工业和信息化普通高等教育“十二五”规划教材立项项目

21世纪高等教育计算机规划教材

COMPUTER

数据结构课程设计案例教程

Course Book of Data Structure Engineering Training

马巧梅 庞晓琼 杨秋翔 付东来 陈文俊 编著

- 精选有代表性的案例
- 融入软件工程的思想
- 案例驱动方式进行题目的讲解

人民邮电出版社
POSTS & TELECOM PRESS

工业和信息化普通高等教育“十二五”规划教材立项项目
21世纪高等教育计算机规划教材

COMPUTER

数据结构课程设计案例教程

Course Book of Data Structure Engineering Training

马巧梅 庞晓琼 杨秋翔 付东来 陈文俊 编著

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

数据结构课程设计案例教程 / 马巧梅等编著. -- 北京 : 人民邮电出版社, 2012.8
21世纪高等教育计算机规划教材
ISBN 978-7-115-28043-5

Ⅰ. ①数… Ⅱ. ①马… Ⅲ. ①数据结构－课程设计－高等学校－教材 Ⅳ. ①TP311.12

中国版本图书馆CIP数据核字(2012)第130848号

内 容 提 要

本书共13章，第1章主要按照软件工程的思想介绍数据结构案例分析与设计的思路和步骤，后面的12章精选12个综合案例，以案例驱动来展示利用数据结构的相关知识解决一些实际问题的过程。每个案例根据实际问题，给出了解决思路，设计了解决问题相对应的数据结构和算法，然后利用C语言进行了具体实现，最后在拓展知识部分引出更深层次的问题供读者借鉴和思考。

本书可作为高等院校计算机及相关专业本科生和专科生数据结构实践环节的教材，也可作为计算机工程技术人员学习的参考书。

21世纪高等教育计算机规划教材

数据结构课程设计案例教程

◆ 编　　著　马巧梅　庞晓琼　杨秋翔　付东来　陈文俊
责任编辑　邹文波

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街14号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
大厂聚鑫印刷有限责任公司印刷

◆ 开本：787×1092　1/16
印张：15.5　　2012年8月第1版
字数：450千字　　2012年8月河北第1次印刷

ISBN 978-7-115-28043-5

定价：29.80元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前 言

“数据结构”课程着眼于对基本数据的逻辑结构和存储结构进行分析和阐述，讲解基本数据结构的应用并介绍典型的基本算法。实践环节作为数据结构教学中的一个重要环节，它是对“数据结构”课程教学理论的延伸和补充，是对数据结构理论知识的综合应用。在“数据结构”课程教学环节中是按照知识点分别进行讲解的，但在处理具体问题的时候，往往要应用数据结构中多个知识点的内容，综合性的案例可以帮助读者利用课程知识处理实际问题，培养解决实际问题的能力。

本书是作者根据多年的教学经验和教学的实际需求编写而成的，书中精选了12 个综合案例，以案例驱动方式来展示一些实际性的例子，使读者更直接地了解和掌握数据结构的内容。书中的每个综合案例都采用通俗易懂的分析方法，融入了软件工程的思想，深入浅出地将数据结构课程设计问题逐步细化并加以解决；能够清晰地看出数据结构的实际应用过程，引导读者思考，形成正确思路。这样的方式对于提高读者驾驭全局能力和增长编写较大程序的经验很有帮助。书中大部分的案例都针对读者在设计中的薄弱环节进行了详细分析、设计和具体实现，如对文件的读写，这个问题许多读者往往是经验不足，容易忽视，从而导致数据只存放在内存中，初始化数据和检查数据结果十分困难等。本书针对不同案例，根据其数据结构、题目要求的不同，相关数据文件结构、文件读写方式也略有不同，有一定的示范意义。本书中案例的实现代码均在 VC++ 6.0 中调试通过，且都力求做到代码清晰，定义规范，流程清楚，可读性强。

本书由马巧梅、庞晓琼、杨秋翔、付东来、陈文俊编著，其中，第 2 章、第 11 章、第 13 章由马巧梅编著，第 7 章、第 8 章、第 9 章由庞晓琼编著，第 3 章、第 10 章、第 12 章由杨秋翔编著，第 1 章、第 4 章、第 6 章由付东来编著，第 5 章由陈文俊编著。在本书的编写中参考了国内外许多文献资料，在此向这些文献的作者表示感谢。

