软件设计与体系结构

第1章 软件工程与软件设计

1.软件工程的概念

1. 软件工程是将系统的、规范的、可度量的的方法应用于软件的开发、运行和维护过程，以及对上述方法的研究
2. 软件工程是用工程、科学和数据的原则与方法，研制、维护计算机软件的有关技术及管理方法

2.软件开发过程模型

1. 瀑布模型
    - 特点：开始就**需求明确**，任务是阶段性逐步执行的，上个阶段的任务失误会蔓延到以后的各个阶段的任务；
    - 局限性：在软件开发的初始阶段就确定软件的全部需求是困难的，不适合大规模软件开发；需求确认后，需要较长的时间才能得到软件初版，再修改损失较大；
2. 快速原型模型
    - 类比建筑项目，根据用户的要求先**快速开发一个初版**，再根据用户意见进行修改
3. 螺旋模型
    - **瀑布模型和原型模型结合**，并**增加了风险分析**，逐步进行软件开发，边开发边评审
    - 组成：需求定义，风险分析，工程实现，评审（不断循环执行）
    - 用户始终参与软件开发的评审，保证软件质量；**支持大型软件开发**
4. 统一软件开发过程
    - 组成：迭代时开发、需求管理、基于构件的软件体系结构、可视化规模、验证软件质量、控制软件变更
    - 4个阶段：**先启阶段，精化阶段，构建阶段，产品化阶段**
    - 工作流程：业务建模，需求，分析与设计，实施，测试，部署，配置与变更管理，项目管理，环境
    - **先启阶段：业务建模和需求； 构建阶段：分析与设计，开发，测试**

3.软件设计

1. 软件设计的重要性(必考)
    - 软件设计是对**软件需求的直接体现**
    - 软件设计为**软件实现提供直接依据**
    - 软件设计将**综合考虑软件系统的各种约束条件并给出相应方案**
    - 软件设计的质量将**决定最终软件系统的质量**
    - **及早发现**软件设计中存在的错误将**极大减少软件修复和维护所需的成本**
2. 软件设计的特征

- 软件设计的**开端**是出现某些新的问题需要软件来解决，这些需要促使设计工作的开始，并成为整个设计工作最初的基础
- 软件设计的**结果**是给出一个方案，它能够用来实现所需的、可以解决问题的软件，方案的描述可能是文字、图表，甚至是数学符号、公式等组成的文档或模型
- 软件设计包含一**系列的转换过程。**
- 产生**新的想法或思路**对软件设计非常重要
- 软件设计的过程是**不断解决问题和实施决策的过程**
- 软件设计是一个**满足各种约束的过程**
- 大多数软件设计是一个**不断演化的过程**

3. 软件设计的要素
    - 目标描述
    - 设计约束
    - 产品描述
    - 设计原理
    - 开发规划
    - 使用描述

第2章 统一建模语言UML

1.UML的特点和用途

- 为使用者提供了**统一的，表达能力强大的可视化建模语言**，以描述应用问题的需求模型、设计模型和实现模型
- 提供对**核心概念的扩展机制**，用户可加入核心概念中没有的概念和符号，可为特定应用领域提出具体的概念、符号表示和约束
- **独立于实现语言和方法学，但支持所有的方法学**，覆盖了面向对象分析和设计的相关概念和方法学
- **独立于任何开发过程，但支持软件开发全过程**
- 提供对**建模语言进行理解的形式化基础**，用元模型描述基本语义，OCL描述良定义规则，自然语言描述动态语义
- 增强**面向对象工具之间的互操作性**，便于不同系统间的集成
- 支持**较高抽象层次开发所需的各种概念**，如协同、框架、模式和构件等，便于系统的重用

2. UML 2.0的建模机制

1. 结构建模
    - 类图：描述系统的静态逻辑结构，包括关联、聚集、继承和依赖关系
    - 包图：特殊类型的类图，描述类和接口如何进行逻辑划分
    - 对象图：类图的实例
    - 构件图：描述了系统实现中的结构和依赖关系
    - 组合结构图：描述较为复杂的系统元素以及元素之间的关系

- 部署图：描述系统硬件的物理拓扑结构以及在此结构上运行的构件

2. 行为建模
    - 活动图：描述行为或行动的流程
    - 交互图：
        1. 顺序图：描述对象在其生存周期内的交互活动
        2. 通信图：描述特定行为中参与交互的对象及其连接关系
        3. 交互概览图：活动图的简化版本，强调执行活动所涉及的元素
        4. 时序图：强调消息的详细时序说明
    - 状态图：刻画一个元素内部的状态迁移
    - 用例图：通过用例来描述系统的功能性需求

