

全国计算机技术与软件专业资格（水平）考试

系统架构设计师

文老师系统架构设计师课程内部辅导资料

论文写作总复习资料

淘宝搜索：文老师系统架构设计师



淘宝店铺购课



微信公众号



抖音号

或在浏览器输入下面链接，登录文老师官方淘宝店（包含系分、架构、系规、高项、中项、软设、网工、网规、数据库、评测、信安等全类目软考辅导）：

wenlaoshi.taobao.com

一	论文整体分析.....	3
1	复习说明.....	3
2	历年真题.....	5
3	写作原则.....	6
4	常见问题.....	6
5	评分标准.....	7
6	论文准备顺序.....	8
二	搭建自己的万能模板.....	8
1.	选择合适的项目.....	9
2.	提前准备论文摘要.....	9
3.	提前准备项目背景.....	10
4.	正文写作.....	10
5.	提前准备结尾.....	10
6.	万能模板（重要）.....	10
三	正文素材及范文.....	12
1	第二版下篇具体应用架构.....	12
2	论面向服务架构设计及其应用.....	18
3	论软件系统建模方法及其应用.....	22
4	论软件系统架构风格.....	24
5	论软件架构评估.....	27
6	论高可靠性系统中软件容错技术的应用.....	31
7	论信息系统的安全架构设计.....	35
8	论企业应用集成.....	38
9	基于构件的软件开发.....	41
10	论企业集成平台的技术与应用.....	45
11	论软件设计模式及其应用.....	49
四	历年论文写作题目.....	53
	2022 年下半年试题 1.....	53
	2022 年下半年试题 2.....	53
	2022 年下半年试题 3.....	54
	2022 年下半年试题 4.....	54
	2021 年下半年试题 1.....	55
	2021 年下半年试题 2.....	56
	2021 年下半年试题 3.....	58
	2021 年下半年试题 4.....	59
	2020 年下半年试题 1.....	61
	2020 年下半年试题 2.....	62
	2020 年下半年试题 3.....	63
	2020 年下半年试题 4.....	65
	2019 年下半年试题 1.....	66
	2019 年下半年试题 2.....	67
	2019 年下半年试题 3.....	67

2019 年下半年试题 4	69
2018 年下半年试题 1	70
2018 年下半年试题 2	70
2018 年下半年试题 3	71
2018 年下半年试题 4	71
2017 年下半年试题 1	71
2017 年下半年试题 2	72
2017 年下半年试题 3	72
2017 年下半年试题 4	73
2016 年下半年试题 1	73
2016 年下半年试题 2	73
2016 年下半年试题 3	74
2016 年下半年试题 4	74
2015 年下半年试题 1	75
2015 年下半年试题 2	75
2015 年下半年试题 3	76
2015 年下半年试题 4	76
2014 年下半年试题 1	76
2014 年下半年试题 2	77
2014 年下半年试题 3	77
2014 年下半年试题 4	78
2013 年下半年试题 1	78
2013 年下半年试题 2	79
2013 年下半年试题 3	79
2013 年下半年试题 4	79
2012 年下半年试题 1	80
2012 年下半年试题 2	80
2012 年下半年试题 3	81
2012 年下半年试题 4	81
2011 年下半年试题 1	82
2011 年下半年试题 2	82
2011 年下半年试题 3	82
2011 年下半年试题 4	83
2010 年下半年试题 1	83
2010 年下半年试题 2	84
2010 年下半年试题 3	84
2010 年下半年试题 4	84

一 论文整体分析

1 复习说明

论文写作概述：

时间：共 2 个小时，从 15:30-17:30。

题目：四选一，必有一题考察软件架构，每个题目下会有概述和子题目，要认真看。

考察形式：笔试，手写约 2500 字。

题目类型：

1、软件架构，最喜欢考察架构风格、面向服务的架构、架构评估，会用到质量属性，其他如 DSSA 和 ABSD 的概念也要了解。

2、系统开发这块，会考需求分析、需求获取、系统建模、设计模式等知识。

3、系统可靠性、安全性、容错技术等。

4、信息系统开发方法、开发模型、生命周期模型、企业应用集成技术。

5、其他：项目管理、数据库等。

论文复习方法：

1、没有项目经验的学员不要担心，听论文专题视频课程，掌握论文写作技巧，参考范文改写，论文完全有可能通过。

2、写论文的前提是至少完成了一遍所有基础视频课程的学习，建议在考前一个半月开始准备论文，确保完成以下任务：

论文写作专题视频课程的学习；

确定好自己要写的项目，并确定论文模板；

依据确定的项目和模板，按照老师给定的顺序准备论文，建议每周写一篇论文，至少完成 5 篇论文，并且能够手写默写出来。

第二版教材的第 20 章：系统架构设计师论文写作要点

这个章节挺有意思的，教材里提供的关于论文写作的注意事项、要点分析、分段写法、摘要模板等内容，都跟文老师一直以来总结的论文总复习资料内容大同小异，是比较适合备考的，不同于之前 19 章里的长篇大论的理论堆叠。因此，关于论文，大家按本复习资料及每年最新直播讲解复习即可。

