软件工程系列规划教材

软件设计与体系结构

周　华　　主编
孙兴平　胡　盛　李　浩　编著
康洪炜　刘俊晖

software design and architecture

科学出版社

软件工程系列规划教材

软件设计与体系结构

周　华　　主编

孙兴平　胡　盛　李　浩
康洪炜　刘俊晖　　编著

科学出版社

北　京

内 容 简 介

本书从CDIO工程理念出发，围绕构思(Conceive)、设计(Design)、实现(Implement)、运作(Operate)四个方面展开阐述，以一种崭新的视角展现软件设计和体系结构的内容，尽可能做到覆盖面广和内容新颖，在保持经典内容的基础上，力求提供来自业界最新的内容和进展。

构思(C)给出一个软件要做什么的总体思路；设计(D)给出一个软件怎么做的方法和手段；实现(I)给出一个软件实际做出来的实现技术和路线；运作(O)给出一个软件如何成功运作的模式和方法。CDIO各个部分相对独立但又互有联系，能够让读者以全面的、主动的和实用的方式学习和掌握相关内容，并着重强调工程实践训练和综合能力培养。

本书可作为计算机相关专业本科生和研究生的教材，同时也是软件工程领域专业人员的优秀参考读物。

图书在版编目(CIP)数据

软件设计与体系结构/周华主编. —北京：科学出版社，2012
软件工程系列规划教材
ISBN 978-7-03-034429-8

Ⅰ. ①软… Ⅱ. ①周… Ⅲ. ①软件设计—教材 Ⅳ. ①TP311.5

中国版本图书馆CIP数据核字(2012)第104566号

责任编辑：贾瑞娜 / 责任校对：刘小梅
责任印制：阎 磊 / 封面设计：迷底书装

科学出版社 出版
北京东黄城根北街16号
邮政编码：100717
http://www.sciencep.com

化学工业出版社印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2012年5月第 一 版 开本：787×1092 1/16
2012年5月第一次印刷 印张：27 1/4
字数：696 000

定价：52.00元

(如有印装质量问题，我社负责调换)

“软件工程系列规划教材”专家委员会

前　言

软件已经成为基于计算机的系统及产品的关键组成成分，并且成为世界舞台上最重要的技术之一。软件现在已从解决问题和信息分析工具发展成为一门独立的产业，覆盖了科技和应用的很多领域，被认为是信息化的灵魂。然而，如何在有限的时间内，利用有限的资源开发高质量的软件仍然是我们面临的难题。

CDIO 代表构思(Conceive)、设计(Design)、实现(Implement)和运作(Operate)，是近年来国际工程教育改革的最新成果。CDIO 改革包括 CDIO 理念和与之相适应的学习目标(教学大纲)、实现与评估标准，以及一系列的规划、设计、实施，评估理论和实践资源。

CDIO 改革的愿景是为学生提供一种在实际系统和产品的构思—设计—实现—运作的背景环境下强调工程基础的工程教育，使学生能够：

- 掌握深厚的技术基础知识；
- 领导新产品和新系统的开发与运行；
- 理解工程技术的研究与发展对社会的重要性和战略影响。

本书力图从 CDIO 工程理念出发，以一种崭新的视角展现软件设计和体系结构的内容，并力求做到覆盖面广和内容新颖，既照顾到经典的方法与技术，又不缺乏时代的特征和进展。

C 代表构思(Conceive)，给出一个软件要做什么的总体思路。因而要对软件和软件环境有一个全面的了解，从程序设计语言、开发工具和各种平台环境到软件工程中的软件过程模型、软件工程经济学、软件过程管理和软件维护及演化等，并通过需求建模对软件要做什么进行描述，同时关注体系结构则有利于控制软件复杂性和提高系统质量。

D 代表设计(Design)，给出一个软件怎么做的方法和手段，它是软件需求的直接体现，为软件实现提供直接依据。重点探讨软件与体系结构设计的各个层面，通过软件设计建模对软件要怎么做进行构建，结合模式思想与常用模式，体系结构风格，模式与设计方法，软件复用思想与方法，以及软件界面设计和评估方法等探讨软件设计和体系结构深层次的问题。

