

全国高等职业教育计算机类规划教材  
工作过程系统化教程系列

过程导向  
项目驱动  
能力培养  
面向就业

- 知名专家指航，企业资源融入，突出先进技术
- 理论实践一体，真实环境训教，突出开发理念
- 特色项目引领，工作过程导向，突出技能培养
- 网络资源支撑，师生自主学习，突出持续发展

# 软件设计与编程 基础（C语言版）

吴艳平 岳淑玲 主 编  
刘铁英 于艳华 副主编  
李明革 丛得成 主 审



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

全国高等职业教育计算机类规划教材·工作过程系统化教程系列

# 软件设计与编程基础 (C 语言版)

吴艳平 岳淑玲 主 编

刘铁英 于艳华 副主编

李明革 丛得成 主 审

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书是一本软件设计与编程的基础入门用书。全书以C语言为媒介,详细地介绍了结构化程序的开发的完整流程,主要内容包括项目背景、需求分析(项目计划书、需求规格说明书)、系统设计(概要设计说明书、详细设计说明书)、编码实现、系统测试与优化。

本书精心选择了“小学生数学选题系统”、“超市管理系统”两个项目,这些教学项目均由企业专家亲自指导,依据从简单到复杂的规律,逐步构建使用者的软件开发与设计理念,与以往的教材相比,更注重学习者能力的培养,具有一定的普遍性、实用性和可操作性。本书充分体现行为导向教学方法,在实施项目时,将面向过程的软件开发方法、算法及C语言相关知识均贯穿在项目这条主线上,让学生学会在工作中处理各类问题的方法,实现理论与实践一体化教学,把培养学生的能力放在首位。本教材同时提供了教材中所用项目的完整代码及配套电子课件。

本书可作为高职高专计算机专业及相关非计算机专业的教材使用,也可作为培训教材,也可供对程序设计感兴趣的初学者入门使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

## 图书在版编目(CIP)数据

软件设计与编程基础: C语言版 / 吴艳平, 岳淑玲主编. —北京: 电子工业出版社, 2009.7

全国高等职业教育计算机类规划教材·工作过程系统化教程系列

ISBN 978-7-121-08963-3

I. 软… II. ①吴…②岳… III. C语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2009)第086184号

策划编辑: 程超群

责任编辑: 张燕虹

印 刷:

北京市李史山胶印厂

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开 本: 787×1092 1/16 印张: 18.75 字数: 480千字

印 次: 2009年7月第1次印刷

印 数: 4000册 定价: 29.00元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。

## 《软件设计与编程基础 (C 语言版)》读者意见反馈表

尊敬的读者:

感谢您购买本书。为了能为您提供更优秀的教材,请您抽出宝贵的时间,将您的意见以下表的方式(可从 <http://www.huaxin.edu.cn> 下载本调查表)及时告知我们,以改进我们的服务。对采用您的意见进行修订的教材,我们将在该书的前言中进行说明并赠送您样书。

姓名: \_\_\_\_\_ 电话: \_\_\_\_\_

职业: \_\_\_\_\_ E-mail: \_\_\_\_\_

邮编: \_\_\_\_\_ 通信地址: \_\_\_\_\_

1. 您对本书的总体看法是:

☐ 很满意    ☐ 比较满意    ☐ 尚可    ☐ 不太满意    ☐ 不满意

2. 您对本书的结构(章节): ☐ 满意    ☐ 不满意    改进意见 \_\_\_\_\_

3. 您对本书的例题: ☐ 满意    ☐ 不满意    改进意见 \_\_\_\_\_

4. 您对本书的习题: ☐ 满意    ☐ 不满意    改进意见 \_\_\_\_\_

5. 您对本书的实训: ☐ 满意    ☐ 不满意    改进意见 \_\_\_\_\_

6. 您对本书其他的改进意见:

7. 您感兴趣或希望增加的教材选题是:

请寄: 100036 北京市万寿路 173 信箱高等教育分社 收

电话: 010-88254565    E-mail: [gaozhi@phei.com.cn](mailto:gaozhi@phei.com.cn)

## 反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为，歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396; (010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