由于编者水平有限，本书中可能存在一些缺点和错误，恳请读者给予批评指正。

编著者

2012 年 7 月

目 录

第 1 章 数据结构课程设计概要

1.1 课程设计目的及意义

数据结构是计算机科学与技术以及相关专业的一门专业基础课程，属于计算机科学与技术专业课程体系中的核心课程之一，在计算机科学领域的主干课程中具有承上启下的作用。目前，计算机处理的对象不仅是简单的数值或字符，更多的是要处理具有不同结构的数据。因此，要设计一个好的软件，除了要掌握一定的计算机程序设计语言的知识之外，还必须研究各类数据的特性以及数据之间存在的关系。这是因为计算机要加工处理的数据必须输入到计算机中，并能够以恰当的方式在计算机中表示并存储起来，还要便于根据需要对数据进行加工和处理。因而当各种数据输入计算机之前，必须先按某种数据的组织形式将数据组织好，然后还要考虑以什么样的方式进行存储，这种组织形式和存储方式直接关系到程序对数据的处理能力和处理效率，并影响到问题的解决。因此，数据结构课程着眼于对基本数据的逻辑结构和存储结构进行分析和阐述，讲解基本数据结构的应用并介绍典型的基本算法。数据结构课程的目的是使学习者学习、分析、研究计算机加工的数据对象的特性，学会数据的组织方法，以便选择合适的数据逻辑结构和存储结构以及相应的操作，将现实中的问题转换为可以在计算机中表示和处理的问题。

数据结构课程设计是一个独立的实践环节，是对数据结构课程教学理论的延伸和补充，是对数据结构抽象理论知识的综合应用。它有助于学生进一步掌握程序设计的技能与方法，初步感受软件开发过程的项目管理方法与规范，更重要的目的是培养学生分析问题、解决问题、编写程序、动手操作的能力以及锻炼学生的设计创新能力。

随着时代的发展，软件开发语言、开发工具也不断发生着变化，许多数据结构已经不需要编程者自己构建，而是由软件开发语言自带。同时，软件开发工作也变得越来越简单化，学习门槛不断的降低，似乎已经成了理工科学生人人皆会的一种技能，没有学过数据结构的人照样能编出像模像样的程序。但是，作为计算机专业或是对计算机有浓厚兴趣的学生来说，数据结构是非常重要的一门课程，它在基础性研究领域和在对算法时间空间效率要求严格的场合发挥着重要的作用。希望计算机专业及相关专业的学生要努力成为伟大的“设计师”，而不仅仅局限在成为熟练的“工匠”。

1.2 课程设计解题思路

用计算机解决一个具体问题通常需经过 3 个步骤，首先将该具体问题抽象为一个适当的数学

模型，然后依据该数据模型设计或选择一个相应的算法，最后编出程序进行调试、测试，直至得到最终的解答。数据结构课程设计也是按此步骤进行，具体过程如下。

1. 抽象数学模型

数据结构（Data Structure）是相互之间存在一种或多种特定关系的数据元素的集合。在任何问题中，数据元素都不是孤立存在的，它们之间总是存在着某种关系。这种数据元素相互之间的关系称为结构。根据数据元素之间关系的不同特性，通常有下列 4 类基本结构。

（1）集合结构：结构中的数据元素之间除了“同属于一个集合”的关系外，别无其他关系。

（2）线性结构：结构中的数据元素之间存在一个对一个的关系。

（3）树形结构：结构中的数据元素之间存在一个对多个的关系。

（4）图状结构或网状结构：结构中的元素之间存在多个对多个的关系。

建模过程就是要抽象出所描述问题要处理的数据对象及其关系、问题求解的要求及方法等。

例 1.1 图书馆的书目检索系统自动化问题。

当你想借阅一本参考书，但不知道书库中是否有该书的时候；当你想找某一方面的参考书而不知图书馆内有哪些这方面的书时，都需要到图书馆去查阅图书目录卡片。在图书馆内有各种名目的卡片：有按书名编排的，有按作者名编排的，还有按分类编排的，等等。若利用计算机实现自动检索，则计算机处理的对象便是这些目录卡片上的书目信息。列在一张卡片上的一本书的书目信息可由登录号、书名、作者名、分类号、出版单位、出版时间等若干项组成。每一本书都有唯一的一个登录号，但不同的书目之间可能有相同的书名，或者有相同的作者名，或者有相同的分类号。由此，在书目自动检索系统中可以建立一张按登录号顺序排列的书目文件和三张分别按书名、作者名和分类号顺序排列的索引表，如图 1.1 所示。由这四张表构成的文件便是书目自动检索的数学模型。计算机的主要操作便是按照某个特定要求（如给定书名）对书目文件进行查询。诸如此类的还有查号系统自动化、仓库账目管理等。在这类文档管理的数学模型中，计算机处理的对象之间通常是一种最简单的线性关系。这类数学模型可称为线性的数据结构。

001	高等数学	樊映川	S01	…
002	大学物理	王薇薇	W01	…
003	化学工程	郝兵	H01	…
004	计算机网络	万树蜂	J01	…
…	…	…	…	…

高等数学	001，…
大学物理	002，…
计算机网络	004，…
…	…

樊映川	001，…
…	…

W	002，…
S	001，…
J	004，…
…	…

图 1.1 图书目录文件示例

例 1.2 铺设煤气管道问题。

假设要在某个城市的 n 个居民区之间铺设煤气管道，则在这 n 个居民区之间只要铺设 $n-1$ 条管道即可。假设任意两个居民区之间都可以架设管道，但由于地理环境的不同，所需经费也不同，采用什么样的施工方案能使总投资尽量少呢？这个问题可抽象为“求图的最小生成树”问题。

