



软件工程

第4-1章 软件设计-概要设计

重庆理工大学

计算机科学与工程学院 李梁

qq:1255214405 课程号: K3388095


liliang@cqut.edu.cn

智慧树课号: K3388095

智慧树网址: www.zhihuishu.com




17:43:52 重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)



第4-1章 软件设计-概要设计

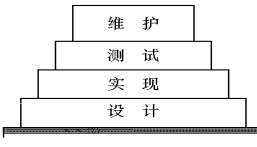
- 1 软件设计概述
 - 1.1 软件设计及设计过程
 - 1.2 软件设计的主要任务
 - 1.3 软件设计的目标
 - 1.4 软件设计的文档
- 2 软件设计原理与优化准则: 模块化设计
 - 2.1 软件结构与软件设计过程
 - 2.2 模块化
 - 2.3 模块化的独立性
 - 2.4 软件设计的优化准则
- 2.5 软件结构图
- 3 结构化设计方法: 面向数据流设计方法 (SD法)
- 4 软件体系结构模型 (自学)
- 5 详细设计

17:43:52 重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)

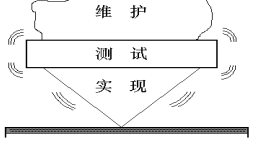


1.1 软件设计概述-软件设计及设计过程

- “设计先于编码”，这是软件工程“推迟实现”基本原则
- 软件系统设计是把软件需求“变换”为用于构造软件的蓝图。
 - ✓ “输入”是需求分析各种模型元素
 - ✓ “输出”是软件设计模型和表示
- 对于软件设计，设计者不可能一次就完成一个完整的软件设计，所以，软件设计是一系列迭代步骤的过程。
- 软件设计是后续开发步骤及软件维护工作的基础。如果没有设计，只能建立一个不稳定的系统结构



有软件设计



没有软件设计

17:43:52




1.1 软件设计概述-软件设计及设计过程

- 软件设计目标是构造系统“怎么做”的模型描述，是对将要实现的软件系统的体系结构、系统的数据、系统模块间的接口，以及所采用的算法给出详尽的描述。




17:43:52 重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)



1.1 软件设计概述-软件设计及设计过程

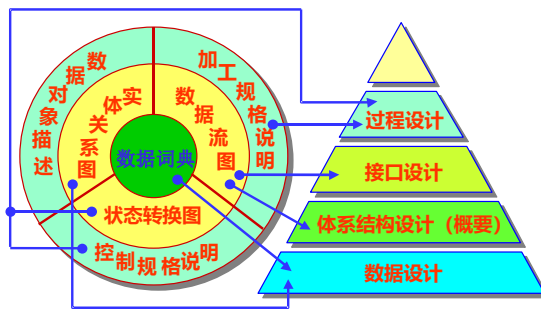
- 结构设计，或称概要设计，或软件总体设计，或高层设计：定义软件组成及各主要成分之间的关系，构造软件系统的整体框。
- ✓ 分析需求规格说明、模块划分，形成具有预定功能的模块组成结构，表示出模块间的控制关系。
- 数据设计：将实体-关系图中描述的对象和关系，以及数据词典中描述的详细数据内容转化为数据结构。
- 接口设计：接口设计根据数据流程图定义软件内部各成份之间、软件与其它协同系统之间及软件与用户之间的交互机制。
- 过程设计，也称模块设计、详细设计，或低层设计：对系统框架、数据结构和界面表示进行细化，对各结构成分所实现的功能，用很接近程序的软件表示形式进行过程性描述。
 - ✓ 设计模块细节，确定模块所需的算法和数据结构等。
- 设计测试和复审

17:43:52 重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)



1.1 软件设计概述-软件设计及设计过程

- 软件设计--将分析模型转换为设计模型



17:43:52 重庆理工大学计算机科学与工程学院 李梁(liliang@cqut.edu.cn)

1.1 软件设计概述-软件设计及设计过程

从管理与技术观点看软件设计过程

①概要（结构、总体）设计：将软件需求转化为软件的系统结构，形成概要设计规格书。

②详细（过程）设计：过程设计和界面设计。形成详细设计规格书。

③数据设计：包括数据结构、文件和数据库的设计。

④结构设计：选择合理的体系结构，对系统进行分解和划分。

⑤过程设计：通过对结构表示进行细化，得到软件详细的界面、数据结构和程序算法。

重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn)