本资料使用说明：

本资料为系统架构设计师论文写作总复习资料，包含注意事项、万能模板、论文素材、范文、真题等全部内容，请大家务必结合视频课程认真学习。具体介绍如下：

第一章是论文整体分析，大家认真浏览一遍，心中有数即可。

2 历年真题

这里仅汇总大题目，供整体分析，每年真题的子题目，大家也有必要了解，都在第四章。

年份	试题一	试题二	试题三	试题四
2022	论基于构件的软件开发方法及其应用	论软件维护方法及其应用	论区块链技术及应用	论湖仓一体架构及其应用
2021	论面向方面的编程技术及其应用	论系统安全架构设计及其应用	论企业集成平台的理解与应用	论微服务架构及其应用
2020	论企业集成架构设计及应用	论软件测试中缺陷管理及其应用	论云原生架构及其应用	论数据分片技术及其应用
2019	论软件设计方法及其应用	论软件系统架构评估及其应用	论数据湖技术及其应用	论负载均衡技术在web 系统中的应用
2018	论软件开发过程 RUP 及其应用	论软件体系结构的演化	论面向服务架构设计及其应用	论 NoSQL 数据库技术及其应用
2017	论软件系统建模方法及其应用	论软件架构风格	论无服务器架构及其应用	论软件质量保证及其应用
2016	论软件系统架构评估	论软件设计模式及其应用	论数据访问层设计技术及其应用	论微服务架构及其应用
2015	论应用服务器基础软	论软件系统架构风格	论面向服务的架构及	论企业集成平台的技

	件		其应用	术与应用
2014	论软件需求管理	论非功能性需求对企业应用架构设计的影响	论软件的可靠性设计	论网络安全体系设计
2013	论软件架构建模技术与应用	论企业应用系统的分层架构风格	论软件可靠性设计技术的应用	论分布式存储系统架构设计
2012	论企业信息化规划的实施与应用	论决策支持系统的开发与应用	论企业应用系统的数据持久层架构设计	论基于架构的软件设计方法及应用
2011	论模型驱动架构在系统开发中的应用	论企业集成平台的架构设计	论企业架构管理及应用	论软件需求获取技术及应用
2010	论软件的静态演化和动态演化及其应用	论数据挖掘技术的应用	论大规模分布式系统缓存设计策略	论软件可靠性评价

3 写作原则

不要猜题，要复用构件（摘要+项目背景+结尾），不要整篇复用。

不要抄范文，只能改范文，可以选择适合自己的范文进行修改，改成自己的论文。

练习时写文章前，提前准备好自己要写的项目，必须是近三年内的中大型商业项目。

做好准备工作，练习论文主题时发现不会的知识点，一定要全部背会。

要勇于迈出第一步，万事开头难，写完第一篇，就不难了，第一篇论文不限时间。

写正文时，不要生硬的回答问题，要根据问题要点组合成一篇通顺的文章。

字数一般在 300+2200 左右，写到最后一页即可，字迹一定要完整，一笔一划。

从系统架构设计师的角度看项目，技术细节适中，书本理论要与实际的项目联系起来。

不要全局都列举一二三到 123，可以局部段落列举 123，论点分开论述，没必要都写编号。

论述要求：不能口语化、文档段落不宜太长。

换个角度看论文，以论文写作技巧视频为依据，对自己的论文进行自评。

4 常见问题

问题 1：论文字数范围，各部分字数如何分配？

答：首先，答题卡最大字数是 330+2750，肯定不能超过答题卡；其次，建议摘要 300 字左右，正文 2000-2700 字都可以，建议 2200 字，写到答题卡最后一页即可；最后字数分配建议摘要 300 字+项目背景 500 字+主体 1200 字+结尾 500 字。

