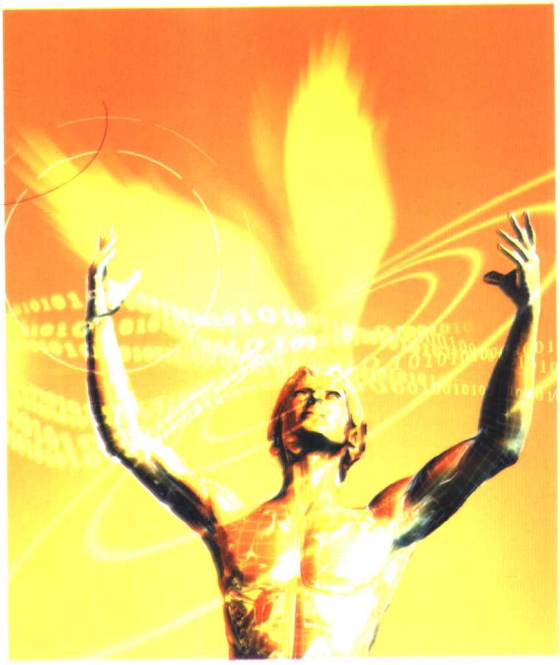
21世纪高等院校
计算机专业
规划教材

# 现代软件工程学

钟 珞 主编　　潘 昊 副主编



国防工业出版社
http://www.ndip.cn

21世纪高等院校计算机专业规划教材

# 现代软件工程学

钟 珞 主 编
潘 昊 副主编

国防工业出版社
·北京·

# 内容简介

本书作为普通高等院校计算机专业研究生的专用教材，从实用角度系统地阐述了现代软件工程学的基本原理、概念和技术方法。

全书共11章，第1章介绍了软件工程的基本概念；第2章阐述了现代软件的需求模式；第3章介绍了现代软件的体系结构；第4章介绍了面向对象的开发方法；第5章介绍了UML软件工程；第6章介绍了分布式系统的软件开发技术；第7章介绍了基于构件的软件开发；第8章介绍了CORBA和DCOM技术；第9章介绍了基于Java的软件开发技术；第10章介绍了现代软件测试技术；第11章进行了典型实例分析。

本书是一本注重系统性、科学性的教材，内容丰富，实用性强，可作为计算机专业和信息类专业及其他相关专业的研究生教材，也可作为高级软件开发人员的技术参考书。

**图书在版编目(CIP)数据**

现代软件工程学/钟珞主编．—北京：国防工业出版社，2004.8

ISBN 7-118-03552-1

Ⅰ.现… Ⅱ.钟… Ⅲ.软件工程 Ⅳ.TP311.5

中国版本图书馆CIP数据核字(2004)第068179号

国防工业出版社出版发行

(北京市海淀区紫竹院南路23号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 16¼ 374千字

2004年8月第1版 2004年8月北京第1次印刷

印数：1—4000册 定价：22.00元

# 前　言

21 世纪是信息社会高速发展的世纪，软件作为信息技术的核心起着至关重要的作用。面对计算机越来越广泛的应用需求，研究如何更快、更好、更方便地开发出各种不同类型、不同目的的软件，这就是软件开发技术和软件工程技术所要解决的问题。

50 多年来，随着计算机系统的发展，软件的开发技术也在发生变化。软件工程首先是为了解决软件危机而提出的，希望用成功的、卓越的开发经验来指导软件开发，通过类似于工业化的管理，提高开发人员的技术水平。

20 世纪 90 年代以来，软件工程不仅从方法论的角度为管理人员和开发人员提供可见的结构和有序的思考，而且大量的成功软件总结出的设计经验，使软件开发人员可以充分利用设计模式、框架、部件等。同时，计算机网络环境提供了经济的全球化的倾向，硬件技术也在快速发展。这些新的需求要求软件能具备充分的适应能力，去适应各种不同类型的连通和变动要求。