1.2 软件设计概述-软件设计的主要任务

设计前进行的系统设计：设计供选择的方案（开发平台、网络构架、部署平台），选取合理的方案或推荐最佳方案

软件设计的八项任务：具体细节过程如果有或存在？

- (1) 制定规范
- (2) 结构设计
- (3) 处理方式设计
- (4) 数据结构及数据库设计
- (5) 可靠性设计（质量设计）
- (6) 过程设计与详细设计
- (7) 编写软件设计文档
- (8) 设计审查和复审

重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn)

1.2 软件设计概述-软件设计的主要任务

(1) 制定规范：制定设计时共同遵守的标准，以便协调组内各成员的工作。

- ✓ 确定设计目标和最合适的设计方法。
- ✓ 规定编码的信息形式和接口规约。
- ✓ 确定命名规范。
- ✓ 确定构件重用策略。
- ✓ 面向模块（部件或构件）进行设计。
- ✓ 规定设计文档的规范。
- ✓ 规定代码编写规范。
- ✓ 概要设计方法；
- ✓ 详细设计和编程方法。

重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn)

1.2 软件设计概述-软件设计的主要任务

(2) 结构设计：基于功能的层次结构，确定软件的体系结构。

- ① 将系统按功能划分成模块的层次结构
- ② 确定每个模块的功能、与需求相对应
- ③ 确定模块间的调用关系
- ④ 确定模块间的接口
- ⑤ 评估模块划分的质量

(3) 处理方式设计：

- ① 确定算法，评估算法的性能
- ② 确定模块间的控制方式
 - ✓ 周转时间
 - ✓ 响应时间
 - ✓ 吞吐量
 - ✓ 精度
- ③ 确定外部信号的接收发送形式

重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn)

1.2 软件设计概述-软件设计的主要任务

(4) 数据设计

- 数据结构的设计：保证数据完整性和安全性
 - ✓ 确定I/O文件的数据结构
 - ✓ 确定算法所必需的逻辑数据结构
 - ✓ 确定对逻辑数据结构所操作的程序模块
 - ✓ 限制和确定各个数据设计的影响范围
 - ✓ 确定接口数据其详细的数据结构和使用规则
- 数据库设计：确定数据库的模式、子模式
 - ✓ 概念结构设计：可以用E-R模型或用3NF关系群表示
 - ✓ 逻辑结构设计：是将概念模型转换成DBMS支持的数据模型。
 - ✓ 物理结构设计：设计数据模式的一些物理细节，选定存储结构和存取方法，以获得数据库的最佳存取效率。
- 数据的保护性设计：
 - ✓ 防卫性设计：设计插入自动检错，报错和纠错的功能
 - ✓ 一致性设计：并发处理封锁和解除机制保持数据不被破坏
 - ✓ 冗余性设计：

重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn)

1.2 软件设计概述-软件设计的主要任务

(5) 可靠性设计（质量设计）：确定软件可靠性和其它质量指标，确保设计具有良好的质量特性，使软件易于修改和易于维护。

(6) 过程设计与详细设计（模块设计、过程设计、低层设计）：将概要设计的框架内容具体化，明细化，将概要设计模型转化为可以操作的软件模型。详细设计不是真正的编码，是设计出程序的详细规格说明，即软件蓝图，包含必要的细节，程序员可以根据它们写出实际的程序代码。

- ✓ 详细设计是结构设计中的各个模块设计过程细节，确定模块所需的算法和数据结构等。
- ✓ 详细设计主要描述每个模块或构件的设计细节，主要包括模块或构件的处理逻辑、算法、接口等。
- ✓ 详细设计决定了最终程序代码的质量。
- ✓ 详细设计说明书又称为程序设计说明书，如果一个软件系统比较简单，层次很少，文档可以不单独编写，有关内容合并入概要设计说明书。

重庆理工大学计算机科学与工程学院 李黎(liliang@cqut.edu.cn)