I 代表实现(Implement)，给出一个软件实际做出来的实现技术和路线，是软件设计的直接实现，是软件体系结构的具体应用。涉及 Web 开发、数据库开发、软件成本估算、软件架构、软件集成和软件测试等重要技术，为开发出高质量的软件提供了原则性的指导，为技术的综合运用指明了实现的路径。

O 代表运作(Operate)，给出一个软件如何成功运作的模式和方法，结合软件架构师的角色和职责，对软件运作的模式进行了深入的讨论，多角度阐述了软件企业的商务模式，全方位探讨了软件产品的营销策略，并通过实际案例阐明了软件运作的基本规律和方法。

本书的组织分为构思篇(Conceive)、设计篇(Design)、实现篇(Implement)和运作篇(Operate)四个部分，各部分的内容按照如上理念完成。在学习过程中，即可以遵循 CDIO 的理念全部地或有重点地学习，也可以有针对性地学习某部分的内容。每一章之后都给出了习题和思考题，用于加深对相关原理、方法或技术的熟悉程度，并引导思考一些深层次的问题。

本书的完成得益于前人所做的工作，成书过程中借鉴了已有著作和论文的内容，在此对列入参考文献的作者表示感谢。本书的部分内容也从互联网上吸取了众多精华，对这些难于列入参考文献的作者也表示感谢。

本书由周华主编，并撰写了第1、9、10、15、16章，孙兴平撰写了第3、4、5章，胡盛撰写了第6、7、11章，李浩撰写了第17章，康洪炜撰写了第2、13章，刘俊晖撰写了第8、14章，其中第12章的内容由孙兴平、李浩、康洪炜、刘俊晖共同撰写。梁志宏老师认真审阅了本书的全部初稿，并提出了许多中肯的修改意见。廖赟、陈清毅、张引、姜迪、杨雨、何臻力、段清、沈勇、杨胜林也为本书付出了辛勤的劳动和智慧。借此机会，作者谨向为本书做出贡献的所有人士表示诚挚的谢意。

最后，特别感谢科学出版社给予本书的支持，感谢各位编辑为本书的策划和出版付出的心血。

限于编者的水平，错误与不妥之处难免存在，衷心希望广大读者指正赐教。

作　者

2011年11月

目　录

第一篇　构思篇(Conceive)

第二篇　设计篇(Design)

第三篇 实现篇(Implement)

第四篇 运作篇(Operate)

第一篇　构思篇(Conceive)

“构建一个软件系统最困难的部分是确定构建什么。其他部分的工作不会像这部分工作一样，在出错之后会如此严重地影响随后实现的系统，并且在以后修补竟会如此的困难。”

——Fred Brooks

第1章 软件环境

1.1 软件与软件分类

软件(Software)是用户与硬件之间的接口界面,用户主要是通过软件与计算机进行交流。软件是计算机系统设计的重要依据。为了方便用户,为了使计算机系统具有较高的总体效用,在设计计算机系统时,必须全局考虑软件与硬件的结合,以及用户的要求和软件的要求。来自众多教材的关于软件的定义一般包含如下三方面内容:

(1) 指令的集合(也称为计算机程序),通过执行这些指令可以满足预期的特征、功能和性能需求。

(2) 数据结构,它使程序可以充分利用信息。

(3) 描述程序功能需求以及程序如何操作和使用的文档。

因而可以认为

软件=程序+数据+文档

中华人民共和国国家标准软件工程术语 GB/T11457—1995 对软件的定义是:与计算机系统的操作有关的程序、规程、规则及任何与之有关的文档。对于计算机来说,硬件就如同是一个人的身躯,软件就如同一个人的思想和灵魂。

1.1.1 软件的特点

软件作为一个产品或服务,与其他产品相比具有很大的区别,具体表现如下特点:

(1) 软件是一种逻辑实体,不是具体的物理实体,它具有抽象性。软件的好坏与否,正确与否,必须通过测试、分析、思考、判断之后才知道。这一特性就给软件的设计、生产和管理带来了很多困难。

(2) 软件产品的生产主要是研制,是知识的结晶,不像传统意义上的硬件制造有明显的制造过程,在软件的开发过程中没有明显的制造过程。

(3) 软件具有"复杂性",其开发和运行常受到计算机系统的限制,对计算机系统有着不同程度的依赖性。为了消除这种依赖性,引入了软件移植,并把可移植性当做衡量软件质量的因素之一。

(4) 软件的生产职能由人采用人工方式来生产,还未完全摆脱手工开发方式,成本昂贵。所以为了加快软件技术的发展,就必须提出和采用新的软件开发方法。例如,可以采用软件复用技术或者使用有效的软件开发环境提高软件开发效率。

(5) 在软件的运行和使用期间,没有硬件那样的机械磨损和老化问题,但存在退化问题,由于修改及需求的变化导致退化。

硬件故障率曲线是一个 U 形曲线(即为浴盆曲线,在产品整个使用寿命期间,典型的故障率变化曲线,形似浴盆),说明硬件随着使用时间的增加,故障率也急剧上升(图 1.1)。

软件故障率曲线,没有 U 形曲线的右半部分说明软件不会随着使用时间的增加使得故障率增加,所以软件不存在磨损和老化问题,但是存在退化问题(图 1.2)。

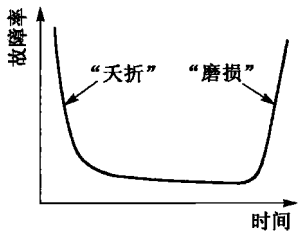

图 1.1 硬件故障率曲线[1]

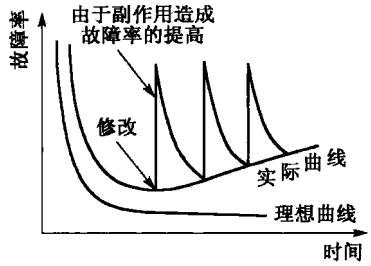

图 1.2 软件故障率曲线

1.1.2 软件的分类

不同类型的工程对象，对其进行开发和维护有着不同的要求和处理方法，目前还找不到一个统一的严格分类标准。一般来说，可以将软件分为七大类：

(1) 系统软件。系统软件是管理、使用和维护计算机系统资源的软件。它使得用户和其他软件将计算机当做一个整体而不需要顾及底层每个硬件是如何工作的。一般来讲，系统软件包括操作系统和一系列基本的工具(如编译器、数据库管理、存储器格式化、文件系统管理、用户身份验证、驱动管理、网络连接等方面的工具)。

(2) 应用软件。应用软件解决属于专用领域的，非计算机本身问题的软件。它是在特定领域内开发，针对用户的某种应用目的所撰写的软件。如 Microsoft 公司的办公软件 Microsoft Office，Adobe 公司的图像处理软件 Photoshop，腾讯公司的通信工具 QQ，ACDSystems 公司的图像浏览软件 ACDSee 等。

(3) 工程/科学软件。工程科学软件的应用范围非常地广泛，涵盖了航空航天学、天文学、地理学、生物学、物理学等。随着科学工程的复杂化，工程/科学软件不再仅仅局限于传统的数值计算，目前的工程/科学计算软件开始带有实时软件和系统软件的特性。计算机辅助设计软件和系统仿真软件均属于工程科学软件的范畴。

(4) 嵌入式软件。嵌入式软件就是嵌入在硬件中的操作系统和开发工具软件，由于存储空间有限，因而要求软件代码紧凑、可靠，大多对实时性有严格要求。嵌入式软件产业发展迅猛，已成为软件体系的重要组成部分。嵌入式软件具有非常广阔的应用前景，应用领域包括：工业控制、交通管理、信息家电、家用电器、医疗设备、电子商务、环境工程与自然、机器人等。

