软件开发项目中的某一任务时,人与人之间必须通过交流来解决各自承担任务之间的(A) 问题,

即通信问题。通信需花费时间和代价,会引起软件错误(B) , (C) 软件生产率。如果一个软件开发小组有 n 个人,每两人之间都需要通信,则共有(D) 条通信路径。假设一个人单独开发软件,生产率是 5000 行/人年,且在每条通信路径上耗费的工作量是 250 行/人年。若 4 个人组成一个小组共同开发这个软件,则小组中每个人的软件生产率为(E) 。若小组有 6 名成员,则小组中每个成员的软件生产率为(F) 。因此,有人提出,软件开发小组的规模不能太大,人数不能太多,一般在(G) 人左右为宜。

供选择的答案:

A.	①分配	②管理	③接口	④协作
B, C.	①降低	②增加	③不变	
D.	①n(n +1) /2	②n(n - 1) /2	③n(n - 1) (n - 2) /6	④n2 /2
E, F.	①4 875	②4 375	③4 625	④5 735
G.	①8 ~15	②1 ~2	③2 ~5	④2 ~8

4. 估算资源、成本和进度时需要经验、有用的历史信息、足够的定量数据和作定量度量的勇气。通常估算本身带有(A) 。项目的复杂性越高,规模越大,开发工作量(B) ,估算的(A) 就(C) 。项目的结构化程度提高,进行精确估算的能力就能(D) ,而风险将(E) 。有用的历史信息(F) ,总的风险会减少。

供选择的答案:

A.	①风范(范型)	②风格	③风险	④度量
B ~F.	①增加	②越多	③降低	④不变
	⑤越少	⑥越高	⑦越大	

5. 在考虑各种软件开发资源时,(A) 是最重要的资源。如果把软件开发所需的资源画成一个金字塔形: 在塔的上层是最基本的资源(A) ,在底部为(B) 。(B) 包括硬件资源和软件资源。(C) 、(D) 和其他硬件设备属于硬件资源。IPSE 工具属于软件资源中的(E) 。为了提高软件的生产率和软件产品的质量,可建立(F) 。

供选择的答案:

A, B.	①方法	②人力	③工具	④上下文环境
C, D.	①虚拟机	②目标机	③自动机	④宿主机
E, F.	①维护工具	②分析设计工具	③支持工具	④编程工具
	⑤可复用构件库	⑥框架工具	⑦原型化模拟工具	

6. 任何软件项目都必须做好项目管理工作,最常使用的进度管理工具是(A) ,当某一开发项目的进度有可能拖延时,应该(B) 。对于一个典型的软件开发项目,各开发阶段需投入的工作量的百分比大致是(C) 。各阶段所需不同层次的技术人员大致是(D) ,而管理人员在各阶段所需数量也不同,相对而言大致是(E) 。

供选择的答案:

A.	① 数据流图	②程序结构图	③因果图	④PERT 图
B.	①增加新的开发人员		② 分析拖期原因加以补救	
	③从别的小组抽调人员临时帮忙		④推迟预定完成时间	

C.

		需求分析	设　　计	编　　码	测　　试
投入工作量	①	25	25	25	25
	②	10	20	30	40
	③	15	30	15	40
	④	5	10	65	30

D.

技术人员水平	①	初级	高级	高级	高级
	②	中级	中级	高级	中级
	③	高级	中高级	初级	中高级
	④	中级	中高级	中级	初级

E.

管理人员数量	①	多	中	少	中
	②	中	中	中	中
	③	多	少	多	多
	④	少	多	少	多

7. 什么是知识产权?

8. 什么是软件配置管理? 什么是基线?

9. 为什么在软件开发中, 不能用简单增加人员的方法来缩短开发时间?

10. 软件工程管理包括哪些内容?

11. 软件项目计划中包括哪些内容?

12. 软件开发成本估算方法有哪几种?

11.4　习题答案

1. A. ②,　B. ④,　C. ⑥,　D. ②,　E. ③,　F. ⑤,　G. ④,　H. ①,　I. ④

2. A. ②,　B. ④,　C. ③,　D. ②,　E. ①

3. A. ③,　B. ②,　C. ①,　D. ②,　E. ③,　F. ②,　G. ④

对于一个小型的软件开发项目, 一个人就可以完成需求分析、设计、编码和测试工作。但是, 随着软件开发项目规模的增大, 就会有更多的人共同参与同一软件项目的工作。例如 10 个人 1 年可以完成的项目, 若让 1 个人干 10 年是不行的。因此, 需要多人组成开发小组共同参加一个项目的开发。但是, 当几个人共同承担软件开发项目中的某一任务时, 人与人之间必须通过交流来解决各自承担任务之间的接口问题, 即所谓通信问题。通信需花费时间和代价, 会引起软件错误增加, 降低软件生产率。

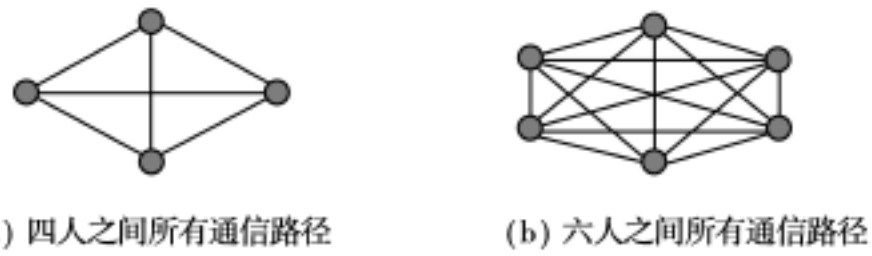
若两个人之间需要通信, 则称在这两个人之间存在一条通信路径。如果一个软件开发小组有 n 个人, 每两人之间都需要通信, 则总的通信路径有 n(n - 1) /2 条。

假设一个人单独开发软件, 生产率是 5 000 行/人年。若 4 个人组成一个小组共同开发这个软件, 则需要 6 条通信路径(如图 2.16(a)) 。若在每条通信路径上耗费的工作量是 250 行/人年。则小组中每个人的软件生产率降低为

$$5\,000 - 6\times 250/4 = 5\,000 - 375 = 4\,625 \text{ 行/人年。}$$

如果小组有 6 名成员,通信路径增加到 15 条(如图 2.16(b)) 。每条通信路径消耗的工作量不变,则小组中每个成员的软件生产率降低为

5 000 - 15× 250 /6 = 5 000 - 625 = 4 375 行 /人年。



(a) 四人之间所有通信路径　　(b) 六人之间所有通信路径

图 2.16

从上述简单分析可知,一个软件任务由一个人单独开发,生产率最高;而对于一个稍大型的软件项目,一个人单独开发,时间太长。因此软件开发小组是必要的。有人提出,软件开发小组的规模不能太大,人数不能太多,一般在 2 ～8 人左右为宜。

