

【软考达人】

软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题



微信扫一扫，立马获取



6W+ 免费题库



免费备考资料

PC版题库: ruankaodaren.com

软考系统架构师案例分析知识点整理

系统规划：包括系统项目的提出预可行性分析；系统方案的制定、评价和改进；新旧系统的分析和比较；现有软件、硬件和数据资源的有效利用；

软件架构设计：XML 技术；基于架构的软件开发过程；软件的质量属性；架构（模型）风格；特定领域软件架构；基于架构的软件开发方法；架构评估；软件产品线；系统演化

设计模式：设计模式概念；设计模式的组成；模式和软件架构；设计模式分类；设计模式实现；

系统设计：处理流程设计；人机界面设计；文件涉及；存储设计；数据库设计；网络应用系统的设计；系统运行环境的集成与设计；中间件；应用服务器；性能设计与性能评估；系统转换设计计划；

软件系统建模：系统需求、建模的作用以及意义；定义问题（目标、功能、性能）与归结模型（静态结构模型、动态行为模型、物理模型）；结构化系统建模；数据流图；面向对象系统建模；统一建模语言（UML）；数据库建模；E-R图；逆向工程；

分布式系统设计：分布式通行协议的设计；基于对象的分布式系统设计；基于 web 的分布式系统设计；基于消息和协同的分布式系统设计；异构分布式系统的互操作性设计；

嵌入式系统设计：实时系统和嵌入式系统特征；实时任务调度和多任务设计；中断处理和异常处理；嵌入式系统的开发设计

系统的可靠性分析与设计：系统故障模型和可靠性模型；系统的可靠性分析与可靠度计算；提高系统可靠性的措施；系统的故障对策和系统的备份与恢复；

系统安全性和保密性设计：系统的访问控制技术；数据的完整性；数据与文件的加密；通信的安全性；系统的安全性设计；

1、 概念类

系统规划

项目计划：包括范围计划、工作范围计划、活动定义、资源需求、资源计划、活动排序、费用估算、进度计划、费用计划；项目辅助计划包括质量计划、沟通计划、人力资源计划、风险计划、采购计划。

虚拟化技术：计算元件在虚拟的基础上运行；有完全虚拟化，准虚拟化，操作系统层虚拟化等；

虚拟化收益：

- 1、 有效提高服务器资源利用率
- 2、 支持运行在不同操作系统之上的多个业务共享一台服务器
- 3、 减少服务器数量，降低硬件成本
- 4、 节约场地面积，减少能耗