其数学模型如图 1.2 所示。图中“顶点”表示居民区，顶点之间的连线及其上的数值表示可以架设的管道及所需经费。求解的算法为：在可能架设的 m 条管道中选取 $n-1$ 条，既能连通 n 个居民区，又使总投资达到“最小”。

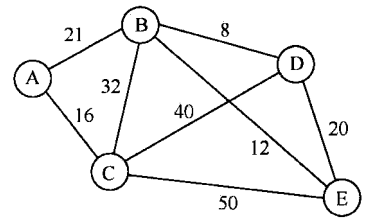

图 1.2　管道铺设问题

总之，正确建立数学模型是解决问题的关键所在，这就要求我们具有扎实的数学基础、高度的抽象思维能力，同时熟练地掌握数据结构课程所介绍的线性表、串、队列、栈、数组、树、图等各种结构（模型）的存储方法和基本的操作算法。

2. 构思关键算法

数学模型建立后，应仔细考虑解题用的关键算法。不同数据结构都有其典型算法，如树的前序、中序和后序遍历算法，图的最小生成树、最短路径算法等。这些算法十分经典，要熟练掌握、灵活运用，但是经典不代表全部，有的题目没有经典算法可以取用，这就需要自行找出问题的关键所在，锻炼自主能力，查阅相关资料，画出求解流程，构思解题算法进行解答，如本书中的迷宫生成和迷宫求解算法。

在构思关键算法的过程中，应结合题目的要求一并进行考虑，这些要求有的体现在时间上，有的体现在空间上，有的则是直接指定了数据结构。有的读者可能认为，题目能够求解出来就好，设置这些要求和限制为解题带来了难度，有的题目直接用数据库解出来很简单，增删改查一条 SQL 语句就完成了。效率由数据库管理系统（DBMS）把关，确实是相当便利。但是，事实上所有的问题都来源于实际，包括题目中的要求和限制，如在单片机的应用中，出于成本控制考虑，CPU 运算能力和存储资源都很有限，没有操作系统，没有 DBMS 怎么办，这就需要从头开始，从最基本的数据结构开始，一步一步地把问题解决。

3. 选择存储结构

算法构思好后，要结合具体的算法选择合适的存储结构，恰当的存储结构能优化算法执行效率并降低算法实现的难度。

数据元素之间的关系在计算机中有两种不同的表示方法：顺序映像和非顺序映像，并由此得到两种不同的存储结构：顺序存储结构和链式存储结构。

顺序映像的特点是借助元素在存储器中的相对位置来表示数据元素之间的逻辑关系。非顺序映像的特点是借助指示元素存储地址的指针表示数据元素之间的逻辑关系，数据的逻辑结构和物理结构是密切相关的两个方面。任何一个算法的设计取决于所选定的数据（逻辑）结构，而算法的实现依赖于采用的存储结构。

例如，在线性结构模型下，可以采用顺序存储结构或链式存储结构来实现，哪种存储结构更加合适，应根据题目的具体要求来确定，若修改、查询较多显然采用顺序存储结构合适，若插入、删除操作频繁则适宜采用链式存储结构。又如，二叉树也可用顺序存储结构和链式存储结构表示，完全二叉树和满二叉树适合采用顺序存储结构，一般的二叉树采用链式存储结构为宜，而链式存储结构又有二叉链表、三叉链表、线索二叉树等表示方法，所以具体设计时应根据算法的实际需求做出选择，需要经常访问双亲节点的算法选择三叉链表存储，经常做二叉树遍历的算法用线索二叉树存储可以提高遍历效率，但线索二叉树在插入、删除节点时较基本的二叉链表复杂。总之，不同的存储结构各有利弊，应有所取舍。

另外，数据结构教科书中介绍的是原始的、抽象的数据结构，这些基本结构需要根据具体求解的问题经过修改、补充后才能直接解题，这个过程往往不是一次性的，而是一个在解题过程中不断修改、不断实现的迭代过程。

4. 设计实现

数据结构课程设计的过程本质上也是软件开发的过程，可以按照软件工程学中软件开发的若干阶段来进行。然而，由于数据结构课程设计中所涉及的问题比较单一，软件规模较小，因此应有所取舍。

1.3 软 件 过 程

软件也是有生命的，其生命周期是从软件的产生到报废的整个过程。软件过程就是在软件的整个生命周期内所涉及的一系列的过程。过程是由活动构成的，而活动又是由任务组成的。任务负责把输入转换为所需要的输出，是最小的执行单元。活动的执行顺序可以是顺序、并发、重复或有选择的执行。在软件工程中，常用软件开发模型来定义软件过程。常用的软件开发模型有：瀑布模型、快速原型模型、增量模型、螺旋模型和喷泉模型。