本书是一本注重系统性、科学性的新编教材，其基本内容是从实用角度讲述软件工程的基本原理、概念和技术方法，包括：

（1）软件生命周期、面向对象的设计模式与软件开发方法。

（2）基于部件的开发方法。

（3）CORBA 和 DCOM 技术、基于 Java 的部件对象技术等。

我们将重点论述对象的软件分析与设计方法，始终以新的软件理论、方法做背景。为此，本书的主要特色如下：

1．以面向对象软件开发方法为主线进行介绍

面向对象技术的最基本的特点是尽可能模拟人的思维方式，随着对某个系统的需求逐步具体的过程而逐步地设计和实现这个系统。它具有模块性、继承性、动态连接性和易维护性，对软件开发方法有着重要的影响。我们将重点论述面向对象的软件分析和设计方法。

2．形式化方法和经典软件开发方法，以及面向对象方法的有机结合

形式化方法、面向操作（行为）的经典软件开发方法在软件开发中曾经获得过巨大的成功，但都存在某些方面的不足。将面向对象方法与上述结合起来，可以有效地取长补短，发挥各自的优势。

3．考虑软件生命周期的全过程以及软件集成的方法

软件是一种产品，不同的软件采用不同的生命周期模型；编好的软件在实现阶段要考虑到软件的集成问题。所以，在开发软件时，必须兼顾软件生命周期和软件集成策略，以期在较短的时间里能迅速开发出高质量的软件产品。

4．基于分布式系统的部件技术

在分布式的网络计算中，创建和利用可复用的软件部件来解决应用软件的开发问题，可以提高开发速度、降低开发成本、增加应用软件的灵活性、降低软件维护费用，是目前

发展最快的软件复用方式。

5. 结合 Java 程序语言进行实现

Java 是应用于 WWW 的最好的语言，它遵循了软件工程的原则，是完全面向对象的开发语言。因为将来的发展方向是基于网络开发，所以，我们选择 Java 语言来实现软件，并提供一个较为完整的典型软件开发实例。

长期以来，人们开发优质软件的能力大大落后于计算机硬件技术日新月异的发展，这种状况已经严重妨碍了信息技术的进步。因此，如何以较高的质量和较低的成本来满足新时代对软件开发和维护所提出的要求已迫在眉睫。毫无疑问，软件工程学在计算机专业及信息类专业的课程中占有非常重要的地位，它对学生将来从事计算机研究和开发工作具有很大的指导作用。

我们从 1985 年开始一直从事高年级本科生及研究生的《软件开发技术》和《软件工程学》及相关课程的教学，积累了丰富的教学经验和教学心得。同时，我们还承担了一些相关的科研项目的开发和研制，从软件的分析、设计、开发到维护，对软件工程现有的体系结构有较全面的认识，并对一些新情况、新方法进行了探索，通过实验不断得到反复认证。这些使我们的讲课内容得到了充实，使得抽象的软件工程的教学变得丰富、具体，教学效果良好。

目前，国内有关软件工程技术与设计方面的资料很多，而教学任务要求尽快将众多新理论、新技术的内容整理出版，以供急需。因时间和水平有限，一定有许多不周到和不准确之处，恳请专家学者提出意见和建议，以便进一步完善。

本书由武汉理工大学钟珞教授任主编，潘昊副教授任副主编，参加编著工作的有钟珞、潘昊、张开松、冯珊、沈琦、田捷、李三德。

编 著 者

2004 年 8 月于武昌马房山

# 目　录

# 第1章 引 论

## 1.1 软件工程概述

软件工程一词是1968年北大西洋公约组织的计算机科学家在当时联邦德国召开的专门讨论解决“软件危机”的国际会议上正式提出并使用的，并由此诞生了一门新兴学科——软件工程学。软件工程学是一门交叉学科，它涉及计算机科学、管理科学、工程学和数学。