软件架构设计

架构风险：架构设计中潜在的、存在问题的架构决策所带来的隐患

敏感点：为了实现某种特定的质量属性，一个或者多个系统组件所具有的特性

权衡点：影响多个质量属性，并且对多个质量属性来说都是敏感点的系统属性

软件质量属性包括：功能性、性能、可用性、可靠性、健壮性、安全性、可修改性、可变性、易用性、可测试性、互操作性

六种质量属性策略：

1、 可用性

错误检测：命令/响应，心跳机制，异常监控

错误恢复：表决（裁决表），主动冗余，被动冗余，备件，状态再同步，检查点/回滚

错误预防：从服务中删除，事物（要么全成功，要么全失败），定期重置，进程监视器

2、 可修改性

局部化修改：维持语义的一致性，预期期望的变更，泛化该模块，限制可能的选择

防止连锁反应：信息隐藏，维持现有的接口，限制通信路径，仲裁者的使用

推迟绑定时间：运行时注册，配置文件，多态，构件更换

3、 性能

资源需求：减少处理时间所需的资源，减少所处理事件的数量，控制资源使用，限制执行时间

资源管理：引入并发，维持数据或计算的多个副本，增加可用资源

资源仲裁：先进/先出，固定优先级，动态优先级调度，静态调度

4、 安全性

抵抗攻击：对用户进行身份验证，对用户进行授权，维护数据的机密性，维护完整性，限制暴露的信息，限制访问

检测攻击：部署入侵检测系统

从攻击中恢复：恢复，识别攻击者

5、 可测试性

输入/输出：记录/回放，将接口—实现分离，优化访问线路/接口

内部监控：当监视器处于激活状态时，记录事件

6、 易用性

运行时：任务模型，用户模型，系统模型

设计时：将用户接口与应用的其余部分分离

支持用户主动：支持用户主动操作

软件架构风格：描述某一类特定应用领域中软件系统组织方式和惯用方式；组织方式描述了系统的组成构建和这些构建的组织方式，惯用模式则反映众多系统共有的结构和语义。

1、 数据流风格

批处理序列：每个处理步骤是一个独立的程序，每一步必须在前一步结束后才能开始，数据必须是完整的，以整体的方式传递

管道/过滤器：每个构件都有一组输入和输出，构件接受数据输入，经过内部处理，然后产生输出数据流。这里构件被称为过滤器，连接件就是数据流传输的管道。

2、 调用/返回风格

主程序/子程序：计算构件作为子程序协作工作，并由一个主程序顺序地调用这些子程序，构件通过共享存储区交换数据。

数据抽象和面向对象：将数据表示和基本操作封装在对象中。

层次结构：构件组织成一个层次结构，每层为上一层提供服务，使用下一层的服务，只能见到与自己邻接的层。

3、 独立构件风格

进程通信：构件是独立的过程，连接件是消息传递。特点是构件通常是命名过程，消息传递的方式可以是点到点、异步或同步方式，及远程过程（方法）调用

事件驱动的系统：构件不直接调用一个过程，而是触发或广播一个或多个事件。构件中的过程在一个或多个事件中注册，当一个事件被触发，系统自动调用在这个事件中注册的所有过程。

4、 虚拟机风格

解释器：一个解释器通常包括完成解释工作的解释引擎，一个包含将被解释的代码的存储区，一个记录解释引擎当前工作状态的数据结构，以及一个记录源代码被解释执行的进度的数据结构。

基于规则的系统：包括规则集、规则解释器、规则/数据选择器及工作内存。

5、 仓库风格

数据库系统：主要有两大类，一个是中央共享数据源，保存当前系统的数据状态；另一个是多个独立处理元素，处理元素对数据元素进行操作。

黑板系统：是一个全局数据库，包含解域的全部状态，是知识源互相作用的唯一媒介。

超文本系统：是一种非线性的网状信息组织方法，以节点为基本单位，链作为节点之间的联想式关联。

6、 复制风格

复制仓库：通过利用多个进程提供相同的服务，来改善数据的可访问性（accessibility of data）和服务的可伸缩性（scalability of service）。

缓存：复制个别请求的结果，以便可以被后面的请求重用。

控制环路架构风格：将过程输出的指定属性维护在一个特定的参考值。

企业服务总线（ESB）：是传统中间件技术与 XML、Web 服务等技术结合的产物，主要支持异构系统集成。ESB 基于内容的路由和过滤，具备复杂数据的传输能力，并可以提供一系列的标准接口。

ESB 的主要功能：

- 1、 应用程序的位置透明性
- 2、 输出协议转换
- 3、 消息格式转换

- 4、消息路由
- 5、消息增强
- 6、安全支持
- 7、监控和管理

设计模式

创建模式：主要用于创建对象，为设计类实例化新对象提供指南

包括：工厂方法（Factory Method）、抽象工厂（Abstract Factory）、单例（Singleton）、构建（Builder）、原型（Prototype）

结构模式：主要用于处理类或对象的组合，对类如何设计以形成更大的结构提供指南

包括：适配器（Adapter）、合成（Composite）、装饰（Decorator）、代理（Proxy）、享元（Flyweight）、门面（Facade）、桥接（Bridge）

行为模式：主要用于描述类或对象的交互以及职责的分配，对类之间交互以及分配责任的方式提供指南

包括：策略（Strategy）、模版方法（Template Method）、迭代器（Iterator）、责任链（Chain of Responsibility）、命令（Command）、备忘录（Mediator）、状态（State）、访问者（Visitor）、解释器（Interpreter）、调停者（Mediator）、观察者（Observer）