1. 瀑布模型

瀑布模型是由美国的计算机科学家 Winston W. Royce 在 1970 年发表的著名文章“Managing the Development of Large Software Systems”中首次提到的软件开发模型。瀑布模型的核心观点就是“分解”，它将软件的生命周期划分为不同的活动，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。瀑布模型至少应划分 3 个活动：分析、设计和实施，但通常都将其划分为 5～7 个基本活动，如图 1.3 所示。

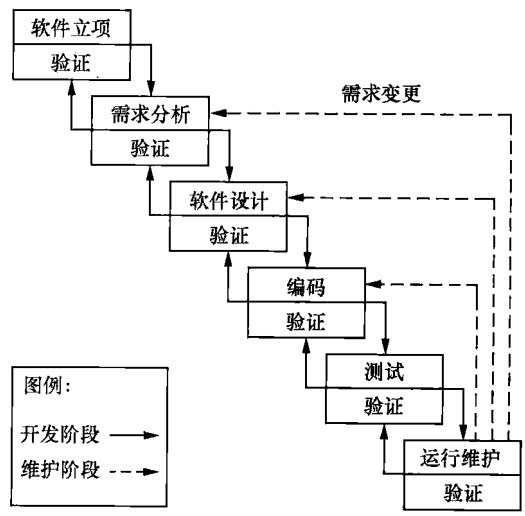

图 1.3 瀑布模型

在瀑布模型中，总是从上一项活动获得当前活动的输入，完成该项活动后输出活动成果，并作为下一活动的输入。同时，需要对该项活动进行评审，若符合要求则继续，否则返回上一项活动甚至更靠前的活动。瀑布模型主要通过控制每项活动的高度和宽度，且在每项活动处设立里程碑，并进行阶段评审的手段控制软件开发的风险。

模型的各项活动目标如下。

（1）软件立项。软件立项主要是确定要解决的问题及问题是否有可行的解决办法。具体来讲就是首先通过调研，确定问题的性质、工程的目标以及工程的规模，并撰写书面报告。然后，从经济、法律、技术 3 个角度论证是否有解决问题的方法，并撰写可行性分析报告。该项活动的输出为：软件立项建议书。

（2）需求分析。需求分析主要是从软件系统的角度分析软件系统应该具备哪些功能。以这些功能为基础构成的概念模型，必须准确地体现用户的需求。在结构化软件开发方法中，概念模型常常体现为实体关系图、数据流图、状态转换图，分别对应数据模型、功能模型、行为模型。该项活动的输出为：软件系统的概念模型、软件需求规格说明书。

（3）软件设计。软件设计主要回答如何实现系统的问题。在该活动中主要完成：建立软件系统的总体结构、定义功能模块的接口、设计全局数据库或数据结构、规定设计约束、实现模块的算法和数据结构、模块的测试方案、编制设计文档等任务。该项活动的输出为：软件系统的逻辑

模型、软件设计规格说明书。

（4）编码。编码主要是依据软件设计规格说明书，采用具体的编程语言逐一实现软件设计规格说明书的要求。一般来讲，在该项活动中，应由编码人员对模块程序进行单元测试，以便验证模块功能及接口与软件规格说明书的一致性，并形成单元测试报告。该项活动的输出为：源代码、单元测试报告。

（5）测试。测试主要是依据需求规格说明书验证已实现的软件系统是否能够满足要求，并撰写测试报告。在该活动中主要完成：模块的装配、集成测试（模块装配过程中的测试）、验收测试（依据需求规格说明书进行逐一验证）。该项活动的输出为：一套完整的软件系统、集成测试报告、验收测试报告。

（6）运行维护。运行维护的主要目标是持久地满足用户的需求。在该活动中主要完成：改正性维护（诊断和改正在使用过程中发现的软件错误）、适应性维护（修改软件以适应环境的变化）、完善性维护（根据用户的要求改进或扩充软件使它更完善）、预防性维护（修改软件为将来的维护活动预先做好准备）。该项活动的输出为：系统的运行报告、维护方案。

模型的优点如下。

（1）简单、活动清晰。瀑布模型将复杂的软件过程转换为一种线性的表示方式，使得软件开发人员容易接受，且各活动的顺序及要完成的任务及目标非常清晰。

（2）对分析、设计、实现进行分离，并尽可能延后实现。瀑布型模型分离了软件过程中的分析、设计、实现，在分析与设计阶段规定仅考虑软件系统的逻辑模型，不涉及软件的物理实现。尽可能延后实现是瀑布模型的主要观点之一。实践表明，对于规模较大的软件项目来说，如果前面阶段的工作没有做好，过早考虑进行程序实现，往往导致大量返工，有时甚至发生无法弥补的问题，带来灾难性后果。

（3）设定里程碑、对每个阶段进行质量控制。为确保目标软件的质量，瀑布模型充分强调了文档在软件过程中的重要性，并且在每项活动结束前都要对所完成的文档进行评审，以便尽早发现问题、改正错误。

模型的缺点如下。

（1）活动间较强的顺序性和依赖性。利用瀑布模型组织软件开发，任何一项活动都必须等前一项活动结束之后，才能开始工作；前一项活动的输出文档就是后一项活动的输入文档。因此，只有前一项活动产生正确的输出，后一项活动的工作才可能获得正确的结果。