4. A. ③,　B. ②,　C. ⑦,　D. ①,　E. ③,　F. ②

5. A. ②,　B. ③,　C. ②,　D. ④,　E. ⑥,　F. ⑤其中,C、D 的答案顺序可互换。

6. A. ④,　B. ②,　C. ③,　D. ③,　E. ①

7. 知识产权就是人们对自己的智力劳动成果所依法享有的权利,是一种无形的财产。它分为工业产权和版权两大类。工业产权包括专利权、商标权等。

8. 软件配置管理,简称 SCM(Software Configuration Management) ,是指一组管理整个软件生存期各阶段中变更的活动。软件配置管理技术可以使软件变更所产生的错误达到最小并最有效地提高生产率。

基线:是软件生存期中各开发阶段的一个特定点,它的作用是把开发各阶段工作的划分更加明确化,使本来连续的工作在这些点上断开,以便于检查与肯定阶段成果。

9. 大量软件开发实践说明:向一个已经延迟的项目追加开发人员,可能使它完成得更晚。因为当开发人员以算术级数增长时,而人员之间的通信将以几何级数增长,往往“ 得不偿失 ”。

10. 软件工程管理的具体内容包括对开发人员、组织机构、用户、文档资料等方面的管理。

11. 软件项目计划内容包括:

(1) 范围

(2) 资源

(3) 进度安排

(4) 成本估算

(5) 培训计划

12. 软件开发成本估算的方法主要有:

(1) 自顶向下估算方法

(2) 自底向上估算方法

(3) 差别估算方法

以及专家估算法、类推估算法、算式估算法等几类方法。

第 12 章　软件工程标准与软件文档

12.1　主要内容

本章主要介绍了软件工程的标准,软件工程的标准的分层,国家标准。软件质量的认证。软件文档的定义、作用、分类,对软件文档编制的质量要求。

12.2　重点难点

掌握软件工程的标准的分层,国家标准。
理解 ISO 9000 系列标准的特点和标准概要。
理解掌握软件文档的定义作用、分类。
理解掌握对软件文档编制的质量要求。

12.3　习题

1.(A)是计算机软件机构实施 ISO 9001 的指南性标准。它的指南性主要表现在:ⅰ)对于针对的标准给予特定的(B),ISO 9001 提供了(C)个质量体系要素。ⅱ)指南性的标准不是(D)的依据。在 ISO 9001 的质量体系要素的每一条都强调了要求的强制性,而在(A)中是建议性指南。由于 ISO 9001 标准本来是针对传统的制造业制定的,而软件业又有许多不同于制造业的特性,所以,(A)起了桥梁作用。

此外,(E)将整个软件生存期分为 17 个过程,并且对每一个过程按(F)的三个层次具体做了解释,为我们进一步理解(A)提供了帮助。

供选择的答案:

A, E.	①ISO 9001	②ISO 9002	③ISO 9000—3	④ISO/IEC 9126
	⑤ISO 8402	⑥ISO/IEC 12207		
B, D.	①说明与解释	②强制性	③建议性	④认证审核
C.	①10	②20	③25	④30
F.	①“任务—活动—过程”		②“活动—任务—过程”	
	③“过程—活动—任务”		④“任务—过程—活动”	

2. ISO 9000 族标准是指国际标准化组织中的质量管理和质量保证技术委员会(ISO／TC176)制定的标准,现有(A)个标准,可分为 5 类:质量术语标准,如(B)。(C),如 ISO 9001、ISO 9002、ISO 9003 系列标准。(D),如 ISO 9004 系列标准。(E),如 ISO 9000 系列标准。(F),如 ISO 10005 质量计划指南,ISO 10007 技术状态管理指南等。

供选择的答案:

A.	①10	②25	③20	④30
B.	①ISO 9000—3	②ISO 9004—4	③ISO 9001	④ISO 8402
C ～F.	①支持性技术标准		②质量管理和质量保证标准的选用和实施指南	
	③质量保证标准		④质量管理标准	⑤质量改进标准

3. 基于软件人员能力成熟度模型的评估的依据是成熟度级别上的(A) 。一个企业的成熟度级别是所有(A) 被实施的(B) 级别。软件企业通过评估结果可以了解(A) 方面的强项和弱项, 明确努力的方向。P-CMM 模型并不禁止处于较低成熟度级别上的企业在需要的时候实施高一级关键过程域的内容。但是, 由于每一级都是向上一级进化的基础, 所以, 跳越成熟度级别的进化是(C) 。基于 P-CMM 的改进工作和其他任何发展项目一样, 第一要有(D) , 第二要(E) , 第三要有专人负责。卡内基—梅隆大学 SEI 的 P-CMM 顾问委员会建议: 在软件工程小组内加入人力资源管理人员, 以进行基于 P-CMM 的改进工作。这样, 带给软件机构管理层的信息就是: 一个致力于改善整个软件过程能力的项目强调的是(F) , 缺一不可。

供选择的答案:

A, B.	①最高	②最低	③中间	④关键过程域
	⑤实践	⑥软件过程改进		
C.	①无风险的	②有风险的	③盲目的	④可预测的
D, E.	①方法	②计划	③跟踪其发展	④加速其进程
F.	①过程、技术和人员		②人力、软件和硬件	③目标、方法和过程
	④计划、测试和维护		⑤过程、评估和检验	

4. 有哪些软件工程标准?

5. 什么是文档?

6. 请说明软件文档的作用?

7. 软件开发项目生存期各阶段包含哪些文档?

8. 软件工程标准化的意义是什么?