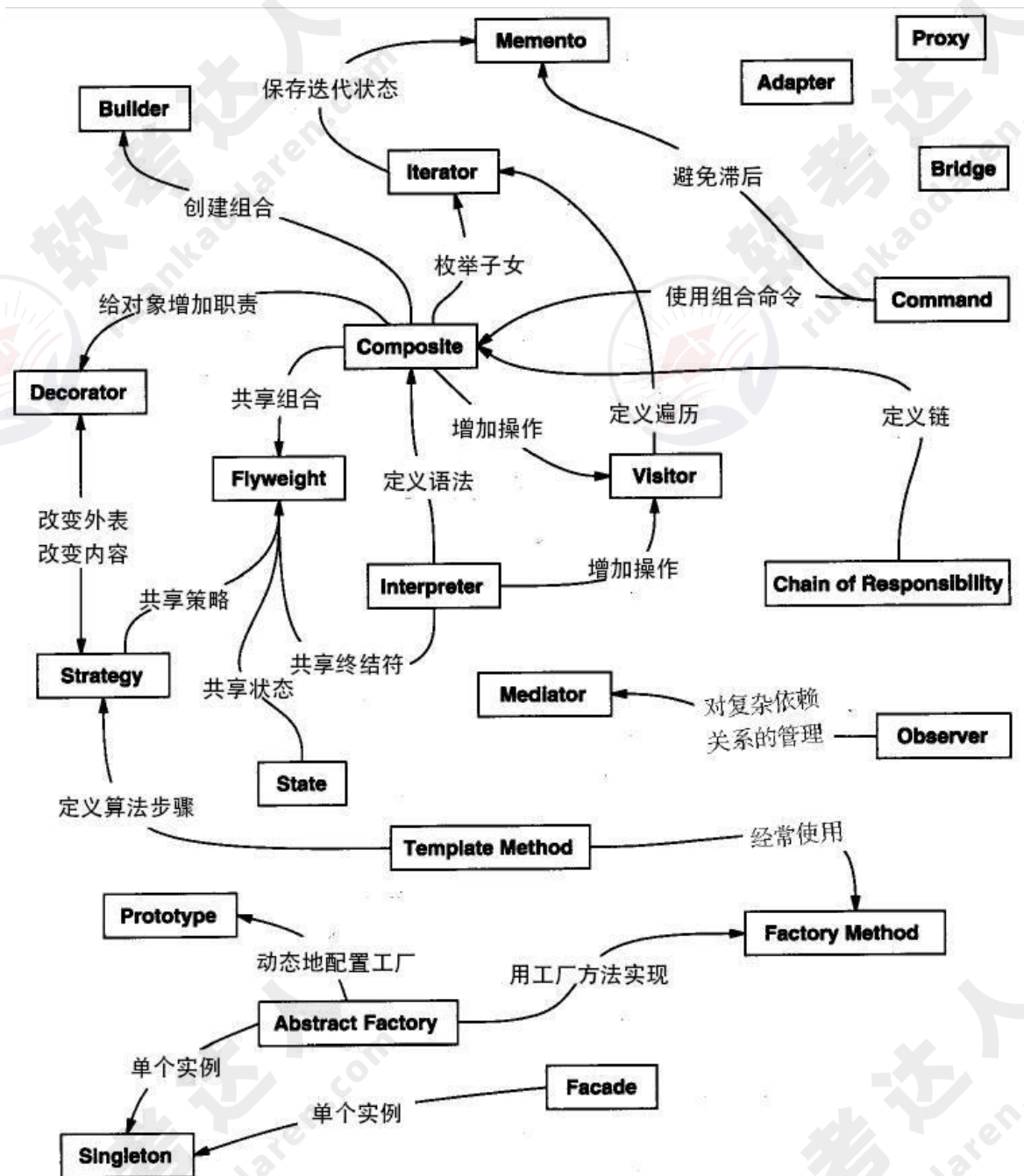


图 设计模式之间的关系 blog.csdn.net/u013963737

系统设计

数据流图：作为一种图形化工具，用来说明业务处理过程、系统边界内所包含的功能和系统中的数据流；

流程图：以图形化的方式展示应用程序从数据输入开始到获得输出为止的逻辑过程，描述处理过程的控制流；

JRP（联合需求分析会议）：通过召开一系列高度结构化的分组会议，快速的分析问题、定义需求；

软件系统建模

用例建模：描述参与者和系统之间的主要交互；用例建模可以描述利益相关者所看到的系统行为；

组件建模：确定系统的子系统、模块和组件结构，为子系统、模块分配需求和职责，每个组建元素作为一个自包含的单元，用于开发、部署和执行。

服务建模：提供了通用的应用程序，并将应用程序定义为一组抽象服务接口。

性能建模：是对系统的性能进行度量，为每个组件确定性能指标。包括执行时间、资源使用、开发复杂性、维护复杂性等。

分布式系统设计

嵌入式系统设计

系统的可靠性分析与设计

冷备份：在数据库关闭的情况下，对数据库中的关键文件进行

热备份：在数据库运行的情况下，对数据库中的关键数据进行备份，要求数据库管理系统提供支持

系统安全性和保密性设计

信息系统的安全威胁来自于：

- 1、 物理环境：对系统所用设备的威胁，如：自然灾害，电源故障，数据库故障，设备被盗等造成数据丢失或者信息泄露
- 2、 通信链路：传输线路上安装窃听装置或者对通信链路进行干扰
- 3、 网络系统：由于因特网的开放性、国际性、无安全管理性，对内部网络形成严重的安全威胁
- 4、 操作系统：操作系统本身的后门或者安全缺陷，如木马和陷阱门
- 5、 应用系统：网络服务或者用户业务系统安全的威胁，包括应用系统自身漏洞
- 6、 管理：人员管理和各种安全管理制度