1.2 软件设计概述-软件设计的主要任务

(7)编写软件设计文档

- ①概要设计规格说明书、数据库设计规格说明书、详细设计规格说明书
- ②用户使用手册、测试计划与测试标准

(8) 设计审查和复审

- ①可追溯性：需求复盖确认
- ②接口：内部接口与外部接口定义的确认。
- ③风险：确认技术条件、预算范围。
- ④实用性：确认对于需求的实用性。
- ⑤技术清晰度：确认代码的可实现性
- ⑥可维护性：确认可维护性
- ⑦质量：确认质量特征
- ⑧各种选择方案：选择方案的标准
- ⑨限制：评估限制的现实性，与需求的一致性
- ⑩其它具体问题：对文档、可测试性、设计过程等进行评估

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

13

1.3 软件设计概述-软件设计的主要目标

软件设计的最终目标：**取得最佳方案**

- ①节省开发费用、
- ②降低资源消耗、
- ③缩短开发时间、
- ④能够赢得较高的生产效率、
- ⑤较高的可靠性、
- ⑥可维护性的方案。

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

14

1.4 软件设计概述-小结

软件设计在开发阶段中的重要性：软件设计是开发阶段中最重要的步骤，它是软件开发过程中质量得以保证的关键步骤。

- 数据设计将实体一关系图中描述的对象和关系，以及数据词典中描述的详细数据内容转化为数据结构的定义。
- 体系结构设计定义软件系统各主要成份之间的关系。
- 接口设计根据数据流图定义软件内部各成份之间、软件与其它协同系统之间及软件与用户之间的交互机制。
- 过程设计则是把结构成份转换成软件的过程性描述。

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

15

2.1 软件设计原理与优化准则-软件结构与过程

软件设计过程的目标

- 设计必须实现分析模型中描述的所有显式需求，必须满足用户希望的所有隐式需求。
- 设计必须是可读、可理解的，易于编程、测试及维护。
- 设计应从实现角度出发，给出与数据、功能、行为相关的软件全貌。

衡量设计的技术标准

- ①结构应是**分层结构**，从而建立软件成份之间的**控制**。
- ②应当模块化，从逻辑上将软件划分为完成特定功能或子功能的构件。
- ③设计应当既包含**数据抽象**，也包含**过程抽象**。
- ④设计应当建立具有具有独立功能特征的模块。
- ⑤应当建立能够降低模块与外部环境之间复杂连接的接口。
- ⑥设计应根据软件需求分析获取的信息，建立可驱动可重复的方法。

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

16

2.1 软件设计原理与优化准则-软件结构与过程

软件结构的划分过程：该划分过程从需求分析确立的目标系统的模型出发，对**整个问题进行分割**，使每个部分用一个或几个软件成份加以解决。

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

17

2.1 软件设计原理与优化准则-软件结构与过程

模块的划分：如果模块是相互独立的，模块越小，每个模块花费的工作量越低；模块数增加时，模块间的联系也随之增加，把这些模块联接起来的工作量也随之增加。因此，要设计成M个模块，使得总的开发成本达到最小。