# 前言

随着企业对软件人才需求量的激增,高职院校人才培养的质量与规格也在不断提升,需要我们培养出具有较高的专业能力、方法能力和社会能力等综合能力的高素质的人才。这就要求我们开设的课程要以工作过程为导向,采用行动导向教学理念,全面培养学生的职业能力和提高学生可持续发展能力,使他们将来能尽快适应岗位变化。但是,目前国内软件专业中还没有一门比较成熟、可行性强、适合职业人才培养的此类课程,因而也没有与此类课程相配套的教材。

本教材是在分析企业工作任务、提取典型工作任务之后,由多年从事软件开发、设计的企业人员和一线教学人员根据软件企业工作过程系统化的原理编写而成的,它的前身是《C 语言程序设计》,曾连续几年作为长春职业技术学院软件专业教材使用。近年来,根据工作过程系统化的指导思想,在企业专家的参与下,将软件开发方法、算法等多学科知识融入教材中,通过两个项目的完整设计过程对教材内容重新进行了组织、整理,根据企业需求及学生的认知规律,将项目划分成多个学习性工作任务,并针对不同的工作任务,提供相应的引导文,引领学生在任务的完成过程中逐步掌握相应的知识与技能。本教材有以下主要特色。

(1) 适应工作过程系统化的教学模式。教学项目均由企业专家亲自指导,具有普遍性、实用性、可操作性。

(2) 以典型工作过程项目为主线,贯穿全书始终。本教材精心选择了“小学生数学选题系统”、“超市管理系统”两个项目,又将项目分解为多个既独立又有一定联系的小任务,使学生在完成任务的过程中,掌握软件设计的基本理念和编程的基本方法。

(3) 案例依据从简单到复杂的规律,逐步构建使用者的软件开发与设计理念,与以往的教材相比,更注重学习者能力的培养。本教材充分体现行为导向教学方法,在实施项目时,将面向过程的软件开发方法、算法及 C 语言相关知识均贯穿在项目这条主线上,让学生学会在工作中处理各类问题的方法,实现理论与实践一体化教学,也真正把培养学生的方法能力放在首位,使学生具有可持续发展的能力。

(4) 充分体现工学结合,从教材的策划到实施,都有企业专家把关,符合企业需求。

本教材共分为 4 章,章节划分如下。

第 1 章 软件开发与程序设计,包括结构化软件开发的一般方法及开发流程。

第 2 章 小学生数学选题系统,包括了该系统的整个开发流程,即项目背景、需求分析、系统设计、编码实现、系统测试与优化。

第 3 章 超市管理系统,包括了该系统的整个开发流程(同“小学生数学选题系统”)。

第 4 章 项目赏析——学生成绩管理系统,展示了用 C 语言开发一个系统的基本方法和步骤。

本书由长春职业技术学院的吴艳平、岳淑玲任主编,刘铁英、于艳华任副主编;王军、董晶、李权威(上海朋道信息有限公司长春分公司)也参与了本书的编写工作。李明革教授、丛得成工程师(启明信息技术股份有限公司)担任本教材主审工作。电子工业出版社的编辑对本书的编写提出了许多宝贵的意见,长春职业技术学院软件专业教研室主任、教务科长陈显刚同志给予了大力支持与帮助,在此表示感谢。

本教材适合作为高职高专计算机类专业软件设计与编程基础类教材，同时也可以作为培训教材，也可供对程序设计感兴趣的初学者入门使用。由于时间仓促以及编者水平有限，书中难免存在错误和疏漏之处，欢迎广大读者和同仁提出宝贵意见和建议。E-mail: wwyypp@126.com。