3.习题二第五题：

开发一个简单的网络购物平台，可以实现基本的用户登录、浏览商品、购买商品、生成订单、支付等功能。请画出该系统中存在的一些主要的类图。

第3章 软件设计基础

1.软件设计的基本概念：

软件设计主要针对需求分析过程中得到的软件需求规格说明，综合考虑各种制约因素，探求切实可行的软件解决方案并最终给出方案的逻辑表示，包括文档、模型等。

2.软件设计过程（问答题）

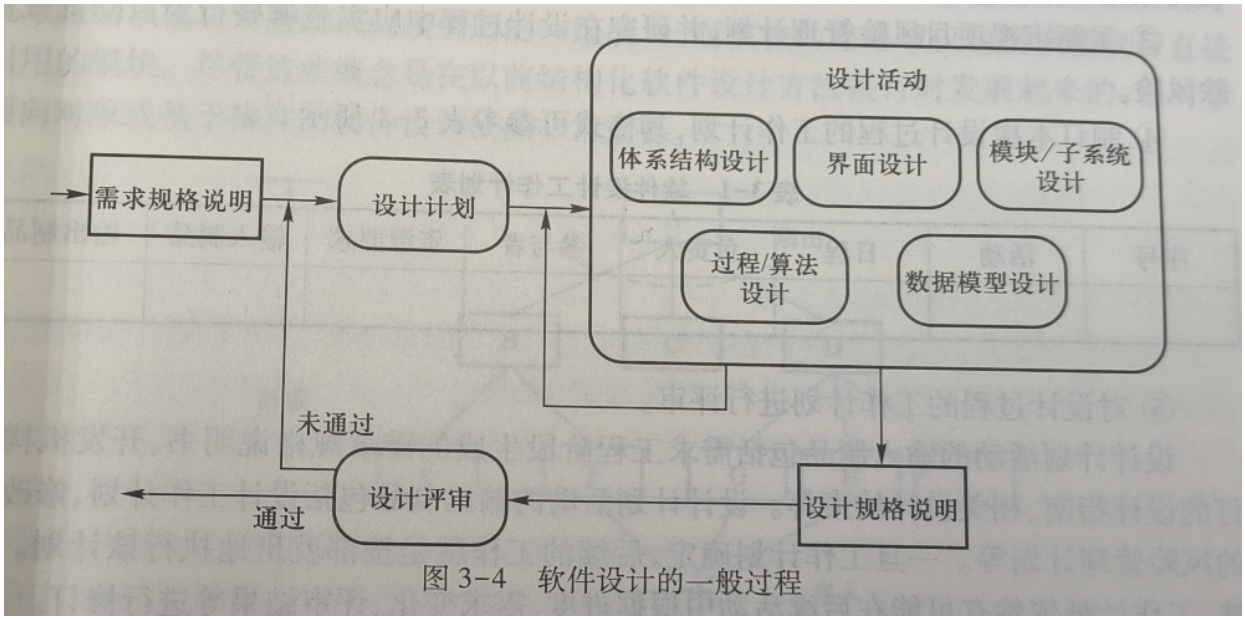

图 3-4　软件设计的一般过程

- 各个活动以需求阶段产生的需求规格说明为基础
- 首先对整个设计过程进行计划
- 然后实施具体的设计活动，这些设计活动本身可能是一个不断迭代和精化的过程
- 在设计活动完成后应形成规格说明

- 然后对设计过程和设计规格说明进行评审
    - 评审未通过再次修订设计计划并对设计进行改进
    - 评审通过则进入后续阶段

软件设计的主要活动

- 软件设计计划
- 体系结构设计
- 界面设计
- 模块/子系统设计
- 过程/算法设计
- 数据模型设计

3.软件设计的质量（问答题）

1. 对软件设计的质量进行综合评价
    - 结构良好
    - 充分性
    - 可行性
    - 简单性
    - 实用性
    - 灵活性
    - 健壮性
    - 可移植性
    - 可复用性
    - 标准化
2. 软件设计对最终软件产品的质量产生的影响有
    - 正确性
    - 可靠性
    - 运行效率
    - 可移植性
    - 可复用性
3. 软件设计对软件开发过程可能产生的影响包括
    - 开发效率
    - 交付时间
    - 风险管理
    - 资源管理
    - 成本

- 人员培训
- 合法性

第4章 面向对象的软件设计方法

1.设计精化（问答题）

1. 设计精化的任务
    - 精化软件架构
    - 调整软件构成类
    - 精化交互模型
    - 精化类之间的关系

作用：对设计模型再进行分析、细化和优化，以生成高质量的设计模型，为后续的实现阶段奠定基础

2.部署模型设计（问答题）

1. 部署模型设计需要考虑以下几点
    - 最终开发完成的软件包括那些制品形式
    - 软件运行环境存在哪些类型的物理节点
    - 不同节点之间的连接和通信形式是什么
    - 软件制品应该如何在物理机节点上进行部署，即他们的部署映射关系

3.案例分析，市场?