12.4 习题答案

1. A. ③, B. ①, C. ②, D. ④, E. ⑥, F. ③

2. A. ③, B. ④, C. ③, D. ④, E. ②, F. ①

3. A. ④, B. ②, C. ②, D. ②, E. ③, F. ①

4. 软件工程标准根据其制定机构与适用范围, 可分为:

①国际标准

②国家标准

③行业标准

④企业规范

⑤项目(课题) 规范

5. 文档是指某种数据媒体和其中所记录的数据。它具有永久性, 并可由人或机器阅读, 通常仅用于人工可读的东西。在软件工程中, 文档常常用来对活动、需求、过程或结果进行描述、定义、规定、报告或认证的任何书面或图示的信息。它们描述和规定了软件设计和实现的细节, 说明使用软件的操作命令。文档是软件产品的一部分, 没有文档的软件就不成其为软件。

6. 软件文档的作用是:

提高软件开发过程的能见度;

提高开发效率;

作为开发人员阶段工作成果和结束标志;

记录开发过程的有关信息便于使用与维护;
提供软件运行、维护和培训有关资料;
便于用户了解软件功能、性能。

7. 软件开发项目生存期各阶段应包括的文档包括如下:
可行性研究报告
项目开发计划
软件需求说明书
数据要求说明书
测试计划
概要设计说明书
详细设计说明书
用户手册
操作手册
测试分析报告
开发进度月报
项目开发总结
程序维护手册

8. 软件工程标准化可以为软件工程各个阶段的活动提供统一的行动规范和衡量准则, 使得各种工作都能有章可循。

第 13 章　软件工程环境与发展

13.1　主要内容

本章主要介绍了软件工程环境的概念、结构、应用技术。软件工程的发展, 软件集成化技术和软件智能化技术。

13.2　重点难点

理解软件开发环境的概念;
理解掌握软件集成化技术。

13.3　习题

1. 什么是软件开发环境? 请列出其发展情况。
2. 请叙述软件开发环境的分类。
3. 何谓软件工具? 通常包含哪几部分?
4. 请叙述集成化 CASE 的五级模型。
5. CASE 工作台有哪些分类?

13.4 习题答案

1.软件开发环境是指在计算机的基本软件的基础上,为支持软件的开发而提供的一组工具软件系统。具体地说,软件开发环境是一组相关工具的集合。这些相关工具按一定的开发方法或一定开发处理模型组织起来的,这些相关工具支持整个软件生存期的各阶段或部分阶段。

软件开发环境的发展阶段:

(1)20 世纪 70 年代,软件开发与设计方法出现了结构化分析技术的一整套相互衔接的 SA-SD 的方法学。与此相适应的计算机辅助软件工程技术则主要由开发孤立的软件工具而逐步向程序设计环境的开发和使用方向发展,出现了第一代的基于正文的 CASE 工具。

(2)20 世纪 80 年代中期与后期,主要是实时系统设计方法,以及面向对象的分析和设计方法的发展,它克服了结构化技术的缺点。在这期间开发了第二代的 CASE 工具,其特点是支持使用图形表示的结构化方法,如数据流图与结构图。其开发环境表现在提高环境中工具的集成性方面,如"集成的项目支持环境"。

(3)20 世纪 80 年代后期和 20 世纪 90 年代初期出现了"基于信息工程 CASE"技术,这种环境集成了用于项目计划、分析、设计、编程、测试和维护的一个工具箱的集合。

(4)20 世纪 90 年代出现一系列集成的 CASE 软件产品,用以实现需求管理、应用程序分析设计和建模、编码、软件质量保证和测试、过程和项目管理及文档生成管理等软件开发工作的规范化、工程化和自动化。

2.软件开发环境可按解决的问题、软件开发环境的演变趋向和集成化程度进行分类:

(1)按解决的问题可分为:程序设计环境、系统合成环境、项目管理环境三类。

(2)按软件开发环境的演变趋向可分为:以语言为中心的环境、工具箱环境和基于方法的环境三类。

(3)按集成化程度有第一代、第二代、第三代的开发环境。

3.软件工具的定义是:可用来帮助和支持软件需求分析、软件开发、测试、维护、模拟、移植或管理而编制的计算机程序或软件。软件工具是一个程序系统。

软件工具通常由工具、工具接口和工具用户接口三部分构成。

4.集成化 CASE 的五级模型包括

(1)平台集成:工具运行在相同的硬件/操作系统平台上

(2)数据集成:工具使用共享数据模型来操作

(3)表示集成:工具使用相同的用户界面

(4)控制集成:工具激活后能控制其他操作

(5)过程集成:工具在一个过程模型和"过程机"的指导下使用

5.CASE 工具台是一组工具集,支持像设计、实现或测试等特定的软件开发阶段。CASE 工具组装成一个工具台后工具能协同工作,可提供比单一工具更好的支持。

CASE 工作台包括:

(1)程序设计工作台。由支持程序设计的一组工具组成。

(2)分析和设计工作台。支持软件过程的分析和设计阶段。

(3)测试工作台。

(4) 交叉开发工作台。这些工作台支持在一种机器上开发软件, 而在其他的系统上运行所开发的软件。

(5) 配置管理(CM) 工作台。这些工作台支持配置管理。

(6) 文档工作台。这些工具支持高质量文档的制作。

(7) 项目管理工作台。支持项目管理活动。

附　录

附录 A　软件工程实验编写报告格式

A.1　可行性研究报告

1. 引言

(1) 编写目的

【阐明编写可行性研究报告的目的, 指明读者对象。】

(2) 项目背景

【应包括:

所建议开发软件的名称;

项目的任务提出者、开发者、用户及实现软件的单位;

项目与其他软件或其他系统的关系。】

(3) 定义

【列出文档中所用到的专门术语的定义和缩写词的原文。】

(4) 参考资料

【列出有关资料的作者、标题、编号、发表日期、出版单位或资料来源, 可包括:

项目经核准的计划任务书、合同或上级机关的批文;

与项目有关的已发表的资料;

文档中所引用的资料, 所采用的软件标准或规范。】

2. 可行性研究的前提

(1) 要求

【列出并说明建议开发软件的基本要求, 如

功能;

性能;

输出;
输入;
基本的数据流程和处理流程;
安全与保密要求;
与软件相关的其他系统;
完成期限。】
(2) 目标
【可包括:
人力与设备费用的节省;
处理速度的提高;
控制精度或生产能力的提高;
管理信息服务的改进;
决策系统的改进;
人员工作效率的提高, 等等。】
(3) 条件、假定和限制
【可包括:
建议开发软件运行的最短寿命;
进行系统方案选择比较的期限;
经费来源和使用限制;
法律和政策方面的限制;
硬件、软件、运行环境和开发环境的条件和限制;
可利用的信息和资源;
建议开发软件投入使用的最迟时间。】
(4) 可行性研究方法
(5) 决定可行性的主要因素
3. 对现有系统的分析
(1) 处理流程和数据流程
(2) 工作负荷
(3) 费用支出
【如人力、设备、空间、支持性服务、材料等项开支。】
(4) 人员
【列出所需人员的专业技术类别和数量。】
(5) 设备
(6) 局限性
【说明现有系统存在的问题以及为什么需要开发新的系统。】
4. 所建议技术可行性分析
(1) 对系统的简要描述
(2) 处理流程和数据流程