（2）风险较大。软件需求的明晰是一个循序渐进的过程，它贯穿软件的整个生命周期。因此，在需求分析阶段完全明晰软件需求，是一件很难完成的事情。但瀑布模型的可逆性较差（每次返工的成本较高）。

（3）客户需要具有极大的耐心。利用瀑布模型组织软件开发，软件的实现放到了最后。因此，软件开发过程中，客户很难尽快见到软件。这种开发模型实际上很难适应目前市场经济条件下的快节奏。

模型适用的开发方法：

瀑布模型适合采用结构化软件开发方法。结构化方法是一种传统的软件开发方法，它是由结构化分析、结构化设计和结构化程序设计三部分有机组合而成的。它的基本思想是把一个复杂问题的求解过程划分为不同的阶段进行处理，每个阶段处理的问题都控制在人们容易理解和处理的范围内。结构化方法的基本要点是：自顶向下、逐步求精、模块化设计。结构化开发方法针对软件生命周期的不同阶段，它有结构化分析（Structured Analysis，SA）、结构化设计（Structured Design，SD）、结构化程序设计（Structured Programming，SP）等方法。因此，瀑布模型非常适合采用结

构化软件开发方法。

瀑布模型的使用范围如下。

（1）用户的需求非常清楚全面，且在开发过程中没有或很少变化。

（2）开发人员对软件的应用领域很熟悉。

（3）用户的使用环境非常稳定，开发工作对用户参与的要求很低。

2. 快速原型模型

一方面，在软件开发之初，用户并不十分清楚需求；另一方面，软件需求规格说明书是一个不可执行的电子文档，通过需求评审发现错误的难度较大。而需求是软件开发中非常重要的活动，尽快明确需求是每一个开发人员和用户最迫切的愿望。为了克服瀑布模型的缺点，美国人 James Martin 于 1991 年提出了一种用于描述软件过程的开发模型——快速原型法，该模型是增量模型的一种。

增量模型和瀑布模型之间的本质区别是：瀑布模型是一种整体开发模型，它规定在开始下一项活动之前，必须完成前一项活动的所有细节。而增量模型是一种非整体开发模型，它推迟某些阶段或所有阶段中的细节，从而较早地产生工作软件。增量模型是在软件的开发过程中以一系列的增量方式开发软件系统。增量方式包括：增量开发、增量提交。增量开发是指在软件的开发周期内，以一定的时间间隔开发部分工作软件；增量提交是指在软件开发周期内，以一定的时间间隔以增量方式向用户提交工作软件及相应文档。增量开发和增量提交可以同时使用，也可单独使用。

原型是一个实际可以运行、容易修改，且可以不断完善的系统。快速原型法的核心是以少量的代价快速地构造一个可执行的软件系统模型，使用户和开发人员可以较快地明晰需求。具体来讲，就是在初步了解用户的基本的、关键的需求后，开发人员首先建立一个自认为符合用户要求的模型系统，并交给用户使用。由于该方法采用一个交互的、快速建立起来的原型系统取代了形式化的、不可执行的软件需求规格说明书，因此用户可以快速地向开发人员提供真实的、具体的软件需求。

快速原型法分为如下两类。

（1）快速需求规格原型。该原型主要反映系统的某一个侧面，它可以密切用户和开发人员的关系，促进相互理解，有助于获得更完整、更精确的软件需求规格说明书。一般地，在软件需求规格说明书确定后，这个模型也就被丢弃了，后面的开发仍然按照瀑布模型进行。该方法非常侧重系统的可测试性、可理解性等。快速需求规格原型如图 1.4 所示。

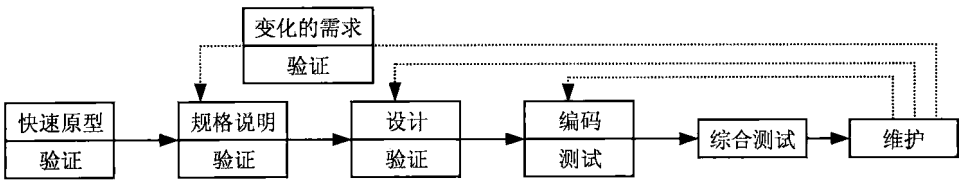

图 1.4　快速需求规格原型

软件的开发需求通常可归为界面需求、功能需求和性能需求，因此，建立的原型模型一般也可分为用户界面原型、功能原型和性能原型。对交互式系统来说，用户最关心的往往是软件的交互界面，包括菜单格式、输出格式、操作命令的使用等，建立用户界面原型时可以忽略功能、性能方面的需求。功能原型一般侧重实现系统最重要的功能，此时可以忽略出错处理、性能等方面的需求。而对实时系统来说，某些性能指标往往是非常关键的，如响应时间、吞吐量等，此时可以建立性能原型，它应该实现系统中最耗时、最占空间的那些功能，而对用户界面和其他功能则可以忽略。