第5章 面向数据流的软件设计方法

1.事务流

第6章 用户界面设计

1.言之有理即可

第7章 软件体系结构风格与设计模式

1.设计模式

表 7-1 设计模式分类

		目标		
		创建型	结构型	行为型
范围	类	工厂方法(Factory Method)	适配器(Adapter)	解释器(Interpreter) 模板方法(Template Method)
	对象	抽象工厂(Abstract Factory) 构建者(Builder) 原型(Prototype) 单件(Singleton)	适配器(Adapter) 桥接(Bridge) 组合(Composite) 装饰器(Decorator) 外观(Facade) 享元(Flyweight) 代理(Proxy)	职责链(Chain of Responsibility) 命令(Command) 迭代器(Iterator) 中介者(Mediator) 备忘录(Memento) 观察者(Observer) 状态(State) 策略(Strategy) 访问者(Visitor)

- 创建型

1. 工厂方法：使得父类可集中描述公共行为，而将特别行为抽放于子类
2. 抽象工厂：将公共的创建行为描述为一个抽象类，而将具体的创建方式用该抽象类的子类描述；创建产品时，可以将多个抽象类中的子类组合在一起，组合出不同类型的产品
3. 单件：一个类最多只有一个实例

- 结构型

1. 组合：有“递归组合”的特征（文本、图像组合起来A，A又和文本组合起来B，B又和文本、图像组合起来C)
2. 代理：构造一个有相同接口的代理对象，将操作请求转发给真实对象，目的是像客户隐藏“转发细节”，提供对真实对象的透明访问；typec转为USB

- 迭代器

1. 迭代器：遍历一个聚合对象中的各个元素
2. 观察者：一个对象有一对多的依赖关系，对象的数据改变——>所有依赖它的对象都得到通知并自动更新

第8章 基于分布构件的体系结构（概念题）

第9章 软件体系结构评估（概念题）

1.软件体系结构评估概述

1. **目的**：在开发过程早期，通过分析系统的质量需求是否在软件体系结构中得到体现来识别软件体系结构设计中的潜在风险，预测系统质量属性，并辅助软件体系结构决策的制定
2. **包括**
    - **评估时机和参与人员**：早评估和晚评估；评估团队和利益相关人员
    - **评估结果和质量属性:**
        - 性能

        - 可靠性
        - 可用性
        - 安全性
        - 可变性
        - 可移植性
        - 功能性
        - 变化性
        - 可分解性
        - 概念完整性
    - **评估的益处和代价：**
        - 把利益相关人员召集在一起
        - 强制特定质量目标的结合
        - 生成冲突目标的优先级
        - 对软件体系结构有一个清晰的说明
        - 提高软件体系结构文档的质量
        - 发现跨项目重用的机会
        - 都得到优化后的软件体系结构

2.软件体系结构评估方法

1. ATAM方法
    - 介绍
        - 介绍ATAM方法
        - 商业动机介绍
        - 软件体系结构介绍
    - 调查和分析
        - 确定软件体系结构方案
        - 产生质量属性效果树
        - 分析软件体系结构方案
    - 测试
        - 集体讨论并确定场景的优先级
        - 进一步分析软件体系结构方案
    - 报告
        - 展示结果
2. SAAM方法
    - SAAM方法的输入
    - SAAM方法的输出

        - 场景的形成
        - 描述软件体系结构
        - 场景的分类和优先级划分
        - 间接场景的单独评估
        - 评估场景交互
        - 形成总体评估
3. ARID方法
    - 排练
        - 确定评审人
        - 准备设计情况介绍
        - 准备种子场景
        - 准备材料
    - 评审
        - 介绍ARID方法
        - 介绍设计
        - 场景的集体讨论和优先级划分
        - 应用场景
        - 总结

问题：包括具体用了什么样的方法？？？不止用一个

总结

1.技巧：

对软件设计的理解，言之有理即可

2.题型

1. 5个简答题 50分
2. 1个设计题 30分
3. 1个开放性题目 20分