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

18

2.1 软件设计原理与优化准则-软件结构与过程

树状结构的特点：整个结构只有一个顶层模块，而对于任何一个下层模块，只有一个上级模块，而且同一层模块之间不发生联系。

网状结构的特点：任意两个模块间都可以有双向的关系。由于模块间不存在上下的关系，任何两个模块都是平等的，因此分不出层次来。

(a) 树状结构 (b) 网状结构 (c) 网状结构

重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn) 19

2.1 软件设计原理与优化准则-软件结构与过程

深度 宽度 控制层次 扇出 扇入

重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn) 20

2.1 软件设计原理与优化准则-软件结构

● 软件体系结构是指软件的整体结构，可以是程序构件或模块的结构或组织，这些构件或模块交互的形式以及这些构件所用数据的结构。

门户网站 业务系统 协同工作系统 教育财务综合数据中心 通用基础系统

17:43:52 21

2.1 软件设计原理与优化准则-软件结构

重庆市教育财务一体化系统 门户网站 统一身份认证 协同工作 个人协同中心 公文处理 经费执行 审计整改 综合 经费执行 审计整改 综合 经费执行 审计整改 综合

17:43:52

2.1 软件设计原理与优化准则-软件结构

某政府采购网络结构

信息门户 公众门户 采购人门户 供应商门户 代理机构门户 监管部门门户 供应商门户

应用功能层 采购管理平台 操作执行平台 数据交换平台

17:43:52

2.2 软件设计原理与优化准则-模块化

- 模块是一个独立命名的，拥有明确定义的输入、输出和特性的程序实体。模块化即是指整个软件被划分成若干单独命名和独立访问、部分、可编址的部分，称之为模块模块。
- 把一个大型软件系统的全部功能，按照一定的原则合理地划分为若干个模块，每个模块完成一个特定子功能，所有的这些模块以某种结构形式组成一个整体，这就是软件的模块化设计（Modular Design）。
- 把问题 / 子问题的分解与软件开发中的系统 / 子系统或系统 / 模块对应起来，就能够把一个大大而复杂的软件系统划分成易于理解的比较单纯的模块结构。
- 软件模块化设计可以简化软件的设计和实现，提高软件的可理解性和可测试性，并使软件更容易得到维护。

➢ 分解、抽象、逐步求精、信息隐蔽和模块独立性，是软件模块化设计的指导思想。

17:43:52 重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn) 24

2.2 软件设计原理与优化准则-模块化

- **模块分解划分的目的:**
 - ①进行功能分解,把复杂的大的功能划分成简单的小的功能,尽量降低每个模块的成本。
 - ②尽量使每个模块间的接口不能太多,太多会使接口成本增加。兼顾二者可取得最佳的划分状态,确保软件总成本最低。
- **模块化的特征**
 - **抽象:**用层次的方式构造和分析复杂系统。
 - **逐步求精:**帮助开发人员把精力集中在与当前开发阶段最相关的那些问题上。
 - **信息隐蔽:**如果一个模块内包含的信息(过程和数据)不允许外部的模块访问的话,其它模块不能对其访问。
 - **局部化:**把一些关系密切的软件元素物理地放得彼此靠近。

17:43:52 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 25

2.2 软件设计原理与优化准则-模块化-抽象

- **抽象是人类认识复杂现象、解决复杂问题中最有用的思维工具**
- 在现实世界中,一定事物、状态或过程之间总会存在某些相似的方面(共性),把这些相似的方面集中和概括起来,暂时忽略略它们之间的差异,这就是**抽象**。
- 抽象是就是**提取出事物的本质特性而暂时不考虑它们的细节**。抓住**主要问题**,隐藏**细节**,这样才能容易分解。



XX分解图 凤凰抽象图

17:43:52 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 26

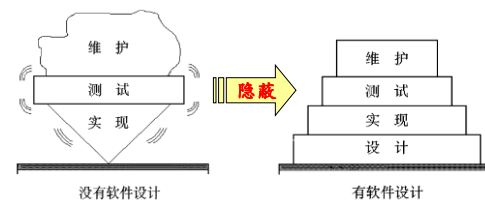
2.2 软件设计原理与优化准则-模块化-抽象

- 一个复杂的软件系统有不同的抽象层次,首先用一些高级的抽象概念来理解和构造,这些高级概念又可以用一些较低级的概念来理解和构造,如此进行下去,直至最低层的具体元素
 - 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抽象层次。在最低的抽象层次用可以直接实现的方叙述问题的解法。

17:43:52 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 27

2.2 软件设计原理与优化准则-模块化-信息隐蔽

- **信息隐蔽原则**建议模块应该具有的特征是:每个模块对其他所有模块都**隐蔽**自己的设计决策。
- 信息隐蔽意味着通过一系列独立的模块可以得到有效的模块化。
- 独立的构件或模块之间的“接口”简单而清晰。

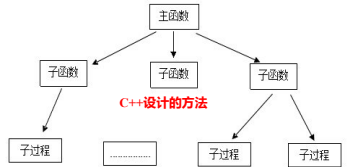


没有软件设计 有软件设计

17:43:52 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 28

2.2 软件设计原理与优化准则-模块化-逐步求精

- **逐步求精**,或称**逐步细化**,是一种**自顶向下**的设计策略。
- 逐步求精是人类采用**抽象到具体**的过程把一个复杂问题趋于简单化控制和管理的**有效策略**。
- **抽象和精化是互补的概念**。