编者

2009年4月

# 目 录

|                              |       |
|------------------------------|-------|
| 第1章 软件开发与程序设计                | (1)   |
| 1.1 任务一：软件的开发                | (1)   |
| 1.2 任务二：结构化程序设计              | (6)   |
| 第2章 小学生数学选题系统                | (13)  |
| 2.1 任务一：“小学生数学选题系统”的项目背景     | (13)  |
| 2.2 任务二：“小学生数学选题系统”的需求分析     | (14)  |
| 2.2.1 子任务一：编写项目计划书           | (14)  |
| 2.2.2 子任务二：编写需求规格说明书         | (20)  |
| 2.3 任务三：“小学生数学选题系统”的设计       | (24)  |
| 2.3.1 子任务一：编写概要设计说明书         | (24)  |
| 2.3.2 子任务二：编写详细设计说明书         | (27)  |
| 2.4 任务四：“小学生数学选题系统”的编码实现     | (35)  |
| 2.4.1 子任务一：界面设计              | (35)  |
| 2.4.2 子任务二：登录模块的功能实现         | (48)  |
| 2.4.3 子任务三：题量设置模块的功能实现       | (59)  |
| 2.4.4 子任务四：四则题库模块的功能实现       | (68)  |
| 2.4.5 子任务五：评分系统模块的功能实现       | (91)  |
| 2.5 任务五：“小学生数学选题系统”的测试与优化    | (100) |
| 第3章 超市管理系统                   | (105) |
| 3.1 任务一：“超市管理系统”的项目背景        | (105) |
| 3.2 任务二：“超市管理系统”的需求分析        | (106) |
| 3.2.1 子任务一：编写项目计划书           | (106) |
| 3.2.2 子任务二：编写需求规格说明书         | (108) |
| 3.3 任务三：“超市管理系统”的设计          | (111) |
| 3.3.1 子任务一：编写概要设计说明书         | (111) |
| 3.3.2 子任务二：编写详细设计说明书         | (114) |
| 3.4 任务四：“超市管理系统”的编码实现        | (121) |
| 3.4.1 子任务一：界面设计              | (121) |
| 3.4.2 子任务二：登录模块的功能实现         | (137) |
| 3.4.3 子任务三：数据结构设计            | (148) |
| 3.4.4 子任务四：商品维护模块的功能实现       | (157) |
| 3.4.5 子任务五：会员管理——会员添加模块的功能实现 | (166) |
| 3.4.6 子任务六：会员管理——会员查询模块的功能实现 | (176) |
| 3.4.7 子任务七：会员管理——会员统计模块的功能实现 | (189) |
| 3.4.8 子任务八：会员管理——会员删除模块的功能实现 | (201) |
| 3.4.9 子任务九：商品销售——购物车清单的功能实现  | (210) |

|        |                                |       |
|--------|--------------------------------|-------|
| 3.4.10 | 子任务十：商品销售——动态处理商品数量的功能实现 ..... | (217) |
| 3.4.11 | 子任务十一：商品销售——商品结算的功能实现 .....    | (222) |
| 3.4.12 | 子任务十二：库存预警模块的功能实现 .....        | (227) |
| 3.5    | 任务五：“超市管理系统”的测试与优化 .....       | (232) |
| 第4章    | 项目赏析——学生成绩管理系统 .....           | (242) |
| 4.1    | 概述 .....                       | (242) |
| 4.1.1  | 学生成绩管理系统的背景 .....              | (242) |
| 4.1.2  | 系统流程概要 .....                   | (243) |
| 4.2    | 明确问题 .....                     | (244) |
| 4.3    | 分析 .....                       | (245) |
| 4.4    | 设计算法 .....                     | (245) |
| 4.4.1  | 概要设计 .....                     | (245) |
| 4.4.2  | 详细设计 .....                     | (247) |
| 4.5    | 实现 .....                       | (248) |
| 4.6    | 测试 .....                       | (274) |
| 4.7    | 设计说明 .....                     | (274) |
| 4.7.1  | 设计内容的综合性 .....                 | (274) |
| 4.7.2  | 分析方法 .....                     | (274) |
| 4.7.3  | 测试 .....                       | (274) |
| 4.7.4  | 待完善的问题 .....                   | (275) |
| 附录A    | 常用字符与ASCII码对照表 .....           | (276) |
| 附录B    | C语言运算符的优先级与结合方向 .....          | (277) |
| 附录C    | C语言常用库函数 .....                 | (278) |
| 附录D    | VC++ 6.0 常用菜单功能说明 .....        | (285) |
| 附录E    | scanf、printf 函数格式字符表 .....     | (290) |
| 参考文献   | .....                          | (291) |