(5) 产品线软件。产品线软件面向多个不同的用户提供一系列符合用户实际应用的功能，可以分为面向有限特定市场的产品线软件，如库存控制软件；面向大众消费品的产品线软件，如财务应用软件、多媒体软件、游戏软件等。

(6) Web 应用程序。随着 B2B 应用和电子商务的发展，Web 应用不仅仅只是一组超文本链接文件，它正朝着复杂化的趋势发展，它可以为用户提供强大的计算功能甚至于还可以与商务应用程序及企业数据库相连接。随着互联网技术的发展和应用软件的成熟，在 21 世纪 SaaS(Software-as-a-service，软件即服务)作为一种完全创新的软件应用模式逐步兴起。软件的厂商将应用软件统一部署在自己的服务器上，通过 Internet 向用户提供所需的应用软件服务，客户可以根据自己的实际需求向厂商订购所需的软件服务，类似于用户不再购买软件而是向厂商租用基于 Web 的软件，并且按照订购服务的数量和事件来向厂商支付费用，节省了购买、构件、维护基础设施和应用程序的成本，它与 on-demand software(按需软件)，the application service provider(ASP，应用服务提供商)，hosted software(托管软件)具有相似的含义。国内著名的 SaaS 厂商如提供在线会计的金蝶友商网，提供在线 CRM(Customer Relationship Management，客户关系管理)的八百客等。

(7) 人工智能软件。人工智能是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。人工智能软件能以人类智能相似的思维方式解决计算和分析问题。其开发涉及信息论、控制论、自动化、仿生学、生物学、心理学、数理逻辑、语言学、医学和哲学等多门学科。人工智能领域的应用程序包括机器人、语言识别、图像识别、自然语言处理和专家系统等。

除了上述的七种分类，业内还可以将软件按照授权方式和软件规模进行分类。

(1) 按照授权方式的分类。不同的软件一般都有对应的软件授权，软件的用户必须在同意所使用软件的许可证的情况下才能够合法地使用软件，按照软件的授权方式来分类，软件可以分为商业软件、开源软件、共享软件、免费软件、公共软件。

• 商业软件(commercial software)是不允许用户随意地复制、研究、修改或者散布的软件，一般被作为商品进行交易。违反此类软件授权的行为通常要承担严重的法律责任。传统的商业软件公司多数会采用此类授权。

• 开源软件(open-source software)是源代码可以被公众使用的软件，并且对此软件的使用、修改和分发也不受许可证的限制或仅给予些许其他的限制。著名的开源软件如 Linux，MySQL，FireFox，Apache 等。

• 共享软件(shareware)通常可以从各种渠道免费下载并使用其试用版，但是在使用功能或者使用时间上受到限制。小到一款 CD 到 MP3 的转换软件，大到 Microsoft 的 Office 和 Vista 操作系统，都可以先下载来试用一番，觉得不错再出钱去购买。

• 免费软件(freeware)是无需付费就可以取得的软件，而且使用上也不会出现任何日期的限制或者是软件使用上的限制，但是通常会有其他的限制，如使用者不能擅自修改软件的源代码，不得将其用于商业用途等，否则视同侵权。

• 公共软件(public domain software)是原作者已放弃权利或者著作权过期或者作者已经不可考究的软件。软件的使用上无任何限制。

(2) 按照软件规模分类(表 1.1)。

表 1.1 软件按规模分类

序号	类别名称	产品规模(源程序行数)/行	参加人数/人	研制周期
1	微型软件	0.5KB	1	1 周～1 月
2	小型软件	1～2KB	1	1～6 月
3	中型软件	5～10KB	2～5	1～2 年
4	较大型软件	50～100KB	8～20	2～3 年
5	大型软件	1MB	100～1000	4～5 年
6	极大型软件	1～10MB	1000～5000	6～10 年

1.2 程序设计语言与开发环境

1.2.1 程序设计语言

程序设计语言(Programming Design Language，PDL)，又称编程语言(Programming Language)，是一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧，用来向计算机发出指令。