(3) 与现有系统比较的优越性

(4) 采用建议系统可能带来的影响

①对设备的影响

②对现有软件的影响

③对用户的影响

④对系统运行的影响

⑤对开发环境的影响

⑥对运行环境的影响

⑦对经费支出的影响

(5) 技术可行性评价

【包括:

在限制条件下, 功能目标是否能达到;

利用现有技术, 功能目标能否达到;

对开发人员数量的和质量的要求, 并说明能否满足;

在规定的期限内, 开发能否完成。】

5. 所建议系统经济可行性分析

(1) 支出

①基建投资

②其他一次性支出

③经常性支出

(2) 效益

①一次性收益

②经常性收益

③不可定量收益

(3) 收益/投资比

(4) 投资回收周期

(5) 敏感性分析

【敏感性分析是指一些关键性因素, 如: 系统生存周期长短、系统工作负荷量、处理速度要求、设备和软件配置变化对支出和效益的影响等的分析。】

6. 社会因素可行性分析

(1) 法律因素

【如, 合同责任、侵犯专利权、侵犯版权等问题的分析。】

(2) 用户使用可行性

【如, 用户单位的行政管理、工作制度、人员素质等能否满足要求。】

7. 其他可供选择的方案

【逐个阐明其他可供选择的方案, 并重点说明未被推荐的理由。】

8. 结论意见

【结论意见可能是:

可着手组织开发;
需待若干条件(如资金、人力、设备等)具备后才能开发;
需对开发目标进行某些修改;
不能进行或不必进行(如技术不成熟,经济上不合算等);
其他。】

A.2　项目开发计划

1. 引言

(1)编写目的

【阐明编写开发计划的目的,指明读者对象。】

(2)项目背景

【可包括:
项目的委托单位、开发单位和主管部门;
该软件系统与其他系统的关系。】

(3)定义

【列出本档中用到的专门术语的定义和缩写词的原文。】

(4)参考资料

【可包括:
项目经核准的计划任务书、合同或上级机关的批文;
文档所引用的资料、规范等;列出这些资料的作者、标题、编号、发表日期、出版单位或资料来源。】

2. 项目概述

(1)工作内容

【简要说明项目的各项主要工作,介绍所开发软件的功能、性能等。若不编写可行性研究报告,则应在本节给出较详细的介绍。】

(2)条件与限制

【阐明为完成项目应具备的条件、开发单位已具备的条件以及尚需创造的条件。必要时还应说明用户及分合同承包者承担的工作、完成期限及其他条件与限制。】

(3)产品

①程序

【列出应交付的程序名称、使用的语言及存储形式。】

②文档

【列出应交付的文档。】

(4)运行环境

【应包括硬件环境、软件环境。】

(5)服务

【阐明开发单位可向用户提供的服务。如人员培训、安装、保修、维护和其他运行支持。】

(6) 验收标准

3. 实施计划

(1) 任务分解

【任务的划分及各项任务的负责人。】

(2) 进度

【按阶段完成的项目, 用图表说明开始时间、完成时间。】

(3) 预算

(4) 关键问题

【说明可能影响项目的关键问题, 如设备条件、技术焦点或其他风险因素, 并说明对策。】

4. 人员组织及分工

5. 交付期限

6. 专题计划要点

【如测试计划、质量保证计划、配置管理计划、人员培训计划、系统安装计划等。】

A.3 需求规格说明书

1. 引言

(1) 编写目的

【阐明编写需求说明书的目的, 指明读者对象。】

为明确软件需求、安排项目规划与进度、组织软件开发与测试, 撰写本文档。

本文档供项目经理、设计人员、开发人员参考。

(2) 项目背景

项目的委托单位、开发单位和主管部门

该软件系统与其他

(3) 定义

【列出文档中所用到的专门术语的定义和缩写词的原文。】

(4) 参考资料

项目经核准的计划任务书、合同或上级机关的批文

项目开发计划

文档所引用的资料、标准和规范。列出这些资料的作者、标题、编号、发表日期、出版单位或资料来源

2. 任务概述

(1) 目标

(2) 运行环境

操作系统: Microsoft Windows 2000 Advanced Server

支持环境: IIS 5.0

数 据 库: Microsoft SQL Server 2000

(3) 条件与限制

3. 数据描述

(1) 静态数据

(2) 动态数据

【包括输入数据和输出数据。】

(3) 数据库介绍

【给出使用数据库的名称和类型。】

(4) 数据词典

(5) 数据采集

4. 功能需求

(1) 功能划分

(2) 功能描述

5. 性能需求

(1) 数据精确度

(2) 时间特性

【如响应时间、更新处理时间、数据转换与传输时间、运行时间等。】

(3) 适应性

【在操作方式、运行环境与其他软件的接口以及开发计划等发生变化时，应具有的适应能力。】

6. 运行需求

(1) 用户界面

【如屏幕格式、报表格式、菜单格式、输入输出时间等。】

(2) 硬件接口

(3) 软件接口

(4) 故障处理

7. 其他需求

【如可使用性、安全保密、可维护性、可移植性等。】

A.4　概要设计说明书

1. 引言

(1) 编写目的

【阐明编写概要设计说明书的目的，指明读者对象。】

(2) 项目背景

【应包括:

项目的委托单位、开发单位和主管部门;

该软件系统与其他系统的关系。】

(3) 定义

【列出本文档中所用到的专门术语的定义和缩写词的原文。】

(4)参考资料

【列出有关资料的作者、标题、编号、发表日期、出版单位或资料来源,可包括:
项目经核准的计划任务书、合同或上级机关的批文;
项目开发计划;
需求规格说明书;
测试计划(初稿);
用户操作手册(初稿);
文档所引用的资料、采用的标准或规范。】

2. 任务概述

(1)目标

(2)运行环境

(3)需求概述

(4)条件与限制

3. 总体设计

(1)处理流程

(2)总体结构和模块外部设计

(3)功能分配

【表明各项功能与程序结构的关系。】

4. 接口设计

(1)外部接口

【包括用户界面、软件接口与硬件接口。】

(2)内部接口

【模块之间的接口。】

5. 数据结构设计

(1)逻辑结构设计

(2)物理结构设计

(3)数据结构与程序的关系

6. 运行设计

(1)运行模块的组合

(2)运行控制

(3)运行时间

7. 出错处理设计

(1)出错输出信息

(2)出错处理对策

【如设置后备、性能降级、恢复及再启动等。】

8. 安全保密设计

9. 维护设计

【说明为方便维护工作的设施,如维护模块等。】

A.5　详细设计说明书

1. 引言

(1)编写目的

【阐明编写详细设计说明书的目的,指明读者对象。】

(2)项目背景

【应包括项目的来源和主管部门等。】

(3)定义

【列出文档中所用到的专门术语的定义和缩写词的原文。】

(4)参考资料

【列出有关资料的作者、标题、编号、发表日期、出版单位或资料来源,可包括:

项目的计划任务书、合同或批文;

项目开发计划;

需求规格说明书;

概要设计说明书;

测试计划(初稿);

用户操作手册(初稿);

文档中所引用的其他资料、软件开发标准或规范。】

2. 总体设计

(1)需求概述

(2)软件结构

【如给出软件系统的结构图。】

3. 程序描述

【逐个模块给出以下的说明:】

(1)功能

(2)性能

(3)输入项目

(4)输出项目

(5)算法

【模块所选用的算法。】

(6)程序逻辑

【详细描述模块实现的算法,可采用:

标准流程图;

PDL 语言;

N-S 图;

PAD;
判定表等描述算法的图表。】
(7) 接口
(8) 存储分配
(9) 限制条件
(10) 测试要点
【给出测试模块的主要测试要求。】

A.6　用户操作手册

1. 引言
(1) 编写目的
【阐明编写手册的目的, 指明读者对象。】
(2) 项目背景
【应包括项目的来源、委托单位、开发单位和主管部门。】
(3) 定义
【列出手册中所用到的专门术语的定义和缩写词的原文。】
(4) 参考资料
【列出有关资料的作者、标题、编号、发表日期、出版单位或资料来源, 可包括:
项目的计划任务书、合同或批文;
项目开发计划;
需求规格说明书;
概要设计说明书;
详细设计说明书;
测试计划;
手册中引用的其他资料、采用的软件工程标准或软件工程规范。】
2. 软件概述
(1) 目标
(2) 功能
(3) 性能
数据精确度【包括输入、输出及处理数据的精度。】
时间特性【如响应时间、处理时间、数据传输时间等。】
灵活性【在操作方式、运行环境需做某些变更时软件的适应能力。】
3. 运行环境
(1) 硬件
【列出软件系统运行时所需的硬件最小配置, 如
计算机型号、主存容量;
外存储器、媒体、记录格式、设备型号及数量;
输入、输出设备;
数据传输设备及数据转换设备的型号及数量。】

(2) 支持软件

【如:

操作系统名称及版本号;

语言编译系统或汇编系统的名称及版本号;

数据库管理系统的名称及版本号;

其他必要的支持软件。】

4. 使用说明

(1) 安装和初始化

【给出程序的存储形式、操作命令、反馈信息及其含意、表明安装完成的测试实例以及安装所需的软件工具等。】

(2) 输入

【给出输入数据或参数的要求。】

①数据背景

【说明数据来源、存储媒体、出现频度、限制和质量管理等。】

②数据格式

【如:

长度;

格式基准;

标号;

顺序;

分隔符;

词汇表;

省略和重复;

控制。】

③输入举例

(3) 输出

【给出每项输出数据的说明。】

①数据背景

【说明输出数据的去向、使用频度、存放媒体及质量管理等。】

②数据格式

【详细阐明每一输出数据的格式, 如: 首部、主体和尾部的具体形式。】

③举例

(4) 出错和恢复

【给出:

出错信息及其含意;

用户应采取的措施, 如修改、恢复、再启动。】

(5) 求助查询

【说明如何操作。】

5. 运行说明

(1) 运行表

【列出每种可能的运行情况, 说明其运行目的。】

(2) 运行步骤

【按顺序说明每种运行的步骤, 应包括:】

①运行控制

②操作信息

运行目的;

操作要求;

启动方法;

预计运行时间;

操作命令格式及说明;

其他事项。

③输入/输出文件

【给出建立或更新文件的有关信息, 如:

文件的名称及编号;

记录媒体;

存留的目录;

文件的支配说明确定保留文件或废弃文件的准则, 分发文件的对象, 占用硬件的优先级及保密控制等。】

④启动或恢复过程

6. 非常规过程

【提供应急或非常规操作的必要信息及操作步骤, 如出错处理操作、向后备系统切换操作以及维护人员须知的操作和注意事项。】

7. 操作命令一览表

【按字母顺序逐个列出全部操作命令的格式、功能及参数说明。】

8. 程序文件(或命令文件) 和数据文件一览表

【按文件名字母顺序或按功能与模块分类顺序逐个列出文件名称、标识符及说明。】

9. 用户操作举例

A.7 测试计划

1. 引言

(1) 编写目的

【阐明编写测试计划的目的, 指明读者对象。】

(2) 项目背景

【说明项目的来源、委托单位及主管部门。】

(3) 定义

【列出测试计划中所用到的专门术语的定义和缩写词的原意。】

(4)参考资料

【列出有关资料的作者、标题、编号、发表日期、出版单位或资料来源,可包括:
项目的计划任务书、合同或批文;
项目开发计划;
需求规格说明书;
概要设计说明书;
详细设计说明书;
用户操作手册;
本测试计划中引用的其他资料、采用的软件开发标准或规范。】

2. 任务概述

(1)目标

(2)运行环境

(3)需求概述

(4)条件与限制

3. 计划

(1)测试方案

【说明确定测试方法和选取测试用例的原则。】

(2)测试项目

【列出组装测试和确认测试中每一项测试的内容、名称、目的和进度。】

(3)测试准备

(4)测试机构及人员

【测试机构名称、负责人和职责。】

4. 测试项目说明

【按顺序逐个对测试项目做出说明:】

(1)测试项目名称及测试内容

(2)测试用例

①输入

【输入的数据和输入命令。】

②输出

【预期的输出数据。】

③步骤及操作

④允许偏差

【给出实测结果与预期结果之间允许偏差的范围。】

(3)进度

(4)条件

【给出测试对资源的特殊要求,如设备、软件、人员等。】

(5) 测试资料

【说明测试所需的资料。】

5. 评价

(1) 范围

【说明所完成的各项测试说明问题的范围及其局限性。】

(2) 准则

【说明评价测试结果的准则。】

A.8 测试分析报告

1. 引言

(1) 编写目的

【阐明编写测试分析报告的目的, 指明读者对象。】

(2) 项目背景

【说明项目的来源、委托单位及主管部门。】

(3) 定义

【列出测试分析报告中所用到的专门术语的定义和缩写词的原文。】

(4) 参考资料

【列出有关资料的作者、标题、编号、发表日期、出版单位或资料来源, 可包括:
项目的计划任务书、合同或批文;
项目开发计划;
需求规格说明书;
概要设计说明书;
详细设计说明书;
用户操作手册;
测试计划;
测试分析报告所引用的其他资料、采用的软件工程标准或软件工作规范。】