# 第1章 软件开发与程序设计

## 1.1 任务一：软件的开发

软件被应用于世界的各个领域，对人们的生活和工作产生了深远的影响。近年来，软件开发作为现代科学研究和解决工程问题的基础，成为当今世界不可缺少的一部分。在不久的将来，软件将成为驱动社会发展的新动力。

### 任务描述

软件开发的目标：以客户与市场为导向、理解软件真正的需求，开发出具有可修改性、有效性、可靠性、可理解性、可维护性、可重用性、可适应性、可移植性、可追踪性和可互操作性并满足用户需要的软件产品，获取最大的商业利益。

### 任务分析与设计

如果硬件是人的身体，那么软件就是人的思维。在开发软件过程中，开发的步骤、方法甚至技巧和习惯都会直接影响软件成品的质量。软件开发项目的成功有以下几个主要的目标作为验证标准：

- (1) 付出较低的开发成本。
- (2) 达到要求的软件功能。
- (3) 取得较好的软件性能。
- (4) 开发的软件易于移植。
- (5) 需要较低的维护费用。
- (6) 能按时完成开发工作，即时交付使用。

### 任务实现

#### 1. 初识“软件”

软件（software）是一系列按照特定顺序组织的计算机数据和指令的集合。

凡是能看见、能摸到的都是硬件，如键盘、显示器、鼠标、主机等。

运行于硬件上的、我们常用的瑞星、Office、操作系统等都是软件！没有软件的计算机是“裸机”，相当于废铁一堆。软件和计算机密不可分，计算机软件是为了解除人们繁重、重复的工作而产生的。软件是为了告诉计算机要做什么、如何做而编写的，是计算机能够理解的一串指令、代码、程序。

#### 2. 软件的分类

##### 1) 按功能划分

- (1) 贴近计算机硬件的小软件。通常“固化”在只读存储器芯片中，因此称为固件。

(2) 系统软件。包括操作系统和编译器等软件，如 Windows、Linux、UNIX、Mac OS。系统软件与硬件共同搭建起一个操作“平台”，可以管理和优化计算机硬件资源的使用。

(3) 支撑软件。是支撑各种软件的开发与维护的软件，又称为软件开发环境，如 Microsoft Visual C、Microsoft Visual Studio、Maromedia Dreamweaver 等。

(4) 应用软件。它的种类最多，如办公软件、电子商务软件、通信软件、行业软件，游戏软件等。根据用户和所服务的领域提供不同的功能，是为了某种特定的用途而开发的软件。本书开发实例的方向就是应用软件。

2) 按软件规模划分

根据开发软件所需的人员数量、时间期限以及完成的源程序（有效）行数，划分出 6 种不同规模的软件，如表 1.1 所示。

表 1.1 软件规模的分类

| 类 别 | 参加人员数     | 研制期限   | 产品规模（源程序行数） |
|-----|-----------|--------|-------------|
| 微型  | 1         | 1~4 周  | 0.5k        |
| 小型  | 1         | 1~6 月  | 1~2k        |
| 中型  | 2~5       | 1~2 年  | 5~50k       |
| 大型  | 5~20      | 2~3 年  | 50~100k     |
| 甚大型 | 100~1000  | 4~5 年  | 1M (=1000k) |
| 极大型 | 2000~5000 | 5~10 年 | 1~10M       |

3. 软件生命周期与开发阶段

软件生命周期是按时间分程的思想方法描述软件的产生直到报废的生命周期。软件生命周期包含制订计划、需求分析、系统设计、程序编码、软件测试、运行维护 6 个开发阶段，如表 1.2 所示。

表 1.2 软件生命周期的 6 个开发阶段

| 开 发 阶 段 | 关 键 问 题                        | 结 束 标 准                   |
|---------|--------------------------------|---------------------------|
| 制订计划    | 问题是什么？<br>有可行的解决途径吗？           | 确定开发系统的目标、可行性分析报告、项目开发计划书 |
| 需求分析    | 系统必须做些什么？                      | 软件需求规则说明书（系统必备功能）         |
| 软件设计    | 如何概括并分解出解决步骤？<br>怎样具体地实现系统的模块？ | HIPO 图                    |
| 程序编码    | 如何用计算机语言表达系统设计？                | 源程序清单                     |
| 软件测试    | 在设计测试用例的基础上，检验软件各个组成部分是否正确？    | 测试文档                      |
| 运行维护    | 已交付的软件能否正确运行维护？                | 改正性维护、适应性维护、完善性维护、预防性维护   |

