

2022 年下半年系统架构设计师第二期模考试卷（论文）

1、

论企业集成平台的理解与应用

企业信息集成是解决“孤岛”问题的需要，技术发展的同时也推动了集成架构等相关的研究。企业集成平台的核心是企业集成架构，包括信息、过程、应用集成的架构。

请围绕“论企业集成平台的理解与应用”论题，依次从以下三个方面进行论述。

- 1、概要叙述你所参与管理或开发的软件项目，以及你在其中所承担的主要工作。
- 2、给出至少 4 种企业集成平台应具有的基本功能，并对这 4 种功能内涵进行简述。
- 3、阐述你在软件开发的过程中都遇到了哪些实际问题及解决方法。

问题内容：

试题答案：

试题解析：

集成平台是支持企业集成的支撑环境，包括硬件、软件、软件工具和系统，通过集成各种企业应用软件形成企业集成系统。由于硬件环境和应用程序的多样性，企业信息系统的功能和环境都非常复杂，因此，为了能够较好地满足企业的应用需求，作为企业集成系统支持环境的集成平台，其基本功能主要有：

(1) 通信服务

它提供分布环境下透明的同步 / 异步通信服务功能，使用户和应用程序无需关心具体的操作系统和应用程序所处的网络物理位置，而以透明的函数调用或对象服务方式完成它们所需的通信服务要求。

(2) 信息集成服务

它为应用提供透明的信息访问服务，通过实现异种数据库系统之间数据的交换、互操作、分布数据管理和共享信息模型定义（或共享信息数据库的建立），使集成平台上运行的应用、服务或用户端能够以一致的语义和接口实现对数据（数据库、数据文件、应用交互信息）的访问与控制。

(3) 应用集成服务

它通过高层应用编程接口来实现对相应应用程序的访问，这些高层应用编程接口包含在不同的适配器或代理中，它们被用来连接不同的应用程序。这些接口以函数或对象服务的方式向平台的组件模型提供信息，使用户在无需对原有系统进行修改（不会影响原有系统的功能）的情况下，只要在原有系统的基础上加上相应的访问接口就可以将现有的、用不同的技术实现的系统互联起来、通过为应用提供数据交换和访问操作，使各种不同的系统能够相互协作。

(4) 二次开发工具

二次开发工具是集成平台提供的一组帮助用户开发特定应用程序（如实现数据转换的适配器或应用封装服务等）的支持工具，其目的是简化用户在企业集成平台实施过程中（特定应用程序接口）的开发工作。

(5) 平台运行管理工具

它是企业集成平台的运行管理和控制模块，负责企业集成平台系统的静态和动态配置、集成平台应用运行管理和维护、事件管理和出错管理等。通过命名服务、目录服务、平台的动态静态配置，以及其中的关键数据的定期备份等功能来维护整个服务平台的系统配置及稳定运行。

3、论基于架构的软件设计方法（ABSD）的应用

基于架构的软件设计（architecture-based software design, ABSD）方法是由架构驱动的。使用该方法，设计活动可以从项目总体功能框架明确就开始，这意味着需求抽取和分析还没有完成（甚至远远没有完成），就开始了软件设计。设计活动的开始并不意味着需求抽取和分析活动就可以终止，而是应该与设计活动并行。特别是在不可能预先决定所有需求时，例如产品线系统或长期运行的系统，快速开始设计是至关重要的。