C++设计的方法

17:43:52 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 29

2.3 软件设计原理与优化准则-模块的独立性

- **模块的独立性 (Module Independence)**是模块化、抽象、信息隐蔽等概念的**直接结果**,是判断模块化结构是否合理的**标准**。
- 模块独立性是指开发具有**独立功能**而和其他模块没有**过多关联**的模块。
- 软件系统中每个模块只涉及软件要求的具体的子功能,和其它的模块的接口是简单的。
- 模块的独立性是软件质量的关键。具有独立模块的软件容易开发,这是由于能够对软件的功能加以分割,而相互接口不复杂,可有一组人员同时开发,由于模块相互独立,在各自设计和修改代码时所引起的二次影响不大,错误传播少。
- **模块独立性两大优点:**
 - ✓ 独立的模块由于分解了功能,简化了接口,使得软件比较容易开发;
 - ✓ 独立的模块比较容易测试和维护。

17:43:52 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 30

2.3 软件设计原理与优化准则-模块的独立性

●模块独立性由两个定性标准度量：

- 耦合是模块之间的互相连接的紧密程度的度量，模块之间的连接越紧密，联系越多，耦合性就越高，而其独立性就越弱。
- 内聚是模块功能强度（一个模块内部各个元素彼此结合的紧密程度）的度量。一个模块内部各个元素之间的联系越紧密，则它的内聚性就越高，相对地，它与其它模块之间的耦合性就会减低，而模块独立性就越强。

●模块独立性愈高，则块内联系越强，块间联系越弱。增强模块独立性的方法是尽量做到高内聚、低耦合。

17:43:52

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

31

模块耦合性

耦合是对一个软件结构内不同模块之间互连程度的度量。一个模块在不需要另一个模块的情况下，能够完整地执行其功能，就称这两个模块完全独立。通常模块之间总是相互关联的，因为完全独立的模块是无法构成系统的。关键是连接的程度和复杂度。耦合度描述了模块间的连接程度，但它能用一个可操作的概念来描述，即程序员在编码、调试和修改一个模块时，需要考虑另外一个模块的概率。如果两个模块是紧耦合的，那么他修改一个模块，需要修改另外一个模块的概率就高。显然整个系统的开发费用及系统的复杂性与模块间的耦合度有直接的关系。

低

耦合性

高

非直接耦合	数据耦合	标记耦合	控制耦合	外部耦合	公共耦合	内容耦合
-------	------	------	------	------	------	------

强

模块独立性

弱

17:43:52

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

32

模块耦合性

- 非直接耦合：两个模块之间没有直接关系，它们之间的联系完全是通过主模块的控制和调用来实现的。
- 数据耦合：调用下属模块时，如交换的都是简单参数或变量（类似很多语言中的值传递）。
- 特征耦合：调用下属模块时，交换的是传递记录信息的数据结构。记录是某一数据结构的子结构（如：传递一个对象、表）。
- 控制耦合：模块间传递信息不是一般的数据，而是开关、标志、名字等控制信息。通过传送，明显地控制选择另一模块的功能，就是控制耦合。
- 外部耦合：允许一组模块访问同一个全局变量（简单一个值）。
- 公共耦合：若允许一组模块访问同一个全局性的数据结构，则称它们为公共耦合。全局性的数据结构可以是共享的通信区、公共的内存区域、任何存储介质文件、物理设备等。
- 内容耦合：一个模块可以直接调用另一个模块中的数据，或直接转移到另一个模块中去，或一个模块有多个入口。

17:43:52

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

33

模块耦合性

(a) 松散公共耦合 (b) 紧密公共耦合

控制耦合缺陷：

- i.被调用模块需要知道调用模块的内部结构和逻辑，降低了重用的可能性。
- ii.控制耦合往往是多余的，模块适当分解后可用数据耦合代替它

17:43:52

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

34

模块内聚性

□功能性内聚：模块中各个部分都是完成某一具体功能必不可少的组成部分，或者说该模块中所有部分都是为了完成一项具体功能而协同工作，紧密联系，不可分割。则称该模块为功能内聚模块。

高

内聚性

低

功能内聚	信息内聚	通信内聚	过程内聚	时间内聚	逻辑内聚	巧合内聚
------	------	------	------	------	------	------