4. 各个开发阶段之间的交接——开发文档

软件并非只包括可以在计算机上运行的程序，与这些程序相关的文件一般也被认为是软件的一部分。软件文档是软件工程实施中的重要成分，它不仅是软件开发各阶段的重要依据，而且也影响软件的可维护性。

各个开发阶段之间的交接实际上就是文档与文档之间的交接。换句话说，每个开发阶段

都应该产生相应的开发文档。文档是开发团队中技术人员的交流工具，可以快速地找到相关的位置，避免重复阅读程序。

一份合格的开发文档应该全面，要有条理地诠释软件开发流程，要包含技术难点的解释、逻辑判断、重要算法，要包括部分业务流程的说明。技术人员通过看文档可迅速了解程序的流程、调用关系，测试人员可以知道开发人员对用户需求了解的程度、是否有偏差，以快速定位错误，提高开发效率和开发质量。

### 5. 软件的开发方法

#### 1) 开发方法的分类

- (1) 功能分解法——计算任务。
- (2) 结构化法——以数据为中心。
- (3) 面向对象法——以对象为中心。
- (4) 组件法——以组件为中心。

#### 2) 结构化系统开发方法 (Structured System Development Methodology)

结构化系统开发方法是由 E.Yourdon 和 L.L.Constantine 提出的，首先对软件进行需求分析，然后进行总体设计，最后结构化编程 (SP)。

按照程序设计方法的发展，程序设计方法可以分为：

- (1) 功能分解法，以计算任务为中心。
- (2) 结构化程序设计，以数据为中心。
- (3) 面向对象程序设计，以对象为中心。
- (4) 组件程序设计，以组件为中心。

初学者更容易理解结构化 (模块化) 的理念，即自顶向下对系统进行分析与设计。这种开发方法能产生清晰、易懂的程序代码，使用程序易于维护。

按用户至上的原则，将系统开发过程划分为若干个相对独立的阶段，然后再考虑局部实现。实施阶段坚持自底向上地逐步实施，即从最基层的模块做起 (编程)，然后按照系统设计的结构，将模块一个个拼接到一起进行调试，自底向上、逐步地构成整个系统。

### 引导文献

软件工程是随着软件的发展而诞生的学科。软件工程由程序、数据和文档组成，其中程序是主体，数据是使程序能正常操纵信息的数据结构，文档是与软件的开发、维护和使用有关的材料。软件开发方法应规范化。

#### 1. 软件开发的主要环节

软件开发的主要环节如图 1.1 所示。

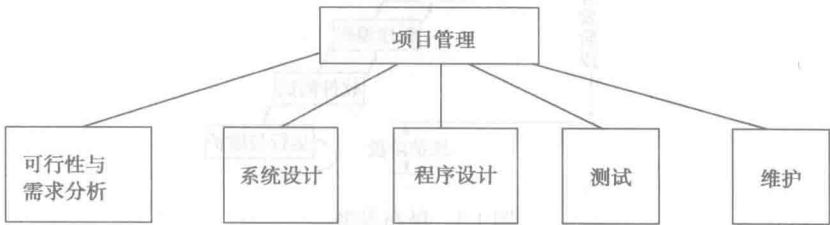


图 1.1 软件开发的主要环节

## 2. 软件开发的策略与步骤

软件开发有三种基本策略：“复用”、“分而治之”、“优化—折中”。

(1) 复用是指“利用现成的东西”。

(2) 分而治之是指把一个复杂的问题分解成若干个简单的问题，然后逐个解决，如图 1.2 所示。

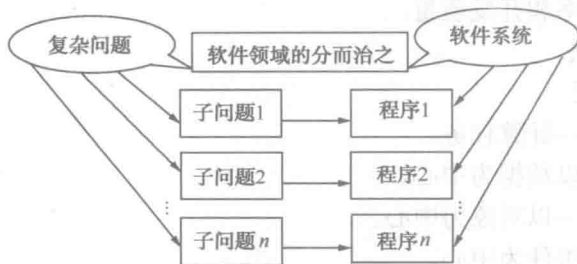


图 1.2 分而治之

软件人员在分而治之时应考虑：问题分解后能否用程序实现？程序能否集成为一个软件系统以解决问题？

(3) 优化—折中。优化工作不是可有可无的事情，而是必须要做的事情。当优化工作成为一种责任时，程序员才会不断改进软件中的算法、数据结构和程序组织，从而提高软件质量。折中策略是指通过协调各个质量因素，实现整体质量的最优。