程序设计语言可分为三大类：机器语言、汇编语言和高级语言。计算机本身只能执行机器语

言程序，因此任何一种语言程序最终都必须翻译成相应的机器语言程序，完成这种翻译工作的程序称为翻译程序或编译程序[2]。

机器语言实际上是一串二进制代码，因此用这种语言去进行程序设计，既烦琐费时又容易出错。因此，后来人们就研究出所谓的汇编语言（也称为符号语言）。汇编语言仍然保持着机器语言的基本特性，只不过把操作码和地址码用符号表示而已。

虽然汇编语言比机器语言更为简单，但用起来还是略显烦琐，此外它没有通用性，因此后来人们又设计出各种高级语言。高级语言不再过度地依赖某种特定的机器或环境，这是因为高级语言在不同的平台上会被编译成不同的机器语言，而不是直接被机器执行。接下来将介绍几种常用的程序设计语言。

（1）BASIC（Beginner's All-purpose Symbolic Instruction Code，又译培基），意思就是"初学者的全方位符号式指令代码"，是一种设计给初学者使用的程序设计语言。BASIC 是一种直译式的编程语言，在完成编写后不须经由编译及链接等手续即可运行，但如果需要单独运行时仍然需要将其创建成可执行文件。

（2）C 语言是一种通用的、过程式的编程语言，广泛用于系统与应用软件的开发。具有高效、灵活、功能丰富、表达力强和较高的移植性等特点，在程序员中备受青睐。

（3）C++是一种使用非常广泛的电脑程序设计语言。它是一种静态数据类型检查的，支持多范型的通用程序设计语言。C++支持过程化程序设计、数据抽象化、面向对象程序设计、泛型程序设计、基于原则设计等多种程序设计风格。

（4）C#是 Microsoft 推出的一种基于. NET 框架的、面向对象的高级编程语言。C# 由 C 语言和 C++派生而来，继承了其强大的性能，同时又以. NET 框架类库作为基础，拥有类似 Visual Basic 的快速开发能力。

（5）Fortran，亦译为福传，是由 Formula Translation 两个字所组合而成，意思是"公式翻译"。它是世界上第一个被正式采用并流传至今的高级编程语言。

（6）Java 是一种面向对象的程序设计语言，拥有跨平台、面向对象、泛型编程的特性。任职于 Sun 公司的詹姆斯·高斯林（James Gosling）等于 20 世纪 90 年代初开发出 Java 语言的雏形，最初被命名为 Oak，目标是设置在家用电器等小型系统的程序语言，应用于电视机、电话、闹钟、烤面包机等家用电器的控制和通信。由于这些智能化家电的市场需求没有预期的高，Sun 公司放弃了该项计划。随着 90 年代互联网的发展，Sun 公司看见 Oak 在互联网上应用的前景，于是改造了 Oak，于 1995 年 5 月以 Java 的名称正式发布。Java 伴随着互联网的迅猛发展而发展，逐渐成为重要的网络编程语言。后来公司被甲骨文公司并购，Java 也随之成为甲骨文公司的产品。

（7）Pascal 是一个有影响的面向对象和面向过程的编程语言，由尼古拉斯·沃斯在 1968 年 9 月设计，在 1970 年发行，作为一个小型的和高效的语言，意图鼓励使用结构化编程和数据结构进行良好的编程实践。

（8）Haskell 是一种纯函数式编程语言，它的命名源自美国数学家 Haskell Brooks Curry，他在数学逻辑方面上的工作使得函数式编程语言有了广泛的基础。Haskell 语言是 1990 年在编程语言 Miranda 的基础上标准化的，并且以 Lambda-Calculi（兰姆达演算）为基础发展而来。这也是为什么 Haskell 语言以希腊字母 Lambda 作为自己的标志。Haskell 语言的最重要的两个应用是 Glasgow Haskell Compiler（GHC）和 Hugs（一个 Haskell 语言的编译器），特色是利用很简单的叙述就可以完成 Linked List、矩阵等数据结构。