软件工程研究的对象是大型软件系统的开发过程，它研究的内容是生产流程、各生产步骤的目的、任务、方法、技术、工具、文档和产品规格。软件工程是指导计算机软件开发和维护的工程学科，采用工程的概念、原理、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来。

从软件的生产管理技术来分，软件工程大致可分为：软件的需求定义和分析技术，软件设计技术和设计审查技术，软件设计表现技术，软件测试技术，软件可靠性理论及其评价性方法，软件扩充和维护，软件成本估算。

从软件工程的内容上划分，软件工程分为理论、结构、方法、工具等部分。理论和结构是方法论与工具的基础和前提。在方法论与工具方面存在着这样的关系：为了找出行之有效的方法就应该具备适合的工具；反之，为了制造、改进软件工具，也应该先有合适的方法。

软件工程主要研究如何以较少的代价获得高质量的软件，要达到这个目的就必须研究软件开发方法和软件开发工具。随着计算机应用日益普及和深化，计算机软件成本逐年上升，软件开发的生产率也远远跟不上普及计算机应用的要求。为了解决这些问题，制定了软件工程的基本目标：

（1）开发尽可能多的软件产品；

（2）提高软件的生产效率；

（3）满足应用的功能需要；

（4）降低软件开发成本；

（5）能按时、按质完成软件开发任务。

自1968年“软件工程”的概念提出以来，专家学者又陆续提出了100多条关于软件工程的准则。著名软件工程专家B.W.Boehm于1983年发表的一篇论文中提出了软件工程的七条基本原理。他认为这七条原理是确保软件产品质量和开发效率的最小准则集合。

（1）分阶段的生命周期计划严格管理；

（2）坚持进行阶段评审；

（3）执行严格的产品控制；

（4）采用现代程序设计技术；

（5）结果应能清楚地审查；

（6）开发小组人员少而精；

（7）承认不断改进软件工程实践的必要性。

这七条原理是相互独立的，其中任意六条原理的组合都不能替代另一条原理，因此，它们是缺一不可的最小集合，然而这七条原理又是相当完备的，人们虽然不能用数学方法严格证明它们是一个完备的集合，但是，可以证明在此之前已经提出的一百多条软件工程原理都可以由这七条原理的任意组合蕴含或派生。[1,2]

## 1.2 软件工程模式

### 1.2.1 传统软件工程模式

20世纪70年代，软件工程普遍采用生命周期方法学，即把软件生存周期划分成若干个阶段，如软件定义阶段、开发阶段和维护阶段。每个阶段的任务相对独立，而且比较简单，便于不同人员分工协作，从而降低整个软件开发工程的困难程度。在实现每个阶段的任务时，采用的是系统化的技术方法——结构化分析和结构化设计技术。

软件定义阶段的任务是确定软件开发工程必须完成的总目标，确定工程的可行性，导出实现工程目标应该采用的策略及系统必须完成的功能，估计完成该工程需要的资源和成本，并且制定工程进度表。概括来说就是问题定义、可行性研究和需求分析三个阶段。这个时期的工作通常又称为系统分析，由系统分析员负责完成。

开发阶段具体设计和实现前一个时期定义的软件，它通常由四个阶段组成：总体设计、详细设计、编码和单元测试、综合测试。其中前两个阶段又称为系统设计，后两个阶段又称为系统实现。

维护阶段的主要任务是使软件持久地满足用户的需求。具体地说，当软件在使用过程中发现错误时应该加以改正；当环境改变时应该修改软件以适应新的环境；当用户有新要求时应该及时改进软件以满足用户的新需求。[1, 2, 3]

在实际开发过程中，我们不可能直线性地通过分析、设计、编程和测试等阶段，各阶段之间的回复是在所难免的。

软件生存周期模型是软件工程中一个重要的概念。基于这种软件生存周期的模型包括：瀑布模型、渐增模型、演化模型、螺旋模型、喷泉模型和智能模型。其中又以瀑布模型(Waterfall Model)最为流行，如图1-1所示。因为一方面它在支持开发结构化软件、控制软件复杂程度、促进软件开发工程化方面起了显著作用，另一方面它为软件开发和维护提供了一种当时较为有效的管理模式。