用户认证方式：

- 1、 用户名和口令认证
- 2、 基于公钥、签名的认证方式
- 3、 持卡认证方式
- 4、 基于人体生物特征的别认证方式

对称加密策略：

机密性：发送者利用对称密钥对要发送的数据进行加密，只有拥有相同密钥的接收者才能正确解密，从而提供机密性；

完整性：发送者根据要发送的数据生成消息摘要，利用对称密钥对消息认证进行加密并附加到数据上发送；接收者使用相同的密钥将对方发送的消息认证码解

密，并根据接收到的数据重新生成消息认证码，比较两个认证码是否相同以验证数据的完整性

公钥加密策略：

机密性：发送者利用接受者的公钥对要发送的数据进行加密，只有拥有对应私钥的接收者才能将数据正确解密，从而提供机密性

完整性：发送者根据要发送的数据生成消息摘要，利用自己的私钥对消息认证码加密并且附加到数据上发送；接收者利用对方的公钥将对方发送的消息认证码解密，并根据接收到的数据重新生成消息认证码，比较两个认证码是否相同以验证数据完整性

2、优点缺点类

系统规划

软件架构设计

设计模式

系统设计

区别	数据流图	流程图
	处理过程可以并行	某个时间点只能处于一个处理过程
	展现系统的数据流	展现系统的控制流
	展现全局的处理过程，过程之间计时标准不同	处理过程遵循一致的计时标准
	适用于系统分析中的逻辑建模阶段	适用于系统设计中的物理建模阶段

传统系统集成方案	基于 Web2.0 的 Mashup 技术
需要传统平台 API 层的支持，集成过程复杂	仅需要用现有的 Web 应用程序公开的基于 Web 的 API（Web 服务）构建应用程序，集成过程简单
需要采用编程的方式完成系统功能扩展，设计数据存储层、业务逻辑层和表现层	直接使用 Mashup 技术集成两个或者多个 WebAPI，创建新的特性与功能
传统集成方式下的表现层大多采用静态展现技术，当客户端发送请求时，需要刷新整个页面	用 Ajax 技术调用基于 Web 的 API，浏览器客户端不需要在每次与服务器通行时重新加载整个页面，动态特性强
使用传统的多层企业应用集成技术，涉及不同的集成层次	采用 SOA 思想集成底层系统，强调功能暴露与服务，以服务形式集成并暴露有系统的能力

软件系统建模

分布式系统设计

分布式数据库的特点有：数据独立性与位置透明性。集中管理和节点自治相结合、支持全局数据库的一致性和可恢复性、复制透明性（适度数据冗余）、易于扩展性。

分布式数据库的优点：

- 1、具有灵活的体系结构
- 2、适应分布式的管理和控制结构
- 3、经济性能优越
- 4、系统可靠性高，可用性好
- 5、局部应用的响应速度快

6、可扩展性好，易于集成现有系统

缺点：

- 1、系统开销大，主要花在通信部分
- 2、复杂的存储结构
- 3、数据的安全性和保密性较难处理
- 4、保持数据一致性算法复杂

分布式数据库中各局部数据库应该满足集中式数据库的基本需求，除此之外还应保证数据库的全局数据一致性，并发操作可串行性，故障的全局可恢复性；

保证方法：

一致性：数据副本一致性，保证分布式事务的 ACID 属性，故障恢复的一致性

嵌入式系统设计

系统的可靠性分析与设计

备份方式	优点	缺点
冷备份	快速简单，维护方便	备份时，数据库关闭，不能做其他工作
	可以恢复到备份时的时间点上	单独使用时，只能恢复到备份时的时间点上，数据丢失
	与热备份相结合，实现实时/秒级恢复	不能按数据库中的表或某个用户进行恢复
热备份	备份时数据库仍然可用	如果热备份出错，所得结果不能用于时间点的恢复
	可以实现实时/秒级恢复	不能出错，否则可能会引起数据库无法恢复

	可对几乎所有数据库实体做恢复，速度快	维护比较困难
--	--------------------	--------

系统安全性和保密性设计

3、原则类

系统规划

软件架构设计

设计模式

系统设计

高质量数据流图设计原则：

1、 复杂性最小化原则：DFD 分层结构就是把信息划分为晓得且相对独立的一大批子集例子，这样就可以单独考查每一个 DFD；

2、 接口最小化原则：在设计模式时，模型中各个元素之间的接口数或者连接数最小化；

3、 数据流一致性原则：过程与过程分解数据流一致，有数据流出就有数据流入；数据流入需要有相应的数据加工；

4、其他

1、 电子政务包括：政务信息查询；公共政务办公；政务办公自动化三个应用领域