问题内容：请围绕“基于架构的软件设计方法（ABSD）的应用”论题，依次对以下三个问题进行论述。

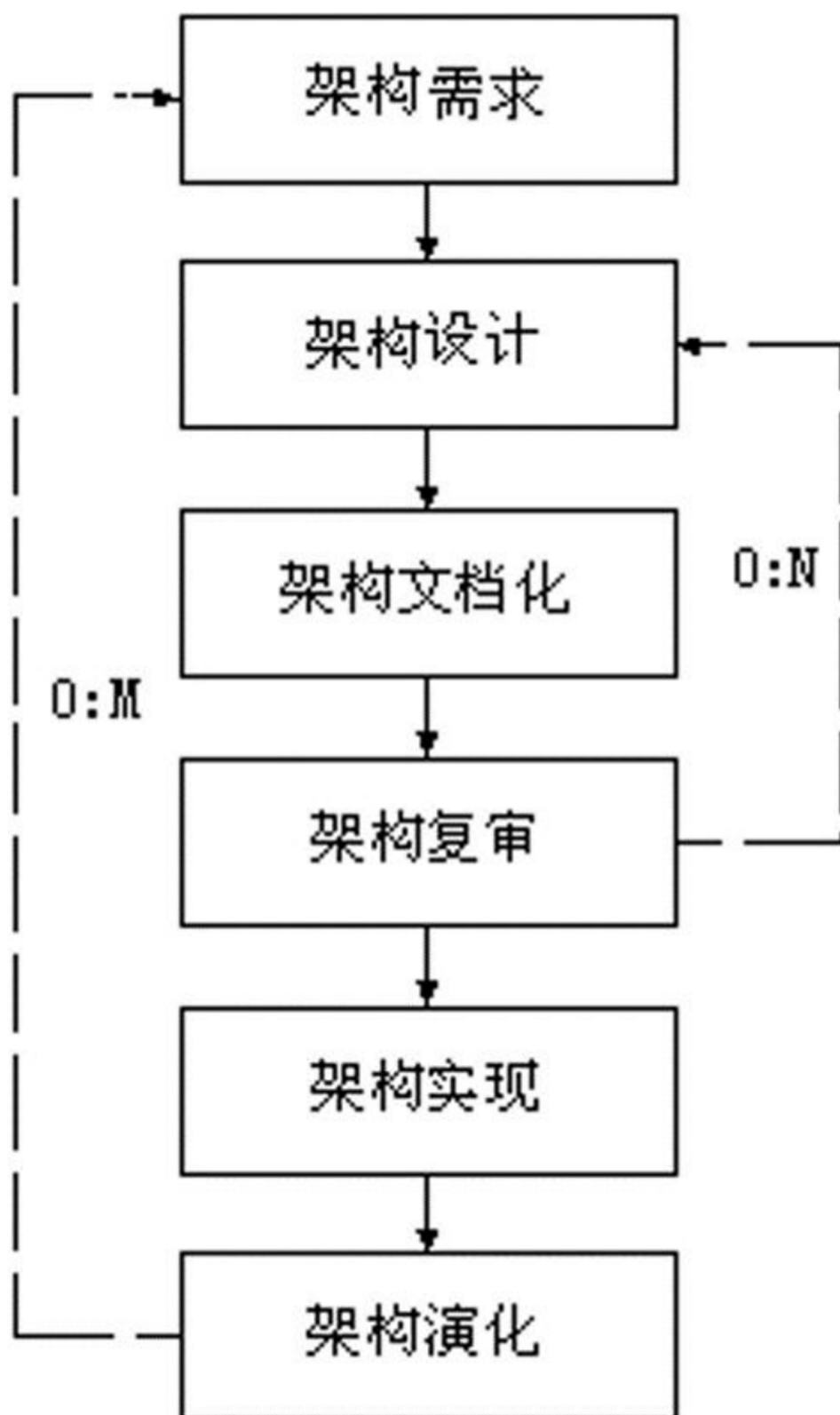
1. 叙述你参与的软件开发项目概要，以及你所担任的工作。
2. 简要叙述基于架构的开发模型把整个软件过程划分为哪些子过程，并详细论述你是如何利用基于架构的软件开发方法进行开发的。
3. 对于你所采用的手段，各举出一点你认为有效果的方面和无效果的方面，简要叙述你对这方面的评价如何？

试题答案：

试题解析：传统的软件开发过程可以划分为从概念直到实现的若干个阶段，包括问题定义、需求分析、软件设计、软件实现及软件测试等。如果采用传统的软件开发模型，软件架构的建立应位于需求分析之后，概要设计之前。

传统软件开发模型存在开发效率不高，不能很好地支持软件重用等缺点。基于架构的开发模型把整个基于架构的软件过程划分为架构需求、设计、文档化、复审、实现、演化等六个子过程，

如图。



3、

论企业应用系统的数据持久层架构设计

数据持久层（Data Persistence Layer）通常位于企业应用系统的业务逻辑层和数据源层之间，为整个项目提供一个高层、统一、安全、并发的数据持久机制，完成对各种数据进行持久化的编程工作，并为系统业务逻辑层提供服务。它能够使程序员避免手工编写访问数据源的方法，使其专注于业务逻辑的开发，并且能够在不同项目中重用本框架，这大大简化了数据的增加、删除、修改、查询功能的开发过程，同时又不丧失多层结构的天然优势，继承延续应用系统架构的可伸缩性和可扩展性。当运用关系型数据库作为数据存储机制时，在业务层与数据源间加入数据持久层，能够解决对象与关系的“阻抗不匹配”问题，将对象的状态持久化存储到关系型数据库中。

