## 名词解释

1. 软件工程（P4）
2. 基线（P58）
3. 软件需求（P71）
4. 用例（93）
5. 软件设计（135Or136？）
6. 软件构造（282）
7. 耦合和内聚

## 问答

1. 需求的分类（P76，6种）
2. 需求分析的任务（82）
3. 界面设计原则（人机交互设计原则）
4. 区分设计与实现的界限（283）
5. 软件构造的主要活动

## 第一章 软件工程基础

1. 名词解释
   1. 软件工程（4）
2. 简答：
   1. 从1950s~2000s之间的特点

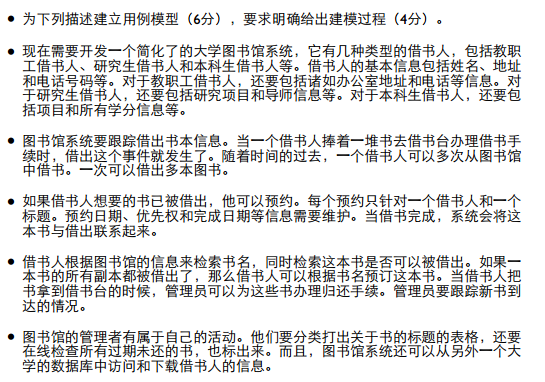
## 第四章 项目启动

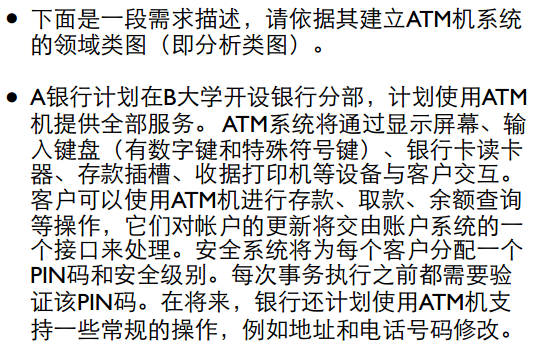
1. 名词解释
   1. 质量属性（54）
   2. 质量模型（54）
   3. 配置管理（58）
   4. 配置项（58）
   5. 基线（58）
2. 软件项目管理活动有哪些？（过程组、具体活动）（49,13年选择）
3. 如何管理团队？
   1. 在实验中采取了哪些办法？有哪些经验？
   2. 团队结构有三种★（50）
4. 质量模型有哪些（h，55）
5. 质量保障有哪些措施？★
   1. 结合实验进行说明★
6. 配置管理有哪些活动？★
   1. 实验中是如何进行配置管理的？★
7. 质量验证的方法有三种（56）
8. 项目的质量保障包括哪些活动？（56）
   1. 例：（2010）结合实验，说明一个项目的质量保障包括哪些活动？
9. 典型的评审过程有六个阶段（56）
10. 变更控制过程是怎样的（61）

## 第五章 软件需求基础

1. 名词解释
   1. 需求（71）
2. 需求获取的三种方法（69）
3. 区分需求的三个层次（71）
   1. *给出一个实例，给出其三个层次的例⼦*
   2. 对给定的需求示例，判定其层次
      1. 例如课程实验/ATM/图书管理…
4. *掌握需求的类型*
   1. 对给定的实例，给出其不同类型的需求例⼦
   2. 对给定的需求⽰例，判定其类型
      1. 例如课程实验/ATM/图书管理…

## 第六章 需求分析方法

1. 为给定的描述
   1. 建立用例图
   2. 建立分析类图
   3. 建立系统顺序图
   4. 建立状态图
   5. 为下列描述建立用例模型（6分），要求明确给出建模过程（4分）。
   * 



1. 简：需求分析的方法
2. 简：需求分析模型（84，6种）
3. 简：结构化分析方法（85，2种）
4. 简：面向对象方法（4种）
5. 数据流图怎么画
6. 实体关系图怎么画
7. 用例描述怎么写
8. 概念类图怎么画
9. 顺序图怎么画
10. 状态图怎么画

## 第七章

1. 为什么需要需求规格说明？结合实验进⾏说明
2. 对给定的需求⽰例，判定并修正其错误
   1. 对给定的需求规格说明⽚段，找出并修正其错误
3. 对给定的需求⽰例，设计功能测试⽤例
   1. 结合测试⽅法