问题 2：我没有做过系统架构设计师怎么办？

答：如果你是专业人员，虽然不是系统架构设计师，但跟过项目，那么就从系统架构设计师的角度去写熟悉的项目；如果你是完全零基础，行外人，就参考范文，找一个觉得合适的项目背景改写。

问题 3：我做的是公司内部的项目，没有客户怎么办？

答：不要在论文里写这是内部项目，当成普通项目写，如某年某月，公司要做什么项目，任命我为系统架构设计师即可，最后收尾仍然写客户一致满意，按照一般项目来写。

问题 4：是否需要写论文题目？

答：答题卡上不能写论文题目，直接在答题卡那里将你选择的题号涂黑即可。

问题 5：数字、英文字母、标点符号该怎么写？

答：遵循小学生作文写作原则，两个数字占一个格子；字母按自然大小连起来写，看着不拥挤，使其自然占几个格子；标点符号要占一个格子，并且每行开头第一个格子不能有标点符号。以上这些都算字数，因为论文答题纸是标识格子的数量，没人会去数你到底写了多少汉字，有多少是空格。

问题 6：项目背景如何构造才比较合理？

关于项目时间：建议离考试时间五年以内，持续时间 8 个月以上，并且已完成的项目；如果说项目真的是比较久远的，或者持续时间较短的，但是规模是足够的，请自己灵活修改时间。

关于项目级别：建议是省市级、集团级，不要是县级、社区级等，如果你做的项目规模是符合要求的，真的是县级的真实项目，那没有问题。

是否要写真实的项目名称：不需要，也不能写，只能写某地级市，某医院，某公司。

关于项目团队成员：可省略，为了保证真实性，也可以简单带过，例如有需求分析人员 3 人，开发人员 8 人，测试人员 5 人等等。

关于项目金额：要和项目时间还有项目团队成员对应，如果项目组一共有 20 人，按每人每月 2 万算，一个月需要 40 万，再乘以持续时间假设是 15 个月，那么项目金额在 600 万左右是合理的，但不要写整数，可以写 565 万，618 万等字眼。

特别注意：如果是自己编的项目，请不要构造太大的项目，项目金额最好在 500 万左右。

问题 7：如何记住论文？

答：一定要自己改写，即使参考了别人的范文，也不能直接 copy，自己动手写过的东西才能记住。文章采用统一的模板和结构，搭建好自己的模板之后不要改动，正文部分按论文题目要求去写，一定要回应子题目。另外，一定要自己动手写，多默写，并且在两个小时内默写完。

5 评分标准

下述情况的论文，需要适当扣 5 分到 10 分：

没有写论文摘要、摘要过于简略、或者摘要中没有实质性内容的论文。

字迹比较潦草、其中有不少字难以辨认的论文

确实属于过分自我吹嘘或自我标榜、夸大其词的论文

内容有明显错误漏洞的，按同一类错误每一类扣一次分

内容仅属于大学生或研究生实习性质的项目、并且其实际应用水平相对较低的论文
下述情况，可考虑适当加分（可考虑加 5 分到 10 分）

有独特的见解或者有着很深入的体会、相对非常突出的论文

起点很高，确实符合当今信息系统发展的新趋势与新动向，并能加以应用的论文

内容详实、体会中肯、思路清晰、非常切合实际的很优秀的论文

项目难度很高，或者项目完成的质量优异，或者项目涉及国家重大信息系统工程且作者本人参加并发挥重要作用、并且能正确按照试题要求论述的论文

下述情况之一的论文，不能给予及格分数：

虚构情节、文章中有较严重的不真实的或者不可信的内容出现的论文

没有项目开发的实际经验、通篇都是浅层次纯理论的论文

所讨论的内容与方法过于陈旧、或者项目的水准非常低下的论文

内容不切题意，或者内容相对很空洞、基本上是泛泛而谈且没有较深入体会的论文

正文与摘要的篇幅过于短小的论文（如正文少于 1200 字）

文理很不通顺、错别字很多、条理与思路不清晰、字迹过于潦草等情况相对严重的论文

6 论文准备顺序

因为我们最终押题要到考前一周左右才能出来，如果到时候才准备论文会显得很仓促，同时也有学员想要提前准备论文，不知道该写什么题目。那么：

1、请各位学员直接按照第三章的论文标题顺序来依次准备论文，这里涵盖了历年常考的论文题目，并按重要程度和频率排序。

2、建议至少准备到第 7 篇，后面的论文，也建议浏览下范文和相关知识点，如果考到不至于什么都不会。

3、大家不要觉得 7 篇论文很多，当你有了自己的模板，写了三篇论文之后，后面写起来就很轻松了，万事开头难。

考前一周左右我们会发布论文押题，大家耐心等待，不要催，不过应该会有几个题目就是在第三章的论文题目里，所以大家提前准备并不会浪费时间。

二 搭建自己的万能模板

请大家在学完本章节后，按步骤一步步的搭建好自己的模板，后面也有万能模板供仿照。如果是零基础学员，不知道怎么选择项目，可以提前看第三章里提供的范文，从中选一个改改作为自己的项目。有自己真实项目经历的，尽量用自己的真实项目，如果自己项目规模不够，可以在真实项目基础上去扩展，因为范文的项目可以预见会被很多人采用，容易导致重复。