## 3. 软件开发模型

软件开发模型是软件开发全部过程、活动和任务的结构框架。最早出现的软件开发模型是于 1970 年由 W. Royce 提出的瀑布模型，而后随着软件工程学科的发展和软件开发的实践，相继提出了演化模型、螺旋模型、增量模型、喷泉模型、智能模型等。

软件也有一个孕育、诞生、成长、成熟、衰亡的生存过程。根据这一思想，可以得到软件生命周期的 6 个开发阶段，即制订计划、需求分析、设计、程序编码、测试、运行与维护。

软件生命周期模型是从软件项目需求定义直至软件经使用后废弃为止，跨越整个生命周期的系统开发、运作和维护所实施的全部过程、活动和任务的结构框架。

(1) 瀑布模型：制订计划，进行需求分析和说明、软件设计、程序编码、软件测试、运行与维护，固定次序、自上而下、相互衔接。瀑布模型如图 1.3 所示。

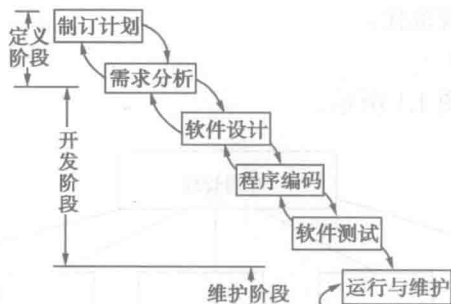


图 1.3 瀑布模型

(2) 演化模型：在项目开发的初始阶段，对软件的需求认识不够清晰，可以先做试验开

发与探索可行性，弄清软件需求；在此基础上获得较为满意的软件产品。第一次得到的试验性产品称为“原型”。

(3) 螺旋模型：开发复杂的大型软件时，结合瀑布模型与演化模型沿着螺旋线旋转，自内原型处向外拓展，每旋转一圈便更新完善一个新的版本。螺旋模型如图 1.4 所示。

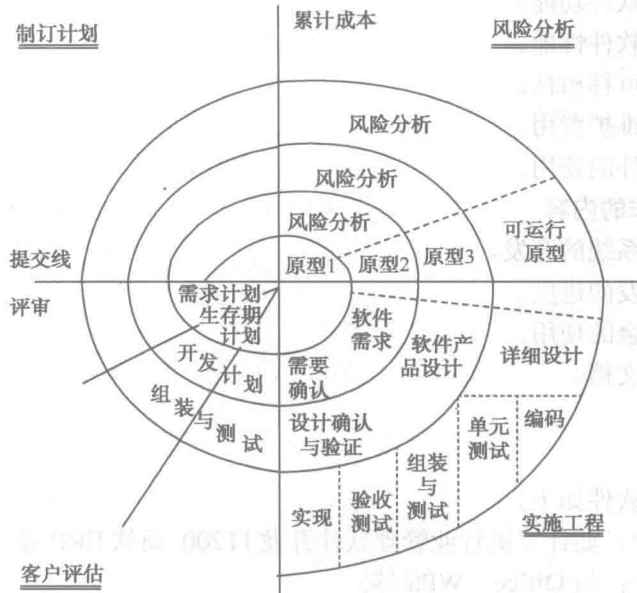


图 1.4 螺旋模型

- ① 制订计划——确定软件目标，选定实施方案，弄清项目开发的限制条件。
- ② 风险分析——分析所选方案，考虑如何识别和消除风险。
- ③ 实施工程——实施软件开发。测试验收产品。
- ④ 客户评估——评价开发工作，提出修正建议。