强

模块独立性

弱

功能单一

功能分散

17:43:52

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

35

模块内聚性

□信息性内聚：模块中多个功能都在同一数据结构上操作，每一项功能有一个唯一的入口点。模块根据不同的要求，确定执行哪一个功能。模块内所有功能都基于同一个数据结构。信息内聚模块可以看成是多个功能内聚模块的组合，并且达到信息的隐蔽。即把某个数据结构、资源或设备隐蔽在一个模块内，不为别的模块所知晓。

信息性内聚也称为顺序性内聚，模块内的各个任务是顺序执行的。通常，上一个任务的输出是下一个任务的输入。

17:43:52

重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn)

36

模块内聚性

□ **通信性内聚**：如果一个模块内各功能部分都使用了相同的输入数据，或产生了相同的输出数据，则称之为通信内聚模块。通常，通信内聚模块是通过数据流图来定义的。

17:43:52 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 37

模块内聚性

□ **过程性内聚**：模块内的各个任务必须按照某一特定次序执行。

- 使用流程图做为工具设计程序时，把流程图中的某一部分划分出组成模块，就得到过程内聚模块。例如，把流程图中的循环部分、判定部分、计算部分分成三个模块，这三个模块都是过程内聚模块。

模块内各处理单元相关，按特定次序执行，采用数据流图时往往会出现此种内聚。

□ **时间性内聚**：模块内的各个任务由相同的执行时间联系在一起。例如，初始化模块。

- 时间内聚又称为经典内聚。这种模块大多为多功能模块，但模块的各个功能的执行与时间有关，通常要求所有功能必须在同一时间段内执行。例如初始化模块和终止模块。

17:43:52 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 38

模块内聚性

逻辑内聚：指模块内执行若干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。

巧合内聚：指一个模块内的各处理元素之间没有任何联系。

17:43:52 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 39

2.4 软件设计优化准则

- 划分模块时，尽量做到**高内聚、低耦合**，保持模块相对独立性。模块划分的准则：“将相关的各部分放在一起，无关的东西不要放在一起。”
- 模块的大小要适中。
- 模块的接口要简单、清晰、含义明确。便于理解，易于实现、易于测试和维护。
- 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。
- 软件结构的深度、宽度、扇入、扇出应当适当。
- 力求设计单入口和单出口的模块。避免“病态连接”，以防止内容耦合。
- 设计功能可预测模块的划分，应防止功能过分局限。

17:43:52 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 40

2.4 软件设计优化准则

接口设计原则：好的接口应当满足设计模式六大原则，很多设计模式，框架都是基于高内聚低耦合这个出发点的。

- 单一职责原则**：一个类只负责一个功能领域中的相应职责。
- 开闭原则**：一个软件实体应当对扩展开放，对修改关闭。
- 里氏代换原则**：所有引用基类（父类）的地方必须能透明地使用其子类的对象。
- 依赖倒转原则**：抽象不应该依赖于细节，细节应当依赖于抽象。换言之，要针对接口编程，而不是针对实现编程。
- 接口隔离原则**：使用多个专门的接口，而不使用单一的总接口，即客户端不应该依赖那些它不需要的接口。
- 迪米特法则**：一个软件实体应当尽可能少地与其他实体发生相互作用，例如外观模式，对外暴露统一接口。

17:43:52 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 41

2.5 软件设计原理与优化准则-软件结构图

- 软件结构**（Structure Chart, 简称SC）是软件系统的模块层次结构，反映了整个系统的功能实现。是一种控制的层次体系，并不表示软件的具体过程。
- 软件结构一般用树状或网状结构图表示，称为**软件结构图**。
- 软件结构图的主要元素有：
 - 模块**：模块用带有名字的方框表示，名称应体现模块的功能。
 - 控制关系**：控制关系用单向箭头或直线表示模块间的调用关系。
 - 信息传递**：用带注释的短箭头表示模块调用过程中传递的信息。
 - 循环调用和选择调用**：在上部模块底部加一个菱形符号表示选择调用，在上部模块的下方加一个弧形箭头，表示循环调用。