这种瀑布模型具有如下几个特点：

（1）瀑布模型具有顺序性和依赖性，即后一阶段的工作必须在前一阶段的工作完成后才能开始；

（2）把逻辑设计与物理设计清楚地划分开，是瀑布模型的重要指导思想；

（3）瀑布模型强调的是优质，即每一步都循序渐进，及早消除隐患，从而保证软件质量。

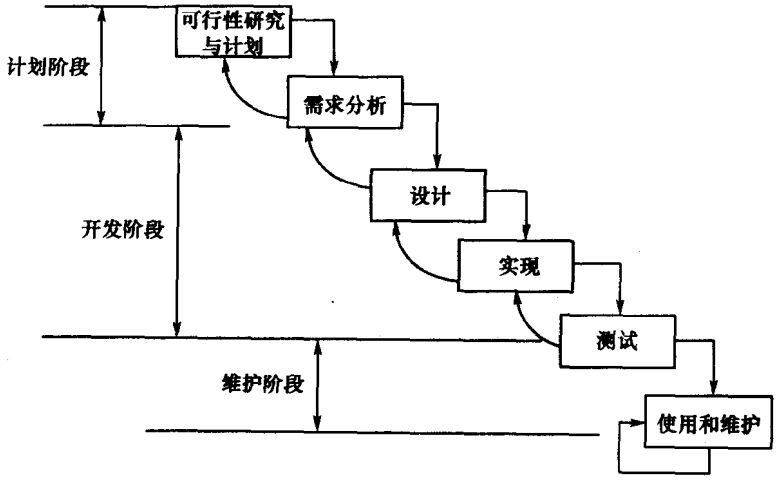



图 1-1 瀑布模型

但它有个致命缺点是只有做出精确的需求分析，才能取得预期的结果。由于各种客观、主观的原因，需求分析往往不很精确，常常给日后的开发带来隐患。

基于这种生命周期模型我们通常采取的是传统的开发方法——结构化开发方法，它的基本要点有如下几点：

（1）自顶向下。将复杂的大问题，分解为小问题，找出问题的关键、重点所在，同时找出技术难点来，然后用精确的思维定性、定量地描述问题。该要点的核心问题是“分解”，即如何划分，采用什么准则。实现的手段是模块化。

（2）逐步求精。将现实世界的问题经抽象转化为逻辑空间或求解空间的问题。复杂问题经抽象化处理变为相对较简单的问题。经几次抽象处理，最后到求解域中只是非常简单的编程问题。

（3）模块化设计。模块化就是把程序划分为若干个模块，而每个模块完成一个子功能，把这些模块汇总起来构成一个有机整体，即可完成指定的功能。

（4）结构化编码。结构化编码的方法强调清晰简洁，它是一种构造程序的技术，有利于提高软件生产率及降低软件维护代价。原则上是尽量使用标准库函数，程序讲究清晰，避免过于精巧。

（5）主程序员组织。主程序员，即组织负责人，是程序生产过程中的总体设计师。程序员，按任务书要求编程，是程序生产线上的“工人”。测试工程师，负责系统测试，是程序生产过程中的检验员。

（6）结构化设计 SD。结构化方法的体系结构是：

① 结构化分析(SA—Structure Analysis)。SA 方法是建立在自顶向下、逐步求精思想基础上的分析方法，它的要点是分解和抽象。经过一系列分解和抽象，到最底层的问题已经是很容易求解的了。

② 结构化设计(SD—Structure Design)。SD 方法是由 IBM 公司的 Constentine 等人花了十几年时间研究出来的一种程序设计方法。SD 是一种用于概要设计的方法，与 SA 方