（2）快速渐进原型。该方法主要通过对系统原型的持续精化，将系统需具备的性质逐步增加，直至所有的性质都被满足。此时，该原型也就成了最终系统。这是一种交互式、增量式的开发，在开发过程中，开发人员始终与用户保持密切联系，它并不要求一次就做出最终系统，而是要经过多次反复和试探，所以更符合人类的认识规律。快速渐进原型如图 1.5 所示。

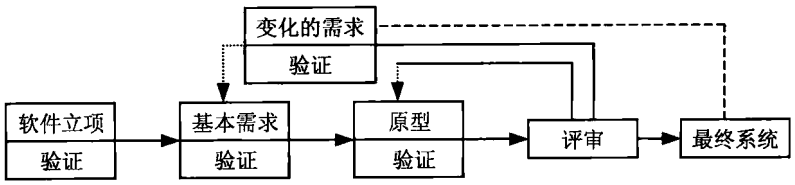

图 1.5　快速渐进原型

例 1.3　利用快速原型法开发一个教学软件系统。

项目需求的简单分析：学习本身是一个复杂的、不可预知的过程。针对不同的人、不同的情景有着非常不同的结果。因此，教学软件的需求分析工作也是较为困难的，其明晰需求的过程也是比较长的。

采用快速原型法开发教学软件系统的优点如下。

（1）可以对教学策略进行较早的评价。

（2）教学软件的交互性是比较复杂的，可利用界面原型和功能原型探索教学软件的交互方式及外观。

（3）能够有效沟通开发人员和教学设计人员以及课件用户之间的思想，是实现教学思想、教学经验与计算机技术统一和结合的基础，符合教学设计和软件过程天生的重复和迭代特征。

（4）原型是进行教学试用、教学效果评价的最基本条件，它为教学软件的快速、高质量开发能够起到不可替代的作用。

采用快速原型法开发的基本步骤如下。

（1）快速分析：根据要开发教学软件的学科教学特点和教学要求，决定原型要着重设计的方面，然后选定合适的软件开发工具，同时确定原型的使用目的是快速需求规格原型还是快速渐进原型。

（2）构造原型：根据设计要求，利用选定的软件开发工具制作出所选定教学内容的教学模块的外观模型。

（3）运行评价：运行原型，通过学科教师、教育专家以及学生的检验、评价和测试，针对原型提出修改意见和需求。

（4）修正改进：根据修改意见修正和完善原型，直至符合教学需求。

模型的优点如下。

（1）容易掌握和接受。原型法的循环反复、螺旋式上升的方法符合人类认识事物的规律。因此，该方法更容易被人们掌握和接受。

（2）有效降低了由于软件需求不明而引发的开发风险。快速原型法强调用户的参与，将模拟手段引入系统分析的初期阶段，特别是对模拟的描述和系统运行功能的检验，都强调用户的主导作用。用户与开发者可以及时沟通，信息反馈及时准确，潜在的问题能够尽早发现、及时解决，增加了系统的可靠性和实用性。

（3）便于系统的交付。快速原型法实际上是将需求调研、分析和软件设计合而为一，使用户一开始就能看到最终系统的概貌。此外，用户全程参与系统开发，消除了心理负担，可以提高对系统功能的理解，有利于系统的移交、运行和维护。

模型的缺点如下。

（1）在大型软件系统中应小心使用。由于大规模的软件系统，本身具有高度的复杂性，很难在较短的时间内获得一个较为满意的原型，而且系统本身的复杂性就意味着迭代次数大幅度增加。因此，在大规模系统中较为可取的方法是采取“分解”的思想，首先将大系统划分为几个较小的系统，然后针对每个小系统实施快速原型法。

（2）开发过程管理工作较为复杂。因为快速原型法的整个开发过程要经过“修改—评价—再修改”的多次反复才能达到最终系统，所以极大地增加了管理负担。

（3）开发人员容易将系统原型取代需求分析和软件设计。往往由于开发时间、成本以及懒惰等因素开发人员很容易犯投机取巧的毛病，利用原型系统直接取代需求分析和软件设计阶段的工作。

（4）缺乏规范化的文档资料。在瀑布模型中，文档作为每一项活动的重要成果，对软件的开发有着非常重要的作用，但文档的工作量是较大的。而在快速原型模型中，开发人员往往认为通过原型即可弄清系统各方面的要求，从而忽略文档的作用。实际上，不论何种开发模型，开发文档都应该引起足够的重视。

快速原型模型的使用范围如下。

（1）对所开发的领域比较熟悉而且有快速的原型开发工具。

（2）项目招投标时，可以以原型模型作为软件的开发模型。

（3）进行产品移植或升级时，或对已有产品原型进行客户化工作时，原型模型是非常适合的。

3. 渐增模型

渐增模型也是增量模型的一种，渐增模型有如下两类。

（1）增量构造模型：其核心特征就是融合瀑布模型和增量开发，即对某些阶段采用瀑布模型，而另一些阶段则采用增量开发。增量构造模型如图 1.6 所示。使用该模型组织开发软件时，是在详细设计、编码、测试等阶段引入增量开发，把软件产品作为一系列的增量构件来设计、编码、集成和测试。