17:43:52 重庆理工大学计算机科学与工程学院 李黎(lililang@cqu.edu.cn) 42

2.5软件设计原理与优化准则-软件结构图

计算每月利息

以功能做模块名

计算月息

以功能的缩写做模块名

打印出错信息

已定义模块

计算月销售额

子程序(或过程)

(a) 模块调用关系

(b) 模块间接口的表示

17:43:52 重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn) 43

2.5软件设计原理与优化准则-软件结构图

结构图的例子——产生最佳解的一般结构

17:43:52 重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn) 44

2.5软件设计原理与优化准则-软件结构图

- 软件结构图的形态特征:**
 - 深度:** 指结构图控制的层次, 也是模块的层数。深度能粗略表示一个系统的大小和复杂程度, 深度和程序长度之间存在着某种对应关系。经验数据一般不超过5。
 - 宽度:** 指一层中最大的模块个数。一般来说, 结构的宽度越大, 则系统就越复杂。经验数据是3~10。
 - 扇出:** 指一个模块直接下属模块的个数。扇出过大, 表示模块过分复杂, 需要控制和协调的下级模块太多。扇出的上限一般为5~9, 平均一般为3或4。
 - 扇入:** 指一个模块直接上级模块的个数。扇入过大, 意味着共享该模块的上级模块数目多, 这有一定的益处, 但是决不能违背模块的独立性原则而片面追求高扇入。
- 画结构图应注意的事项:**
 - 模块不能重名。
 - 调用关系只能从上到下。