法配合使用。其目标是建立一个结构良好的软件系统。SD 方法的基础是数据流程图，因此也称为面向数据流的设计方法。

③ 结构化程序设计(SP—Structure Programing)。把一个复杂问题的求解过程分阶段进行，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。

基于以上的这些特点，传统的软件开发方法有以下几种：

**1. Parnas 方法**

该方法是由 D．Parnas 在 1972 年提出的。Parnas 的方法提出两个原则问题。

（1）信息隐蔽原则：在概要设计时列出将来可能发生变化的因素，并在模块划分时将这些因素放到个别模块的内部。这样，在将来由于这些因素变化而需修改软件时，只需修改这些个别的模块，其他模块不受影响。信息隐蔽技术不仅提高了软件的可维护性，同时也避免了错误的蔓延，改善了软件的可靠性。现在信息隐蔽原则已成为软件工程学中的一条重要原则。

（2）第二条原则是在软件设计时应对可能发生的种种意外故障采取措施。软件是很脆弱的，很可能因为一个微小的错误而引发严重的事故，所以必须加强防范。

Parnas 对软件开发提出了深刻的见解。但是，他没有给出明确的工作流程。所以这一方法不能独立使用，只能作为其他方法的补充。

**2. SASD 方法**

1978 年，E．Yourdon 和 L．L．Constantine 提出了结构化方法，即 SASD 方法，也可称为面向功能的软件开发方法或面向数据流的软件开发方法。1979 年 TomDeMarco 对此方法作了进一步的完善。它首先用结构化分析(SA)对软件进行需求分析，然后用结构化设计(SD)方法进行总体设计，最后是结构化程序设计(SP)。这一方法不仅开发步骤明确，SA、SD、SP 相辅相成，一气呵成，而且给出了两类典型的软件结构(变换型和事务型)，便于参照，使软件开发的成功率大大提高，从而深受软件开发人员的青睐。

**3. 面向数据结构的软件开发方法**

它包含了两种方法：

（1）Jackson 方法。1975 年，M．A．Jackson 提出了一类至今仍广泛使用的软件开发方法。这一方法从目标系统的输入、输出数据结构入手，导出程序框架结构，再补充其他细节，就可得到完整的程序结构图。这一方法对输入、输出数据结构明确的中小型系统特别有效。

（2）Warnier 方法。1974 年，J．D．Warnier 提出的软件开发方法与 Jackson 方法类似。差别有三点：一是它们使用的图形工具不同，分别使用 Warnier 图和 Jackson 图。另一个差别是使用的伪码不同；最主要的差别是在构造程序框架时，Warnier 方法仅考虑输入数据结构，而 Jackson 方法不仅考虑输入数据结构，还考虑输出数据结构。

**4. 问题分析法(PAM)**

PAM(Problem Analysis Method)是 20 世纪 80 年代末由日立公司提出的一种软件开发方法。PAM 方法希望能兼顾 Yourdon 方法、Jackson 方法和自底向上的软件开发方法的优点，而避免它们的缺陷。它的基本思想是：考虑到输入、输出数据结构，指导系统的分解，在系统分析指导下逐步综合。这一方法的具体步骤是：从输入、输出数据结构导出基本处理框，分析这些处理框之间的先后关系，按先后关系逐步综合处理框，直到画

出整个系统的PAD图。从上述步骤中可以看出，这一方法本质上是综合的自底向上的方法，但在逐步综合之前已进行了有目的的分解，这个目的就是充分考虑系统的输入、输出数据结构。

PAM方法的另一个优点是使用PAD图。这是一种二维树形结构图，是到目前为止最好的详细设计表示方法之一，远远优于NS图和PDL语言。

这一方法在日本较为流行，软件开发的成功率也很高。由于在输入、输出数据结构与整个系统之间同样存在着鸿沟，这一方法仍只适用于中小型问题。

### 1.2.2 现代软件工程模式

现代软件工程是在传统软件工程模式的基础上，为了强调人在系统开发中的作用，同时为了适应软件新技术的发展趋势而提出的。其基本要点如下：

（1）以人为主，充分利用软件开发方法及软件开发工具。

（2）开发人员的组织管理对软件开发成功与否至关重要。

（3）基于软件组件的软件开发技术。各种功能的可重用软件组件不断问世。这使得在软件开发过程中编程工作量日趋减少，取而代之的是在设计好系统体系结构后，利用软件组件构造或重构软件系统。