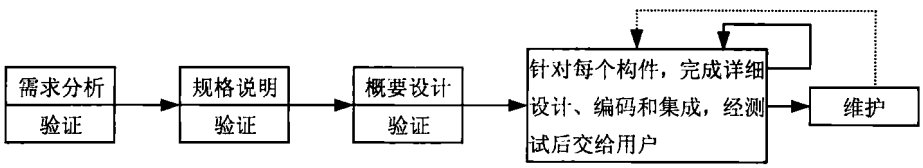

图 1.6　增量构造模型

例 1.4　采用增量构造模型开发字处理软件。

首先，按照瀑布模型的要求，完成了需求分析、规格说明、概要设计三项活动，获得字处理软件应具备的文件管理、编辑、文档生成、拼写、语法检查和页面排版功能。然后，按增量开发方式，完成每个构件的详细设计、编码、集成和测试工作。

第 1 个增量构件：提供基本的文件管理、编辑和文档的生成功能。

第 2 个增量构件：提供更完善的编辑和文档生成功能。

第 3 个增量构件：实现拼写和语法检查功能。

第 4 个增量构件：完成高级的页面排版功能。

（2）演化提交模型：其核心特征就是融合瀑布模型、增量开发和增量提交的一个软件过程。与增量构造模型不同的是，它引入了增量包的概念，针对每个增量包采用瀑布模型组织开发，经过不断的迭代形成最终产品。演化提交模型如图 1.7 所示。该模型的软件过程是一个随时间变化而不断交错的线性序列，每一个线性序列都会产生软件的一个可发布的“增量”。一般来说，在采

用该模型组织软件开发时，第 1 个增量产品往往是软件系统的核心功能，即第 1 个增量产品实现了基本的、关键的、具有鲜明特色的软件需求，而忽略非关键的、补充性质的软件需求。用户对每一个增量产品的使用、评估都会作为下一个增量产品发布的新特征和功能，在每一个增量发布后不断重复该过程，直到产生最终的软件系统。

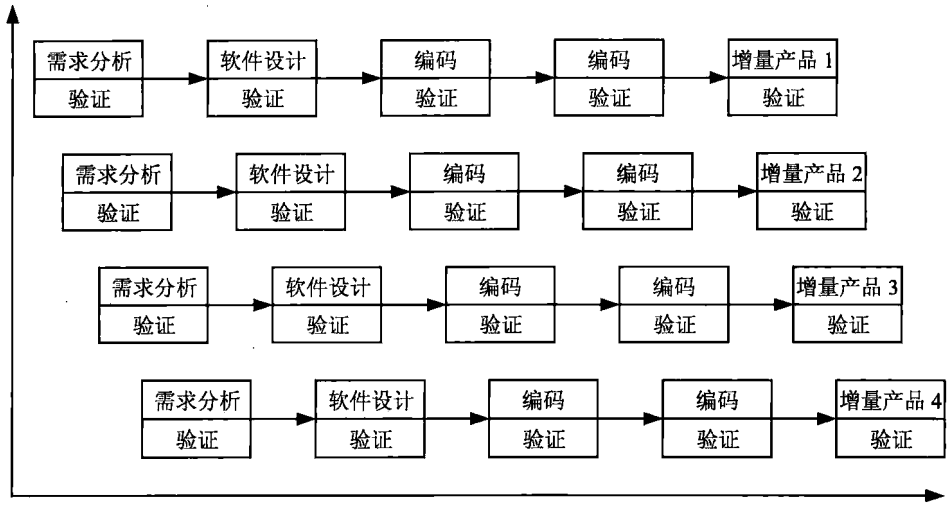

图 1.7　演化提交模型

例 1.5　采用演化提交模型开发字处理软件。

首先，发现增量包：文件管理、编辑和文档，按照瀑布模型的要求，完成该增量包所有活动，并发布增量产品 1。其次，发现增量包：拼写、语法检查以及用户对编辑和文档生成功能的反馈意见，完成该增量包的所有活动，并发布增量产品 2。最后，发现增量包：页面排版，按照瀑布模型的要求，完成该增量包的所有活动，并发布增量产品 3，形成最终产品。

模型的优点如下。

（1）资源分配方式灵活。不管是增量构造模型，还是演化提交模型，其活动都具有较大的并行性和重复性。因此，在对软件开发资源的配置上要求不高，可以节约资源，提高了资源配置的灵活性。

（2）风险控制能力较强。可先实现软件系统基本的、关键的和具有鲜明特色的需求，试探产品的受欢迎程度，从而决定下一步软件开发的计划。

（3）增强用户的信心。不管是增量构造模型，还是演化提交模型，都引入了提前实现软件的理念。因此，与其他开发模型相比，在采用该模型组织开发软件时，用户能够较早地看到开发成果，从而起到镇静剂的作用，增强用户的信心。