2. 测试计划执行情况

(1) 测试项目

【列出每一项测试项目的名称、内容和目的。】

(2) 测试机构和人员

【给出测试机构名称、负责人和参与测试人员名单。】

(3) 测试结果

【按顺序给出每一测试项目的:
实测结果数据;
与预期结果数据的偏差;
该项测试表明的事实;
该项测试发现的问题。】

3. 软件需求测试结论

【按顺序给出每一项需求测试的结论。包括:

证实的软件能力;
局限性(即项目需求未得到充分测试的情况及原因)。】

4. 评价

(1) 软件能力

【经过测试所表明的软件能力。】

(2) 缺陷和限制

【说明测试所揭露的软件缺陷和不足,以及可能给软件运行带来的影响。】

(3) 建议

【提出为弥补上述缺陷的建议。】

(4) 测试结论

【说明能否通过。】

A.9　项目开发总结报告

1. 引言

(1) 编写目的

【阐明编写总结报告的目的,指明读者对象。】

(2) 项目背景

【说明项目来源、委托单位、开发单位及主管部门。】

(3) 定义

【列出报告用到的专门术语的定义和缩写词的原文。】

(4) 参考资料

【列出有关资料的作者、标题、编号、发表日期、出版单位或资料来源,可包括:
项目经核准的计划任务书、合同或上级机关的批文;
项目开发计划;
需求规格说明书;
概要设计说明书;
详细设计说明书;
用户操作手册;
测试计划;
测试分析报告;
本报告引用的其他资料、采用的开发标准或开发规范。】

2. 开发结果

(1) 产品

【可包括:
列出各部分的程序名称、源程序行数(包括注释行)或目标程序字节数及程序总计数量、存储形式;
产品文档名称等。】

(2) 主要功能及性能

(3)所用工时

【按人员的不同层次分别计时。】

(4)所用机时

【按所用计算机机型分别计时。】

(5)进度

【给出计划进度与实际进度的对比。】

(6)费用

3. 评价

(1)生产率评价

【如平均每人每月生产的源程序行数、文档的字数等。】

(2)技术方案评价

(3)产品质量评价

4. 经验与教训

A.10　程序维护手册

1. 引言

(1)编写目的

【阐明编写手册的目的,指明读者对象。】

(2)开发单位

【说明项目的提出者、开发者、用户和使用场所。】

(3)定义

【列出报告中所用到的专门术语的定义和缩写词的原文。】

(4)参考资料

【列出有关资料的作者、标题、编号、发表日期、出版单位或资料来源,以及保密级别,可包括:

用户操作手册;

与本项目有关的其他文档。】

2. 系统说明

(1)系统用途

【说明系统具备的功能,输入和输出。】

(2)安全保密

【说明系统安全保密方面的考虑。】

(3)总体说明

【说明系统的总体功能,对系统、子系统和作业做出综合性的介绍,并用图表的方式给出系统主要部分的内部关系。】

(4)程序说明

【说明系统中每一程序、分程序的细节和特性。】

①程序 1 的说明

a. 功能

【说明程序的功能。】

b. 方法

【说明实现方法。】

c. 输入

【说明程序的输入、媒体、运行数据记录、运行开始时使用的输入数据的类型和存放单元、与程序初始化有关的入口要求。】

d. 处理

【处理特点和目的, 如:

用图表说明程序中的运行逻辑流程;

程序主要转移条件;

对程序的约束条件;

程序结束时的出口要求;

与下一个程序的通信与联结(运行、控制) ;

由该程序产生并供处理程序段使用的输出数据类型和存放单元;

程序运行所用存储量、类型及存储位置等。】

e. 输出

【程序的输出。】

f. 接口

【本程序与本系统其他部分的接口。】

g. 表格

【说明程序内部的各种表、项的细节和特性。对每张表的说明至少包括:

表的标识符;

使用目的;

使用此表的其他程序;

逻辑划分, 如块或部, 不包括表项;

表的基本结构;

设计安排, 包括表的控制信息。表目结构细节、使用中的特有性质及各表项的标识、位置、用途、类型、编码表示。】

h. 特有的运行性质

【说明在用户操作手册中没有提到的运行性质。】

②程序 2 的说明

【与程序 1 的说明相同。以后其他各程序的说明相同。】

3. 操作环境

(1) 设备

【逐项说明系统的设备配置及其特性。】

(2) 支持软件

【列出系统使用的支持软件, 包括它们的名称和版本号。】

(3) 数据库

【说明每个数据库的性质和内容, 包括安全考虑。】

①总体特征

【如:

标识符;

使用这些数据库的程序;

静态数据;

动态数据;

数据库的存储媒体;

程序使用数据库的限制。】

②结构及详细说明

a. 说明该数据库的结构, 包括其中的记录和项;

b. 说明记录的组成, 包括首部或控制段、记录体;

c. 说明每个记录结构的字段, 包括: 标记或标号、字段的字符长度和位数、该字段的允许值范围;

d. 扩充: 说明为记录追加字段的规定。

4. 维护过程

(1) 约定

【列出该软件系统设计中所使用全部规则和约定, 包括:

程序、分程序、记录、字段和存储区的标识或标号助记符的使用规则;

图表的处理标准、卡片的连接顺序、语句和记号中使用的缩写、出现在图表中的符号名;

使用的软件技术标准;

标准化的数据元素及其特征。】

(2) 验证过程

【说明一个程序段修改后, 对其进行验证的要求和过程(包括测试程序和数据) 及程序周期性验证的过程。】

(3) 出错及纠正方法

【列出出错状态及其纠正方法。】

(4) 专门维护过程

【说明文档其他地方没有提到的专门维护过程, 如:

维护该软件系统的输入输出部分(如数据库) 的要求、过程和验证方法;

运行程序库维护系统所必需的要求、过程和验证方法;

对闰年、世纪变更所需的临时性修改等。】

(5) 专用维护程序

【列出维护软件系统使用的后备技术和专用程序(如文件恢复程序、淘汰过时文件的程序等) 的目录, 并加以说明, 内容包括:

维护作业的输入输出要求;
输入的详细过程及在硬设备上建立、运行并完成维护作业的操作步骤。】
(6) 程序清单和流程图
【引用资料或提供附录给出程序清单和流程图。】

附录 B　参考作业

B.1　网上书店 My-eBookStore 介绍

假设个体书店店主某某委托计算机专业的大学毕业生组成的开发小组(5 人) 为他创建网上书店系统 My-eBookStore, 以便能够扩展书店的客户群、科学管理、提高效益。该书店以经营英语、计算机书籍为主。在网上书店建立初期, 要求在确保基本功能正常的情况下, 尽量简化, 并且在 3 个月内完成。