（4）软件组件是标准化设计、成品化生产的，极易构造使用，从而大大简化了设计、编程、测试各个环节的工作量，提高了工作效率和生产效率。 由于在软件开发过程中最大限度地采用软件组件，使得软件开发过程变为系统分析、系统构造、系统测试的集成过程。

这里涉及到软构件和软件复用的概念，软构件(Software Component)就是将具有一定集成度并可以重复使用的软件组成单元。软件复用可以表述为：构造新的软件系统可以不必每次从零做起，直接使用已有的软构件，即可组装(或加以合理修改)成新的系统。从软件工程的角度来看，软件复用发生在构造新软件系统的过程中，即在一个程序的构造期间，对已有代码的使用就是软件复用，但在程序执行期间重复调用某段代码(如循环语句的循环体)，则不属于软件复用。复用方法合理化并简化了软件开发过程，减少了总的开发工作量与维护代价，降低了软件的成本又提高了生产率。另一方面，由于软构件是经过反复使用验证的，自身具有较高的质量，因此由软构件组成的新系统也具有较高的质量。

软构件的开发与运用可以说刚刚开始。在一些公共领域，例如软件的用户界面，通用软构件的使用已经屡见不鲜。国内许多单位在开发软件时，已经普遍使用Delphi、PowerBuilder等环境和工具，大大提高了工作效率。

由于软构件的使用可以渗透到符合软构件标准规范的所有系统中，不必耗费巨资开发自己的系统去与大公司竞争，即使中小公司也可以在市场中找到自己的位置，从而将给这些国家和地区的软件产业带来另一个新的发展机会。

现代软件工程与传统的软件工程在阶段的划分上既有联系又有区别。它的主要任务有以下几个方面：

1. 系统分析

从系统需求入手，从用户观点出发建立系统用户模型。用户模型从概念上全方位表

达系统需求及系统与用户的相互关系。系统分析是在用户模型的基础上，建立适应性强的独立于系统实现环境的逻辑结构。分析阶段独立于系统实现环境，可以保证建立起来的系统结构具有相对的稳定性，便于系统维护、移植或扩充。

**2. 系统设计**

在系统设计阶段，首先考虑具体的实现环境。在设计时，可能会对系统结构做一些调整，但为了保持系统结构的稳定性，应尽可能避免由于实现环境的特定要求而改变系统结构。对于复杂的大系统，还可以根据组件之间关联的紧密程度，将关联密切的多个组件形成一个子系统，子系统之间具有松散的耦合。在系统实现阶段，对于需要开发的软件组件，选择采用某种合适的程序设计语言编写相应的源代码程序，完成系统实现工作。

**3. 系统测试**

测试包括单元测试、集成测试和系统测试。就功能而言与传统软件工程模式中系统测试的功能相同。

**4. 软件组件**

在现代软件工程的开发过程中，软件组件只是一个辅助或支撑系统构造的过程。软件组件开发主要是开发与维护系统构造过程中用到的组件。将软件组件作为一个单独的过程，目的是将组件作为构造软件的“零部件”。随着软件技术的不断发展及软件工程的不断完善，软件组件将会作为一种独立的软件产品出现在市场上，供应用开发人员在构造应用系统时选用。

**5. 系统开发人员的组织管理**

现代软件工程不仅包括软件开发方法、工具和过程，更强调人在开发过程中的作用。一个复杂的系统开发过程，涉及到众多的以人为主的各种开发活动，通过这些活动的有机配合与协调才能保证系统开发的成功。因此，系统开发人员的组织管理是现代软件工程中的重要方面。