模型的缺点如下。

（1）软件的体系结构要求开放式的。不管是增量构造模型，还是演化提交模型，软件的各组成部分是逐渐被加入，加入要求不破坏原有系统。因此，在使用该模型组织开发软件时，对软件的开放性要求较高。

（2）易失去对软件开发过程管理的整体性。渐增模型的灵活性使得软件过程适应软件需求变化的能力大大优于瀑布模型，但很容易陷入边做边改的怪圈中，从而使软件过程的控制失去整体性。

（3）增量包间的交集处理较为困难。在演化提交模型中，渐增模型引入了增量包的概念，尽管该做法克服了瀑布模型要求一次性确定完整需求的缺点，但是如果增量包之间存在相交的情况

且未很好处理，则必须做全盘系统分析。

渐增模型的使用范围如下。

（1）该模型适合对已有的软件产品升级或新版本开发的任务。

（2）若用户对软件产品的完成期限要求比较苛刻，可以采用渐增模型。

（3）若软件开发人员对所开发软件产品涉及的业务领域非常熟悉而且已开发完成一个原型系统，则采用增量模型也非常适合。

4. 螺旋模型

螺旋模型是美国工程师 Barry Boehm 在 1986 年发表的一篇文章“A Spiral Model of Software Development and Enhancement”中提出的。该模型的基本原理是在瀑布模型的每一个开发活动中引入风险分析机制，以求尽量降低软件的开发风险。螺旋模型的基本原理如图 1.8 所示。

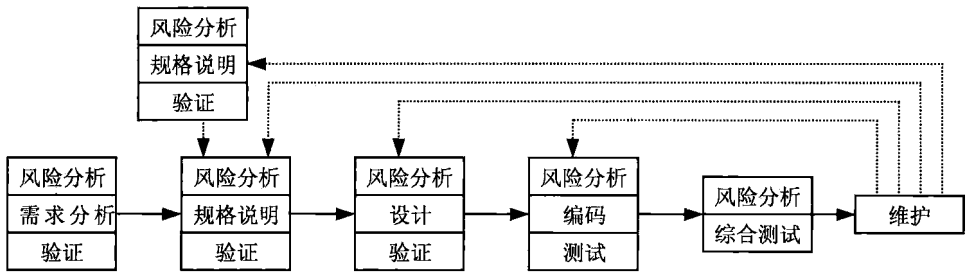

图 1.8　螺旋模型的基本原理

螺旋模型将瀑布模型与原型的迭代特征结合起来，并加入两种模型均忽略了的风险分析，弥补了两者的不足。螺旋模型在每一个迭代周期内都包含 4 个不同的阶段：需求定义、风险分析、工程实现和评审阶段。完整的螺旋模型如图 1.9 所示。

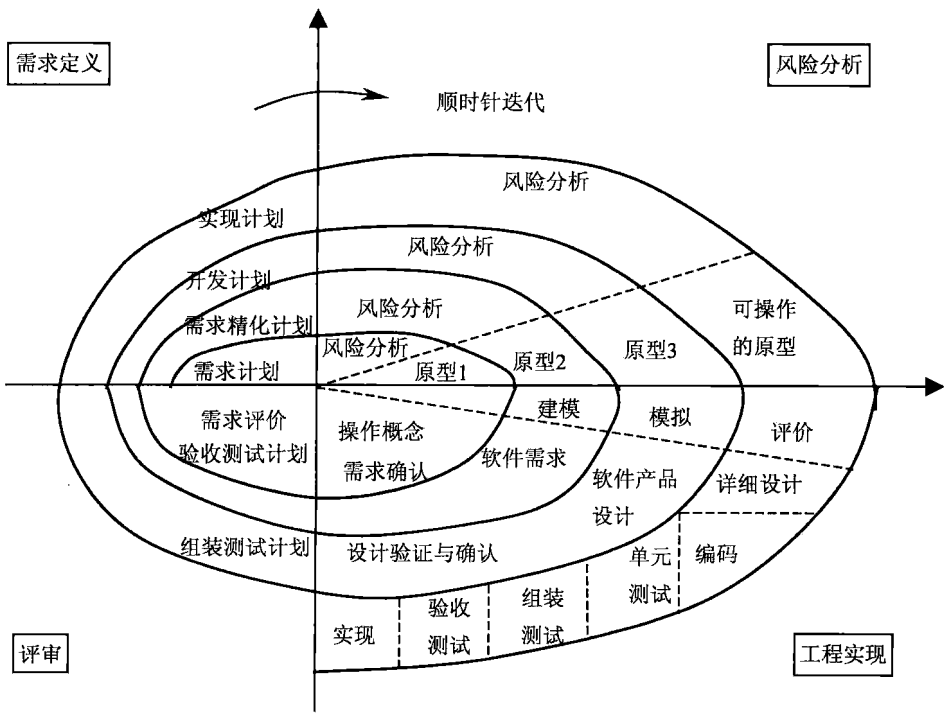

图 1.9　完整的螺旋模型