## 第八章 软件设计基础

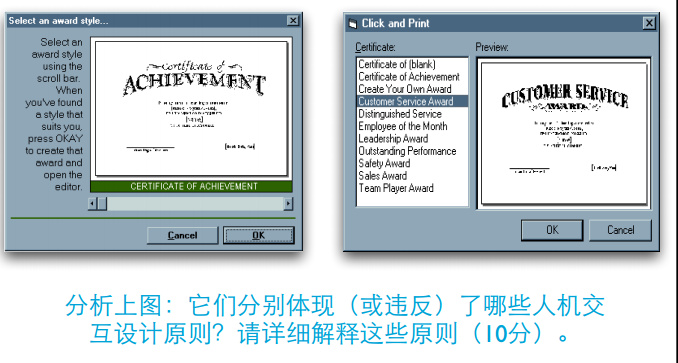
1. 名词解释
   1. 软件设计
2. 软件设计的核⼼思想是什么？
3. 软件⼯程设计有哪三个层次？各层的主要思想是什么？软件设计的核心思想是什么（139）
4. 软件设计设计过程的四个主要活动（140）
5. 软件设计的方法（140）
6. 常见的设计视角

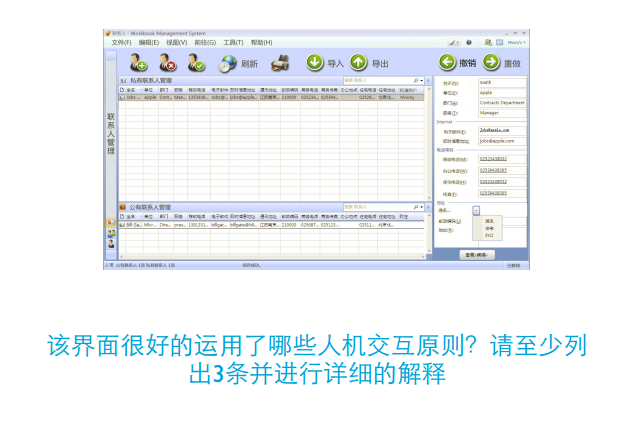
## 第九、⼗章

1. 体系结构的概念
2. 体系结构的风格的优缺点★
3. 体系结构设计的过程？
4. 包的原则
5. 体系结构构建之间接⼝的定义（\*）
6. 体系结构开发集成测试⽤例
   1. Stub和Driver

## 第⼗⼀章

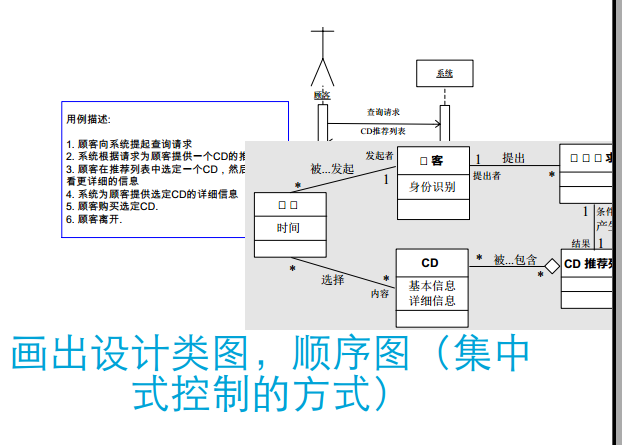
1. 名词解释
   1. 可⽤性
2. 能够列出⾄少5个界⾯设计的注意事项，并加以解释
   1. 例⼦\*\*违反了哪些条界⾯设计原则
3. 精神模型、差异性
4. 导航、反馈、协作式设计





第⼗⼆章

1. 详细设计的出发点
2. 职责分配★
3. 协作★
4. 控制风格（208）★
5. 给定分析类图、系统顺序图和设计因素描述
   1. 建⽴设计类图★
   2. 或者详细顺序图★
6. 协作的测试
   1. MockObject



## 第⼗三章 模块化与信息隐藏

1. 名词解释
   1. 解释耦合与内聚★
2. 耦合与内聚★
   1. 对例⼦\*\*，说明它们之间的耦合程度与内聚，给出理由★
3. 六种耦合解释及例子
4. 七种内聚解释及其例子
5. 信息隐藏★
   1. 基本思想★
   2. 两种常⻅的信息隐藏决策
   3. 对例⼦\*\*，说明其信息隐藏程度好坏★