网站主要提供的基本服务项目有: 用户的注册, 登录; 用户的分级浏览或图书选购; 店主对进书、售书、库存、账目、客户的管理; 以及网站的日常维护(比如, 网上书店简介; 网上书店信息发布; 客户留言及对客户留言的反馈)。

如果可能, 店主某某还希望利用电子商务突出自己网上书店的特色和提高书店的经营效益。

要求网页能够提供两级图书目录和三层信息(一级目录是图书的基本分类目录; 其下是图书的二级目录, 它对应于某基本分类之下的书名及其简要信息; 当客户点击二级目录中的某本书之后, 系统应当显示详细介绍该书的文字与图形信息)。

一般客户可以浏览网上书店内容。欲购书的客户需要注册(提供邮购和管理所必需的有效信息, 如姓名、地址、电话等) 取得惟一的用户名成为会员。会员登录后便可以购书一本或多本。店主在客户确认网上订书单后的 7 日内收到其足额购书汇款单后的 10 日内根据与客户的约定时间送书到客户手中, 同时应收到有客户签名的送书单回执。

店主应该能够对网上书店的进书、售书订单、库存、账目(比如, 日结账, 日销售额与赢利额) 客户信息进行查询和管理。

为了便于测试所开发的 My-eBookStore 系统, 系统开发小组应该输入各 30 册以上的英语图书和计算机图书到系统中。

假设网上书店系统运行在 Windows 2000 平台之上, 所选用的数据库是 Access 或 SQL Server; 交互网页技术可以采用 ASP 技术(微软方案) 或 JSP 技术(Java 方案) 或 PHP 技术; 软件文档的开发和编制可以采用 Visio、Rose 等工具。

由于大学生开发小组对于所涉及到的学科知识和开发技术与工具并不完全熟悉, 因此他们需要边学边干, 并且采用原型法进行有效的团队开发。

要求: 学生开发小组对上述客户需求仔细研究、分析, 同时参考网站开发项目 2、项目 3 介绍中有关网站开发的功能与技术指标, 并且在考察同类著名网站功能和设计特色的基础上, 拟定出本小组要开发网站系统的问题定义与网站原型基本功能与特色、开发技术与工具、设计初步方案、开发计划与成员分工等文档。然后再按照本课程设计指导书的各项要求进行设计、开发、测试与文档编制和总结。项目的分析与设计任务可以采用传统的结构化分析与设计方法,

也可以采用面向对象要分析与设计方法(如用 UML 工具 Rose)。

B.2 创业网站 My-eCompany 介绍

假设由擅长于计算机平面设计、英汉翻译、Java 编程与网络应用开发的几名大学毕业生创办了一个小的创业公司从事承接上述专业任务及其相关信息咨询业务。创业初期的任务是开发该公司的创业网站 My-Company,以便向公众展示公司的宗旨、业务(典型业务与报价)、构成、专长与特色、成果及联系方式。创业网站的功能除了可以参考项目 2 中所述的网上书店 My-eBookStore 之外,还可以参考以下一般企业网站具有的基本功能。在创业网站建立初期,同样要求在确保基本功能正常的情况下,尽量简化,并且在 3 个月内完成。

一般企业网站基础方案

项目	服务内容	具体描述	备注
独立域名	域名注册	国际顶级域名一个	1 个
网站建设	主机空间共享	40 G 使用空间	Windows2000 平台
	主页设计(中文版或英文版)	根据企业特点选择设计主页	企业提供有关资料
	精美网页制作	10 个中文或英文精美页面(企业任选)	标准 A4 页面
	产品库图文并茂	15 个产品图片和详细文字说明产品库发布	企业提供产品图片
	商情展示专用窗口	各种供求信息分类发布、高级搜索引擎	自主操作,不限数量
	电子邮箱	1 个以企业域名为后缀的电子邮箱* * * @name. com	空间为 5 M
	网站计数器	记录客户访问数量	1 个
	客户留言板	方便客户与企业之间沟通	1 个
	新闻发布系统	企业随时添加、修改、删除公司的新闻动态	
	信息反馈单	及时得到商业信息、实现企业在线订购	1 个
	商务办公室	网上商务办公、信息交互传递、更新网站内容	1 个
	共享数据库	多重发布、多次链接的信息服务	
网站维护			

假设创业网站 My-Company 系统运行在 Windows 2000 平台之上, 所选用的数据库是 Access 或 SQL Server; 交互网页技术可以采用 ASP 技术(微软方案) 或 JSP 技术(Java 方案) 或 PHP 技术; 软件文档的开发和编制可以采用 Visio、Rose 等工具。由于大学生开发小组对于所涉及到的学科知识和开发技术与工具并不完全熟悉, 因此他们需要边学边干, 并且采用原型法进行有效的团队开发。

要求: 学生开发小组对上述客户需求仔细研究、分析, 同时参考网站开发项目 1、项目 3 介绍中有关网站开发的功能与技术指标, 并且在考察同类著名网站功能和设计特色的基础上, 拟定出本小组要开发网站系统的问题定义与网站原型基本功能与特色、开发技术与工具、设计初步方案、开发计划与成员分工等文档。然后再按照本课程设计指导书的各项要求进行设计、开发、测试与文档编制和总结。项目的分析与设计任务可以采用传统的结构化分析与设计方法, 也可以采用面向对象要分析与设计方法(如用 UML 工具 Rose) 。

B. 3　政府机构网站 Our-eOrganization 介绍

假设由擅长计算机网站开发的几名大学毕业生创办了一个小的创业公司, 为了培养队伍, 他们主动为政府的某个机构开发一个政府机构实验网站 Our-eOrganization, 网站内容主要包括该政府机构的如下内容: 政务公开; 机构概览; 网上办公; 网上监督; 公众反馈; 机构特色内容与特色功能; 网站的日常服务与安全。在考虑该网站的内容、功能和技术指标时还可以参看有关我国城市政府网站的以下三个表(摘自《计算机世界》34 期 2002 年 9 月 9 日 A24-A26) 。由于是实验网站, 在网站开发初期, 要求在确保基本功能正常的情况下, 尽量简化, 并且在 2 至 3 个月内完成。