(4) 喷泉模型：主要支持面向对象的开发方法。分析、设计和编码之间不存在明显的边界。喷泉模型如图 1.5 所示。

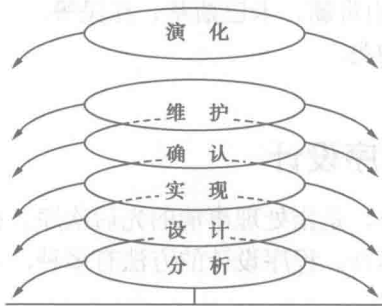


图 1.5 喷泉模型

(5) 智能模型：基于知识的软件开发模型，综合以上模型，结合专家系统，采用归约和推理机制，帮助软件人员完成开发工作。

## 1. 软件工程实现的目标

- (1) 付出较低的开发成本。
- (2) 实现要求的软件功能。
- (3) 获取较好的软件性能。
- (4) 提高软件的可移植性。
- (5) 降低系统的维护费用。
- (6) 按时交付软件的使用。

## 2. 软件工程工作的内容

- (1) 分阶段实施系统的开发。
- (2) 控制调整开发的进度。
- (3) 控制优化资金的使用。
- (4) 提交阶段性文档。

## 拓展任务

(1) 较常见应用软件如下。

- ① 行业管理软件：如计算机行业管理软件开龙 IT200 商软 ERP 等。
- ② 文字处理软件：如 Office、WPS 等。
- ③ 信息管理软件：如 Access 数据库。
- ④ 辅助设计软件：如 AutoCAD、Photoshop。
- ⑤ 媒体播放软件：如暴风影音、豪杰超级解霸、Windows Media Player、RealPlayer 等。
- ⑥ 系统优化软件：如 Windows 优化大师、超级兔子魔法设置。
- ⑦ 图形图像软件：Coreldraw、Painter、MAX, MAYA 等。
- ⑧ 数学软件：Mathematica、Maple、Matlab、MathCad 等。
- ⑨ 统计软件：SAS、SPSS 等。
- ⑩ 后期合成软件：After Effects、Combustion、Digital Fusion、Shake、Flame 等。
- ⑪ 杀毒软件：如瑞星、金山毒霸、卡巴斯基、江民等。

(2) 列举出更多的软件并归类。

## 1.2 任务二：结构化程序设计

程序设计是软件开发的核⼼，是指处理事情的先后次序。在计算机语言中，程序定义为完成特定任务的计算机指令的集合。程序设计的方法有多种，本书选用最容易理解的结构化设计方法。

## 任务描述

工欲善其事，必先利其器！开发软件必须树立正确的开发观念和基本的程序开发技巧，为今后专业化开发打好基础。

了解结构化程序设计的理念在软件开发中的重要地位,掌握结构化程序设计的精髓和技巧,选择最恰当的入门语言。

### 任务实现

#### 1. 结构化设计方法 (Structured Design, SD)

从系统设计的角度出发,软件设计方法可以分为三大类。第一类是根据系统的数据流进行设计,称为面向数据流的设计或者过程驱动的设计,以结构化设计方法为代表。第二类是根据系统的数据结构进行设计,称为面向数据结构的设计或者数据驱动的设计,以 LCP (程序逻辑构造) 方法、Jackson 系统开发方法和数据结构化系统开发 (DSSD) 方法为代表。第三类是面向对象的设计。

结构化设计方法是在模块化、自顶向下细化、结构化程序设计等程序设计技术的基础上发展起来的。该方法实施的要点是:建立数据流的类型,指明流的边界,将数据流图映射到程序结构,用“因子化”方法定义控制的层次结构,用设计测量和一些启发式规则对结构进行细化。

#### 2. 软件设计的原则

##### 1) 抽象化

对软件进行模块设计时,可以有不同的抽象层次。在最高的抽象层次上,可以使用问题所处环境的语言描述问题的解法;而在较低的抽象层次上,则采用过程化的方法。

(1) 过程的抽象:在软件工程过程中,从系统定义到实现,每进展一步都可以看做是对软件解决方案的抽象化过程的一次细化。在软件计划阶段,软件被当做整个计算机系统中的一个元素来看待。在软件需求分析阶段,用“问题所处环境的为大家所熟悉的术语”来描述软件的解决方法。在从概要设计到详细设计的过程中,抽象化的层次逐次降低。当产生源程序时,到达最低的抽象层次。