1.2.2 开发环境

开发环境，也称为集成开发环境（Integrated Development Environment，IDE，也有人称为

Integration Design Environment、Integration Debugging Environment)是一种辅助程序开发人员开发软件的应用软件。

IDE 通常包括编程语言编辑器、编译器/解释器、自动建立工具、通常还包括调试器。有时还会包含版本控制系统和一些可以设计图形用户界面的工具。许多支持面向对象的现代化 IDE 还包括了类别浏览器、物件检视器、物件结构图。虽然目前有一些 IDE 支持多种编程语言(如 Eclipse、NetBeans、Microsoft Visual Studio),但是一般而言,IDE 主要还是针对特定的编程语言而量身打造(如 Visual Basic)。接下来将介绍几个常用的开发环境。

(1) Delphi,是 Windows 平台下著名的快速应用程序开发工具(Rapid Application Development,RAD)。Delphi 使用的核心是由传统 Pascal 语言发展而来的 Object Pascal,以图形用户界面(Graphical User Interface,GUI)为开发环境,透过 IDE、VCL 工具与编译器,配合链接数据库的功能,构成一个以面向对象程序设计为中心的应用程序开发工具。Delphi 所编译的可运行档,虽然容量较大,但因为产生的是真正的原生机器码,效能上比较快速。除了使用数据库的程序之外,不需安装即可运行,在使用上相当方便。

(2) Eclipse 是著名的跨平台自由集成开发环境(IDE)。最初主要用于 Java 语言开发,目前亦有人通过插件使其作为 C++、Python、PHP 等其他语言的开发工具。Eclipse 的本身只是一个框架平台,但是众多插件的支持,使得 Eclipse 拥有较佳的灵活性。许多软件开发商以 Eclipse 为框架开发自己的 IDE。

(3) Microsoft Visual Studio(简称 VS)是美国 Microsoft 公司的开发工具包系列产品,可以用来创建 Windows 平台下的 Windows 应用程序和网络应用程序,也可以用来创建网络服务、智能设备应用程序和 Office 插件。VS 是一个基本完整的开发工具集,它包括了整个软件生命周期中所需要的大部分工具,如 UML 工具、代码管控工具、集成开发环境等。所写的目标代码适用于 Microsoft 支持的所有平台,包括 Microsoft Windows、Windows Mobile、Windows CE、.NET Framework、.NET Compact Framework 和 Microsoft Silverlight。Visual Studio 包含基于组件的开发工具(如 Visual C#、Visual J#、Visual Basic 和 Visual C++),以及许多用于简化基于小组的解决方案的设计、开发和部署的其他技术。

1.3 数据库环境

计算机自诞生以来就被用于进行事务处理,并存储大量的信息。计算机对信息的处理分为以下几个阶段:人工管理、文件管理、数据库管理。数据库的出现为数据处理带来了革命性的变化。数据库是由一些互相关联的文件以及一组使得用户可以访问和修改这些文件的程序构成的集合,其基本目标是提供一个可以方便有效地存储、访问和维护数据信息的环境[3]。相对于文件管理来说,数据库管理具有以下优势:数据库使得分散孤立的数据变得通用,使数据成为具有逻辑关系的数据集合,用户可以通过统一的访问接口和方法对数据进行操作和访问,并且具有高并发性和低冗余性。正因为这些特点,自从数据库出现以来数据库技术获得了长足的进步。数据库系统从 20 世纪 60 年代末开始发展至今大致经历了三代:第一代,层次数据库、网状数据库;第二代,关系型数据库;第三代,以面向对象技术为主要特征的面向对象数据库。本节我们将详细介绍关系型数据库、面向对象数据库和非结构化数据库三种类型的数据库。

1.3.1 关系数据库

1. 层次型数据库

层次性数据库是以层次模型组织的数据库,是最早出现和应用的数据库。1968 年,IBM 公