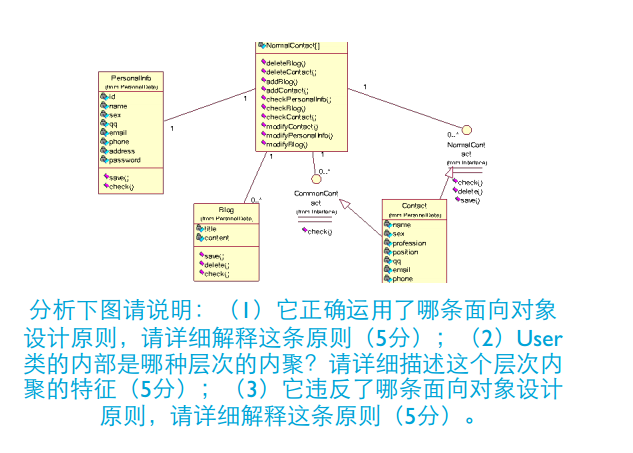


## 第⼗四章 面向对象下的模块化

1. Principles from Modularization 模块化原则
   1. 《Global Variables Consider Harmful》 全局变量考虑有害
   2. 《To be Explicit》 要明确
   3. 《Do not Repeat》 不要重复
   4. Programming to Interface (Design by Contract) 面向接口编程（抽象设计）
   5. The Law of Demeter 迪米特法则
   6. Interface Segregation Principle(ISP) 接口分离原则
   7. Liskov Substitution Principle (LSP) Liskov替换原则（里氏代换原则）
   8. Favor Composition Over Inheritance 使用组合代替继承
   9. 集中信息与行为的原则
   10. Single Responsibility Principle 单一职责原则
2. • 对给定的⽰例，发现其所违反的原则，并进⾏修正

## 第⼗五章 面向对象下的信息隐藏

1. 信息隐藏的含义
2. 封装
3. 多态
4. OCP 开闭原则
5. DIP 依赖倒置原则



## 第⼗六章 详细设计的设计模式

1. 如何实现可修改性、可扩展性、灵活性
2. 策略模式
3. 抽象⼯⼚模式
4. 单件模式
5. 迭代器模式
6. *给定场景，应⽤设计模式并写出代码*
7. *给出代码，要求⽤设计模式改写*

## 第⼗七、⼗⼋章

1. 构造包含的活动
2. 名词解释
   1. 重构
   2. 测试驱动开发
   3. 结对编程
3. 给定代码段⽰例，对其进⾏改进或者 发现其中的问题
   1. 简洁性/可维护性
   2. 使⽤数据结构消减复杂判定
   3. 控制结构
   4. 变量使⽤
   5. 语句处理
   6. How to write unmaintainable code
   7. 防御与错误处理
4. 单元测试⽤例的设计
5. 契约式设计代码设计
6. 防御式编程代码设计
7. 表驱动

## 第⼗九章

1. 掌握⽩盒测试和⿊盒测试的常⻅⽅法，并进⾏能够优缺点⽐较
2. 能解释并区别⽩盒测试三种不同的⽅法：语句覆盖、分⽀覆盖和路径覆盖
3. 给出⼀个场景，判断应该使⽤哪种测试⽅法，如何去写（\*）
   1. 对给定的场景和要求的测试⽅法，设计测试⽤例
      1. 给出功能需求，则要求写功能测试⽤例
      2. 给出设计图，则要求写集成测试⽤例，Stub and Driver
      3. 给出⽅法的描述，则要求写单元测试⽤例，Mock Object
      4. JUnit基本使⽤⽅法

## 第⼆⼗、⼆⼗⼀章

1. 如何理解软件维护的重要性？
2. 开发可维护软件的⽅法
3. 演化式⽣命周期模型
4. ⽤户⽂档、系统⽂档
5. 逆向⼯程、再⼯程

## 第⼆⼗⼆、⼆⼗三章

1. 软件⽣命周期模型
2. Build-and-fix model
3. Waterfall model
4. Iterative Models
   1. Incremental model
      1. Incremental Delivery
   2. Evolutionary
      1. Evolutionary development
      2. Prototyping
      3. Spiral Model
5. 解释与⽐较不同过程模型（要求、特征描述、优点、缺点）
6. 对给定的场景，判定适⽤的开发过程模型
7. 软件⼯程知识体系的知识域

## 其他

1. 什么是可用性设计

产品在特定使用环境下为特定用户用于特定用途时所具有的有效性（effectiveness）、效率（efficiency）和用户主观满意度（satisfaction）。其中：

有效性 -用户完成特定任务和达到特定目标时所具有的正确和完整程度；

效率 -用户完成任务的正确和完整程度与所使用资源（如时间）之间的比率；

满意度 -用户在使用产品过程中所感受到的主观满意和接受程度。

可用性设计就是在以用户为中心的宗旨下，进行产品（系统）的设计，以使产品满足功能需要、符合用户的行为习惯和认知，同时能高效愉悦的完成任务和工作，达到预期的目的。

============================