(2) 数据抽象:数据抽象与过程抽象一样,允许设计人员在不同层次上描述数据对象的细节。例如,可以定义一个数据对象 student,并将它规定为一个抽象的数据类型,用它的构成元素来定义它的内部细节。

student 的抽象: `student { name; age; sex; }`

此时,数据抽象 student 本身由另外一些数据“name (名字)、age (年龄)、sex (性别)”抽象构成。在定义 student 为一个抽象的数据类型之后,可以引用它来定义其他用数据对象,如“Rose, John”,而不必涉及 student 的内部细节。

使用 student 定义其他数据对象: `student Rose, John。`

(3) 控制抽象:与过程抽象和数据抽象一样,控制抽象可以包含一个程序控制机制而无须规定其内部细节。控制抽象的例子就是在操作系统中用以协调某些活动的同步信号。

##### 2) 自顶向下,逐步细化

将软件的体系结构按自顶向下方式,对各个层次的过程细节和数据细节逐步细化,直到用程序设计语言的语句能够实现为止,从而最后确立整个的体系结构。最初的说明只是概念性地描述了系统的功能或信息,但并未提供有关功能的内部实现机制或有关信息的内部结构的任何信息。设计人员对初始说明仔细推敲,进行功能细化或信息细化,给出实现的细节,

划分出若干成分。然后再对这些成分施行同样的细化工作。随着细化工作的逐步展开，设计人员就能得到越来越多的细节。

### 3) 模块化

模块又称构件，是能够单独命名并独立地完成一定功能的程序语句的集合。例如，高级语言中的过程、函数、子程序等都可作为模块。

模块化是软件的一个重要属性。模块化的特性提供了人们处理复杂问题的一种方法，同时也使得软件能够被有效地管理。

软件系统的层次结构正是模块化的具体体现。也就是说，整个软件被划分成若干单独命名和可编址的部分，称为模块。这些模块可以被组装起来以满足整个问题的需求。

模块和其他用模块之间的接口应尽可能独立。

(1) 系统设计通常分为结构设计和过程设计两个阶段。

① 结构设计：确定系统由哪些模块组成，以及这些模块之间的相互关系。结构设计是总体设计阶段的任务。

② 过程设计：确定每个模块的处理过程。过程设计是详细设计阶段的任务。

(2) 设计的后处理：在确定系统的软件结构以后，还必须做好下述工作。

① 为每个模块开发一份功能说明。

② 为每个模块提供一份接口说明。

③ 定义局部的和全程的数据结构。

④ 给出所有的设计限制或约束。

⑤ 进行总体设计评审。

⑥ 如果需要和可能，则进行设计“优化”。

(3) 模块的独立性与信息隐蔽原理。每个模块的实现细节（过程和数据）对于其他模块来说应该是隐蔽的，即包含在模块中的信息（过程或数据）对于其他不需要这些信息的模块来说，是不能访问的，或者是“不可见”的。

有效的模块化可以通过定义来实现，模块间的通信对于软件功能来说是必要的。通过信息隐蔽，可以实施对模块的过程细节和局部数据结构的限制。

模块的独立性是软件质量的关键：模块化程度较高的软件容易开发，模块化程度较高的软件也比较容易测试和维护。

### 4) 控制层次

控制层次也称为程序结构，它表明了程序构件（模块）的组织情况。控制层次往往用程序的层次（树形或网状）结构来表示。

(1) 程序结构的深度：程序结构的层次数称为结构的深度，反映了程序的规模和复杂程度。

(2) 程序结构的宽度：同层次结构中的最大模块个数称为结构的宽度。

(3) 模块的扇入和扇出：扇出表示一个模块直接调用（或控制）的其他用模块数目。扇入定义为调用（或控制）一个给定模块的模块个数。多扇出意味着需要控制和协调许多下属模块。多扇入的模块通常是公用模块。

### 5) 结构的划分

程序结构可以按水平方向或垂直方向进行划分。

(1) 水平划分按程序功能定义模块各分支。顶层模块是控制模块，用来协调程序各个功