问题内容：

请围绕“企业应用系统的数据持久层架构设计”论题，依次从以下三方面进行论述。

1. 概要叙述你参与分析和设计的企业应用系统开发项目以及你所担任的主要工作。
2. 分析在企业应用系统的数据持久层架构设计中有哪些数据访问模式，并详细阐述每种数据访问模式的主要内容。
3. 数据持久层架构设计的好坏决定着应用程序性能的优劣，请结合实际说明在数据持久层架构设计中需要考虑哪些问题。

试题答案：

试题解析：

一、简要描述所参与分析和设计的企业应用系统开发项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、分析在企业应用系统的数据持久层架构设计中有哪些数据访问模式，并详细阐述每种数据访问模式的主要内容。

企业应用系统的数据持久层架构设计中主要有五种数据访问模式：

（1）在线访问（Online Access）。OA 是最基本的数据访问模式，也是在实际开发过程中最常采用的。这种数据访问模式会占用一个数据库连接，读取数据，每个数据库操作都会通过这个连接不断地与后台的数据源进行交互。

（2）数据访问对象（Data Access Object）。DAO 模式是标准的 J2EE 设计模式之一，开发人员常常用这种模式将底层数据访问操作与高层业务逻辑分离开。一个典型的 DAO 实现通常包括：一个 DAO 工程类；一个 DAO 接口；一个实现了 DAO 接口的具体类，包含访问特殊数据源中数据的逻辑；数据传输对象。

（3）数据传输对象（Data Transfer Object）。DTO 是经典 EJB 设计模式之一，它本身是一组对象或者数据的容器，需要跨越不同的进程或者网络的边界来传输数据。对象本身应该不包含具体的业务逻辑，并且通常这些对象内部只能进行一些诸如内部一致性检查和基本验证之类的方法，而且这些方法最好不要再调用其他的对象行为。在具体实现 DTO 时，可以使用编程语言内置的集合对象，也可以通过创建自定义类来实现 DTO 对象。

（4）离线数据模型（Off-line Data Model）。ODM 以数据为中心，数据从数据源获取之后，

将按照某种预定义的结构存放在系统中，成为应用的中心。离线方式可以使得对数据的各种操作独立于各种与后台数据源之间的连接或者事务；通过与 XML 集成数据可以方便地与 XML 格式的文档之间相互转换；独立于数据源，ODM 定义了数据的存储结构和规则。

（5）对象关系映射（Object Relational Mapping）。ORM 是随着面向对象软件开发方法发展而产生的，面向对象开发方法是主流的开发方法，关系型数据库是企业级应用环境中永久存放数据的主流数据存储系统。对象和关系数据是业务实体的两种表现形式，业务实体在内存中表现为对象，在数据库中表现为关系数据。ORM 一般以中间件的形式存在，能够帮助将应用程序中的数据转换成关系型数据库中的记录；或者将关系数据库中的记录转换成应用程序中便于操作的对象。

三、数据持久层架构设计的好坏决定着应用程序性能的优劣，无论在 C/S，还是在 B/S 结构中，持久层在处理数据的同时，对服务器锁的类型和持续时间、输入输出活动量以及处理器负荷等产生主要影响，并由此影响应用程序的总体性能。在持久层设计阶段需要考虑的问题包括：网络流量问题；返回结果集的问题；查询或锁定超时的问题；应用程序开发工具的问题；使用游标的问题；应用层设计的问题等。

4、

论软件可靠性设计技术的应用

随着软件的日益普及，系统中软件成分不断增加，使得系统对软件的依赖越来越强。软件的可靠性对系统可靠性的影响越来越大。而实践证明，保障软件可靠性最有效、最经济、最重要的手段是在软件设计阶段采取措施进行可靠性控制，为此提出了软件可靠性设计的概念。

软件可靠性设计就是在常规的软件设计中，应用各种方法和技术，使软件设计在兼顾用户功能和性能需求的同时，全面满足软件的可靠性要求。软件可靠性设计应和软件的常规设计紧密结合，贯穿于软件设计过程的始终。

问题内容：

请围绕“软件可靠性设计技术的应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与管理和开发的软件项目以及你在其中所承担的主要工作。
2. 结合项目实际，论述你在项目开发过程中，进行软件可靠性设计时遵循的基本原则；论述你在该项目中所采用的具体可靠性设计技术。
3. 阐述你在具体的可靠性设计工作中，为了分析影响软件可靠性的主要因素，所采用的可靠性分析方法。