17:43:52 重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn) 45

2.5软件设计原理与优化准则-软件结构图

17:43:52 重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn) 46

2.5软件设计原理与优化准则-软件结构图

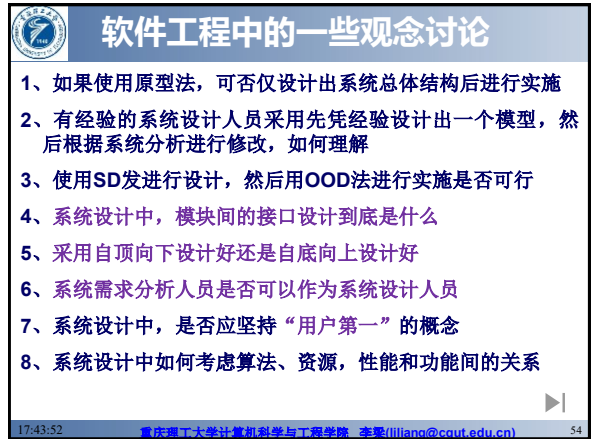
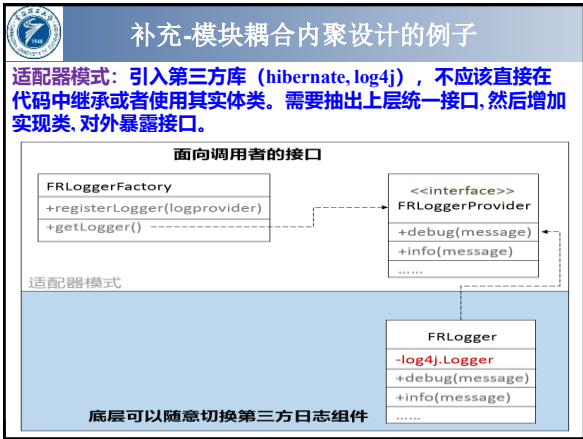
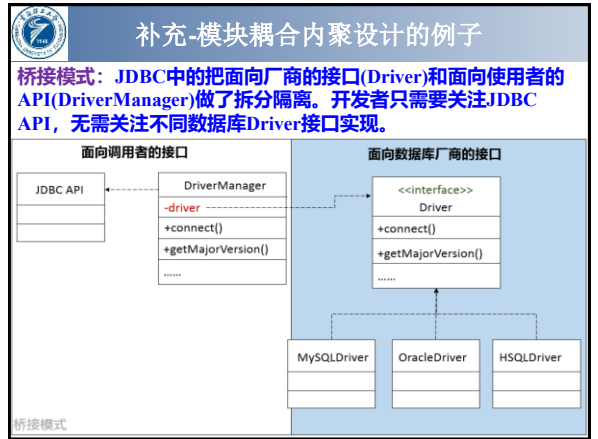
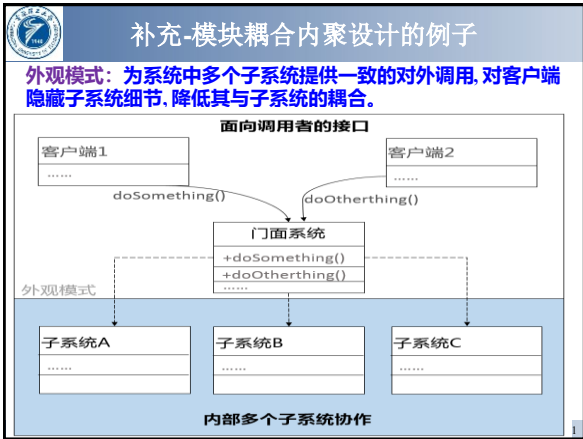
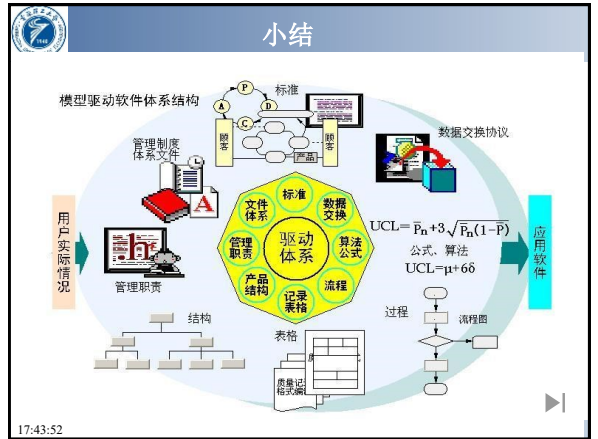
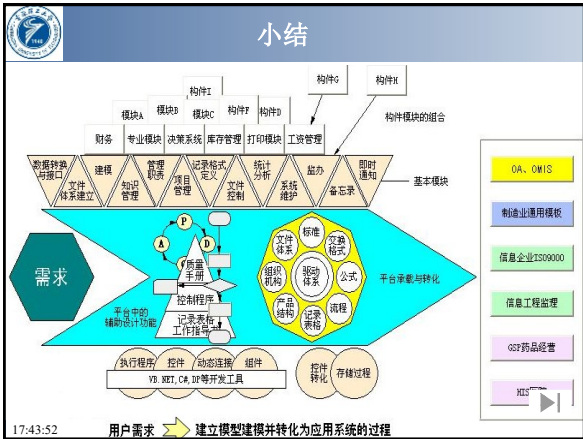
- 改进软件结构提高模块独立性
- 在满足模块化要求的前提下尽量减少模块数量, 在满足信息需求的前提下尽可能减少复杂的数据结构
- 模块规模应适中
- 软件结构的深度、宽度、扇入数和扇出数都要适当
- 模块的作用域应该在控制域之内
- 力求降低模块接口的复杂程度, 设计单入口、单出口的模块

17:43:52 重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn) 47

小结

- 设计的基本原理和概念包括模块化、抽象、体系结构、信息隐蔽、模块独立、逐步求精和重构等, 这些原理和概念描述了计算机软件的属性、所使用的设计方法和所使用的编程语言。
- 设计通常被描述为一个多步过程, 其主要任务是从需求信息中综合出数据的表示、程序结构、接口特征和过程细节。
- 软件体系结构提供了待建系统的整体视图, 它描述软件构件或模块的结构和组织、构件或模块的性质以及他们之间的连接。

17:43:52 重庆理工大学计算机科学与工程学院 李黎(liliang@cqu.edu.cn) 48





软件工程的一些观念讨论

- 9、采用面向对象的实现技术，系统设计到那一步为好
- 10、系统设计时是否要考虑实现技术，分析、调查阶段是否考虑实现技术
- 11、系统设计时是否要考虑系统规模、资金、集成环境等，分析阶段考虑否
- 12、有没有不能完成系统设计的系统，有没有不能实现的系统
- 13、系统分析人员和系统设计人员谁的要求高



17:43:52

青岛理工大学计算机科学与工程学院 李鹏(lilipang@qut.edu.cn)

55