组织管理方法有以下几个要点：

（1）明确系统开发人员与用户之间的责任与义务；

（2）明确各类开发人员的主要工作及责任；

（3）制定或选择工程开发规范。

在新的软件开发模式上发展出了与之相配合的软件开发方法，即目前广受计算机软件界青睐的主流开发方法——面向对象的方法。

面向对象方法是一种把面向对象的思想应用于软件开发过程中，指导开发活动的系统方法，简称 OO 方法，它是建立在对象概念(对象、类和继承)基础上的方法。面向对象技术是软件技术的一次革命，在软件开发史上具有里程碑的意义。随着 OOP(面向对象编程)向 OOD(面向对象设计)和 OOA(面向对象分析)的发展，最终形成面向对象的软件开发方法 OMT(Object Modeling Technique)。这是一种自底向上和自顶向下相结合的方法，而且它以对象建模为基础，从而不仅考虑了输入、输出数据结构，实际上也包含了所有对象的数据结构。所以 OMT 彻底实现了 PAM 没有完全实现的目标。不仅如此，OO 技术在需求分析、可维护性和可靠性这三个软件开发的关键环节和质量指标上有了实质性的突破，彻底地解决了在这些方面存在的严重问题。

OMT 的基础是对象模型，每个对象类由数据结构(属性)和操作(行为)组成，有关的所有数据结构(包括输入、输出数据结构)都成了软件开发的依据。OMT 解决了需求分析这一问题，因为需求分析过程已与系统模型的形成过程一致，开发人员与用户的讨论是从用户熟悉的具体实例(实体)开始的。OMT 的基础是目标系统的对象模型，而不是功能的分解。因此当需求变化时对象的性质要比对象的使用更为稳定，从而使建立在对象结构上的软件系统也更为稳定，大大改善了可维护性。[4,5,6]

另一种开发方法——可视化开发方法是 20 世纪 90 年代软件界最大的两个热点之一。随着图形用户界面的兴起，用户界面在软件系统中所占的比例也越来越大，有的甚至高达 60%～70%。产生这一问题的原因是图形界面元素的生成很不方便。为此 Windows 提供了应用程序设计接口 API(Application Programming Interface)，它包含了 600 多个函数，极大地方便了图形用户界面的开发。但是在这批函数中，大量的函数参数和使用数量更多的有关常量，使基于 Windows API 的开发变得相当困难。为此 Borland C++推出了 Object Windows 编程。它将 API 的各部分用对象类进行封装，提供了大量预定义的类，并为这些类定义了许多成员函数。利用子类对父类的继承性，以及实例对类的函数的引用，应用程序的开发可以省却大量类的定义，省却大量成员函数的定义或只需做少量修改以定义子类。

Object Windows 还提供了许多标准的缺省处理，大大减少了应用程序开发的工作量。但要掌握它们，对非专业人员来说仍是一个沉重的负担。为此人们利用 Windows API 或 Borland C++的 Object Windows 开发了一批可视开发工具。可视化开发就是在可视开发工具提供的图形用户界面上，通过操作界面元素，诸如菜单、按钮、对话框、编辑框、单选框、复选框、列表框和滚动条等，由可视开发工具自动生成应用软件。这类应用软件的工作方式是事件驱动。对每一事件，由系统产生相应的消息，再传递给相应的消息响应函数。这些消息响应函数是由可视开发工具在生成软件时自动装入的。

传统的程序设计方法，不论是需求分析，还是系统设计，都是针对数学模型的，出发点是“怎样做(How)？”，即用计算机求解这个实际问题应该“怎样做？”。从实际问题到求解模型(数学模型)的抽象都是围绕“怎样做？”去进行的。因此它存在着生产率低下、软件重用程度低、难维护、不能满足用户需求等问题。