假设政府机构实验网站 Our-eOrganization 系统运行在 Windows 2000 平台之上, 所选用的数据库是 Access 或 SQL Server; 交互网页技术可以采用 ASP 技术(微软方案) 或 JSP 技术(Java 方案) 或 PHP 技术; 软件文档的开发和编制可以采用 Visio、Rose 等工具。由于大学生开发小组对于所涉及到的学科知识和开发技术与工具并不完全熟悉, 因此他们需要边学边干, 并且采用原型法进行有效的团队开发。

要求: 学生开发小组对上述客户需求仔细研究、分析, 同时参考网站开发项目 1、项目 2 介绍中有关网站开发的功能与技术指标, 并且在考察同类著名网站功能和设计特色的基础上, 拟定出本小组要开发网站系统的问题定义与网站原型基本功能与特色、开发技术与工具、设计初步方案、开发计划与成员分工等文档。然后再按照本课程设计指导书的各项要求进行设计、开发、测试与文档编制和总结。项目的分析与设计任务可以采用传统的结构化分析与设计方法, 也可以采用面向对象要分析与设计方法(如用 UML 工具 Rose) 。

城市政府网站评估指标及权重

一级指标	二级指标	三级指标
1. 网站内容服务指标	1.1 政务公开*	1.1.1 政府公报 1.1.2 政策法规 1.1.3 政务新闻 1.1.4 机构设置与职责 1.1.5 办事规程 1.1.6 网站背景 ……
	1.2 本地概览	……
	1.3 特色内容	
2. 网站服务功能指标*	2.1 网上办公*	2.1.1 导航服务 2.1.2 办事指南 2.1.3 网上咨询 2.1.4 网上查询 2.1.5 网上申报 2.1.6 网上审批 2.1.7 政府网上采购 2.1.8 相关机构链接 ……
	2.2 网上监督	
	2.3 公众反馈	2.3.1 政府信箱 2.3.2 网上调查 2.3.3 交流论坛 ……
	2.4 特色功能	
3. 网站建设指标	3.1 设计特色	3.1.1 美观性 3.1.2 专业性 3.1.3 易用性 3.1.4 通用性
	3.2 信息特性	3.2.1 时效性* 3.2.2 全面性 3.2.3 条理性 3.2.4 多媒体
	3.3 网络特性	3.3.1 连接/浏览速度 3.3.2 站点可用性 3.3.3 网络安全*

注:带号标记* 为较重要的指标

B.4　课程设计过程与具体要求

1. 学习课程设计指导书和分组

学习研究课程设计指导书，进行分组（网站开发项目5人一小组；个人主页设计项目2人一组）并且明确每个学生在开发小组中扮演的角色及承担的职责（包括选出组长）。

2. 确定目标、初步方案，准备、试用开发环境与工具

每个小组确定开发网站目标及初步方案；选择、准备、试用开发平台、数据库、交互网页开发技术、网页设计工具及其他有关开发工具。

3. 学习与搜集素材，借阅、购置必要的书籍与材料

学习开发小组及成员根据自己承担的任务利用各种途径（图书馆、因特网、书店、同学亲友等）进行针对性的学习并收集相关素材，包括精选、购置必要的书籍。

4. 课堂与课下结合开发项目

因为需要自学和探索的内容与软件较多，每个学生要特别发挥积极主动精神投入课程设计和开发活动。除了实验室正式安排的课程设计时间之外，学生需要充分利用好课余时间，自己有计算机的学生更要充分利用有利条件以取得尽可能好的开发成果，力争获得最大收益。

5. 各阶段的开发工作

小组开发各阶段的任务及成员角色分工参看后面的“开发阶段任务及角色分工一览表”。需要说明的是，表中给出的各阶段顺序是迭代进行的，可能需要反复多次改进才能最后完成。开发过程中，小组长必须承担起领导责任，不定期召开小组开发工作研讨会（建议5～8次）。会前有准备，会议有记录（包括日期、出席人员、主题、讨论纪要、结论与问题、计划与行动分工。事后要保存好供老师检查），会后有分工和检查。开发小组工作会议的可能内容是：

（1）选题、网站内容及开发方案研讨；小组成员分工；开发计划拟定。

（2）同类著名网站浏览、分析；网站需求分析；网站原型及成员分工确认。

（3）原型主页设计及网页组织研讨。

（4）数据库设计及应用研讨。

（5）交互网页开发技术或其他专门开发技术或开发工具使用的研讨。

（6）网站原型集成测试、原型功能改进与扩充；开发文档整理、汇总。

6. 做好小组与个人的开发记录、总结，做好小组内外的交流与互助

各个开发小组及其每个成员可以互相研讨、帮助，但必须独立完成自己承担的开发任务与文档编制任务，不得抄袭他人成果。在课程设计进行期间，每个小组由小组长建立项目开发记录本（不少于30页），每周至少做一次记录，包括小组会议记录，小组记录本要保存好供老师检查。

建议每个小组成员也建立自己个人的开发记录或日志。记录的内容可以包括：个人在小组中承担的任务、计划与进度；相关学科与软件工具学习内容摘要与存在问题、难点；好的创意与建议；开发或学习心得；文档草稿；重要信息与线索记录等。

这样做可以有助于项目开发工作和自己的学习，也有助于最后完成个人和小组的课程设计报告。

参考文献

1. 张海藩 . 软件工程导论(第三版) . 北京: 清华大学出版社, 1998
2. 郑人杰等 . 实用软件工程(第二版) . 北京: 清华大学出版社, 1997
3. Software Engineering——A Practitionaer ’s Approach, Roger S. Pressman, (英文版, 第 4 版) . 北京: 机械工业出版社 &McGraw-Hill, 1997
4. 软件工程: Java 语言实现 . Stephen R. Schach 著, 袁兆山等译 . 北京: 机械工业出版社, 1999
5. [美] Watts S. Humphrey 著, 袁显译 . 小组软件开发过程 . 北京: 人民邮电出版社, 2000
6. 张龙祥编著 . UML 与系统分析设计 . 北京: 人民邮电出版社, 2001
7. [美] I. Jacobson, G. Booch, J. Rumbaugh 著, 周伯生等译 . 统一软件开发过程 . 北京: 机械工业出版社, 2002
8. 郑人杰, 殷人昆, 陶永雷编著 . 实用软件工程(第二版) . 北京: 清华大学出版社, 1997
9. 殷人昆 . 软件工程复习与考试指导 . 北京: 高等教育出版社, 2001
10. 殷人昆, 田金兰, 马晓勤 . 实用面向对象软件工程教程 . 北京: 电子工业出版社, 1998
11. Ronald J. Norman,“ Object-Oriented Systems Analysis and Design ”. 中译名《面向对象系统分析与设计》. 北京: 清华大学出版社和 Prentice Hall 联合影印出版, 1997