试题答案：

试题解析：

- 一、概要论述你参与管理和开发的信息系统项目以及你在其中所承担的主要工作。
- 二、结合项目实际，论述你在进行软件可靠性设计时遵循的基本原则，你所采用的具体可靠性

设计技术的基本内容。

可靠性设计需要遵循的原则有：

- 1、软件可靠性设计是软件设计的一部分，必须在软件的总体设计框架中使用，并且不能与其他设计原则相冲突。
- 2、软件可靠性设计在满足提高软件质量要求的前提下，以提高和保障软件可靠性为最终目标。
- 3、软件可靠性设计应确定软件的可靠性目标，不能无限扩大，并且排在功能、用户需求、开发费用之后考虑。

常见的可靠性设计技术有容错设计、检错设计、降低复杂度设计等技术。

容错设计技术：对于软件失效后果特别严重的场合，如飞机的飞行控制系统、空中交通管制系统等，采用容错设计技术。常见的容错设计技术有三种：恢复块设计、N 版本程序设计和冗余设计。

恢复块设计：选择一组软件操作作为容错设计单元，把普通的程序块变成恢复块。一个恢复块包含有若干个功能相同、设计差异的程序块文本，一个运行文本，多个备份文本，构成“动态冗余”，一旦运行文本出现故障，则用备份文本替换。软件容错的恢复块方法就是使软件包含有一系列恢复块。

N 版本程序设计：N 版本程序的核心是通过设计出多个模块或不同版本，对于相同初始条件和相同输入的操作结果，实现多数表决，防止其中某一软件模块/版本的故障提供错误的服务，以实现软件容错。

冗余设计：在一套完整的软件系统之外，设计一种不同路径、不同算法或不同实现方法的模块或系统作为备份，在出现故障时可以使用冗余的部分进行替换，从而维持软件系统的正常运行。缺点是费用和资源的消耗会有所增加。

检错技术：在软件系统中，无需在线容错的地方，或不能采用冗余设计技术的部分，如果对可靠性要求较高，故障有可能导致严重的后果时，一般采用检错技术，在软件出现故障后能及时发现并报警，其缺点是不能自动解决故障。

降低复杂度设计：软件复杂性与软件可靠性有着密切的关系，是产生软件缺陷的重要根源。在设计时考虑降低软件的复杂性，是提高软件可靠性的有效方法。降低复杂度设计的思想是在保证实现软件功能的基础上，简化软件结构，缩短程序代码，优化软件数据流向，降低软件复杂度，从而提高软件可靠性。

（结合实际工作，具体解释遵循的原则和采用的一种或多种可靠性设计技术）

三、阐述你在具体的可靠性设计工作中，为了分析影响软件可靠性的主要因素，所采用的可靠性分析方法。

在软件可靠性设计之前和软件可靠性设计过程中，都需要采用软件可靠性分析和预测方法，来确定当前系统中的主要可靠性因素和目标。常见的软件可靠性分析方法包括故障树分析方法、失效模式与效应分析方法等。

故障树分析方法：一种自顶向下的软件可靠性分析方法，即从软件系统不希望发生的事件（顶事件），特别是对人员和设备的安全及可靠性产生重大影响的事件开始，向下逐步追查导致顶事件发生的原因，直至基本事件（底事件），从而确定软件故障原因的各种可能组合方式和（或）发生概率。基本的步骤是软件故障树的建立、定性分析和定量分析。

失效模式与效应分析方法：在软件开发阶段的早期，通过识别软件失效模式，分析造成的后果，研究分析各种失效模式产生的原因，寻找消除和减少其有害后果的方法，以便尽早发现潜在的问题，并采取相应的措施，从而提高软件的可靠性和安全性。SFMEA 的分析对象可以是开发早期阶段的高层次的子系统、部件，也可以是详细设计阶段的单元、模块。对于不同的分析对象，其软件失效模式是不同的，采用的 SFMEA 分析方法也不同，前者采用系统级分析方法（system FMEA），后者为详细级分析方法（detailed FMEA）。其基本的步骤是系统定义、软

件失效模式分析、软件失效原因分析、软件失效影响分析、改进措施分析。
(结合实际工作，具体阐述自己所采用的一种或多种可靠性分析方法)

更多备考资料和学习福利，可扫码添加希赛萧萧老师，申请入群