为了克服传统方法的缺点，人们不断地在创造新的方法。除了上面提到的面向对象的方法与技术，演化成一种完整的软件开发方法和系统的技术体系，即软件工程的第二代——对象工程，还有之后的软件工程的第三代——软件过程工程。人们在研究和实践中发现，为了提高软件生产率，并使软件质量得到保证，其关键在于软件开发和维护中的管理和支持能力，并认识到最关键的是“软件过程”。

进入20世纪90年代之后，软件工程的一个重要进展就是给予构建的开发方法。为了提高软件生产力，不草率地开发应用程序，则要尽可能地利用可复用构件，组成新的应用软件系统，随着Internet技术的飞速发展，大量的分布式处理系统需要开发，这种方法的重要性也日益显露出来从而成为软件工程的第四代，也称为构件工程。

软件开发是一项复杂的工作，针对软件开发的管理和控制，发展出一门专门的学科：软件工程。而软件工程现在还在不断的发展，因此我们也需要在实践中不断地学习和创新发展。

## 本章小结

本章主要是对传统软件工程和现代软件工程进行对比介绍。除了简要地回顾传统软件工程的定义、基本目标和模式外，对现代软件工程模式也做出了说明，并对现代软件工程中提出的新的概念加以解释，比如软构件、软件复用以及面向对象的软件开发方法和可视化开发方法。软件工程的发展很快，因此我们需要不断学习新的方法和技术。

## 思考题

1.1 软件工程研究的主要内容是什么？

1.2 现代软件工程和传统软件工程的区别和联系？

1.3 软件工程的七条基本原理具有什么现实意义？

1.4 现代软件工程种所采用的面向对象的软件开发方法与传统方法的区别？

1.5 什么是软件复用技术？

1.6 软件生存周期模型有哪几种，各有什么特点？

# 第2章　现代软件需求

## 2.1　软件需求概述

软件需求是软件工程中一个极为重要的阶段。需求分析做得好不好直接关系到以后整个软件质量的高低。一个不完善的需求分析，即使有再强大的软件开发队伍和熟练的开发技术，也难以开发出一个高质量的软件。因此做好软件需求分析是软件开发的一个极其关键的环节。

### 2.1.1　软件需求的定义

软件开发中主要包含需求、设计、编码和测试四个阶段，其中需求工程是软件工程第一个也是很重要的一个阶段。那么什么是软件需求？软件需求包含着多个层次，不同层次的需求从不同角度与不同程度反映着细节问题。

IEEE 软件工程标准词汇表(1997 年)中定义需求为系统或系统部件要满足合同、标准、规范或其他正式规定文档所需具有的条件或动能。IEEE 公布的定义是从用户角度(系统的外部行为)，以及从开发者角度(一些内部特性)来阐述需求的。

另一种定义认为需求是“用户所需要的并能触发一个程序或系统开发工作的说明”。需求分析专家 Alan Davis(1993)拓展了这个概念：“从系统外部能发现系统所具有的满足于用户的特点、功能及属性等”。

为达到这一目的，最简单的方法就是把系统看成一个黑盒，然后考虑为了完整地描述这个黑盒能做些什么，你所需要做的事情。除了输入输出以外，还需要考虑系统的一些其他特性，包括它的性能和其他类型的复杂行为，以及系统与它的环境交互的其他方式。

采用类似的方法，我们需要做四类主要事情来完全描述系统：

（1）系统的输入——不仅是输入的内容，还有输入的设备、形式、外观以及感觉等必要的细节。许多开发人员都已经注意到，这一领域可能包括大量的细节，并且具有易失性，尤其是 GUI、多媒体或互联网；

（2）系统的输出——必须支持对输入设备的描述，如语音输出或可视化显示等，以及系统所产生的协议和格式；

（3）系统的属性——非行为需求，如可靠性、可维护性、可得到性以及吞吐量等开发人员必须考虑的因素；

（4）系统环境的属性——附加的非行为需求，如系统在不同操作约束、负载和操作系统兼容性中运行的能力。