1. 选择合适的项目

选择中大型商业项目，一般金额在 200W 以上，研发周期在 8 个月以上的项目。

推荐项目

政府或大型信息系统项目：各国企、事业单位、军方、医院、银行、股份公司大企业的 ERP、OA 等；云计算、大数据等各种软件及信息系统。

不能选项目

小型企业项目：如进销存系统、图书管理系统、单机版系统。

尚未完成的项目：一定要已经完成并上线运行，并最好在三年内。

纯建网站项目：如建设企业或政府门户网站，只有静态网页链接介绍，没有后台大型应用的。

硬件项目：如综合布线、安防、视频会议等。

纯技术项目：如数据升级迁移、内部技术研究等。

建议：在推荐的范围内有限选择自己做的或熟悉的中大型项目，如果是零基础没有做过项目的学员，就参考后面的范文，用别人的项目来模仿。

2. 提前准备论文摘要

摘要自己参考范文或者下面的格式，根据自己选择的项目准备一个就可以了。建议逻辑上分两段（但是实际写作不要写两段，因为格子可能不够），第一段是通用的介绍项目背景，第二段是根据不同的论文题目发挥的，简单回应子题目并介绍论文结构。

注意：摘要最多只有 330 个格子，千万不能超，建议写 300 个字即可。

包含内容：项目名称、项目金额、项目历时、项目简介、我的责任；本文讨论主题概括（具体可以参考后面的万能模板）。

论文摘要格式：

（1）本文讨论.....系统项目的.....（指的是项目主题，例如进度管理等），该系统是由某单位建设的，投资多少万，系统是用来做什么的（项目背景，简单功能等）。在本文中，首先讨论了.....（过程、方法、措施），最后.....(主要是不足之处/如何改进/特色之处/发展趋势等)。在本项目的开发过程中，我主要担任了.....（在本项目中的角色）。

（2）根据.....需求（项目背景），我所在的.....组织了.....项目的开发。该项目.....（项目背景、简单功能介绍）。在该项目中，我担任了.....（角色）。我通过采取.....(过程、方法、措施等)，使项目圆满成功，得到了用户的一致好评。但通过项目的建设，我发现.....(主要是不足之处/如何改进/特色之处/发展趋势等)

（3）...年...月,我参加了.....项目的开发,担任.....(角色)。该项目投资多少，建设工期是多少，该项目是为了（项目背景、功能介绍）。本文结合作者的实践，以.....项目为例，讨论.....（论文主题），包括.....（过程、方法、措施）。

(4)是.....(戴帽子，讲述论文主题的重要性，比如进度的重要性)。本文结合作者的实践，以.....项目为例，讨论.....(论文主题)，包括.....(过程、方法、措施)。在本项目的开发过程中，我担任了.....(角色)。

3. 提前准备项目背景

项目背景及过渡部分建议 500-600 字左右，不能超过太多，需要提前准备好，尽量通用化，不要和论文主题相关，这样考试时候无论什么论文主题都可以直接默写，只需要在最后写一段过渡语句，过渡到下一个论点。

包括内容：项目开发的原因、你的岗位职责、项目开发周期及规模、项目功能组成介绍、项目技术(可省略)。

具体可以参考后面的万能模板来写。

4. 正文写作

正文应该按照论文题目和子题目的要求来写作，并且一定要回应论文子题目。

正文需要写 1200 字左右。正文部分不在万能模板里，需要根据不同的题目来准备，详细见第三章内容，本章可以先略过，先准备好自己的万能模板。

5. 提前准备结尾

结尾是个非常有意思的部分，从其本身意义来说，是让你总结项目收获和不足的，然后另一方面，还是你整体补救论文的最后一步。如果你到最后发现字数不够，结尾就需要多写一些，如果你发现字数多了，结尾就要少写一点，如果你觉得前面写的很差，结尾就一定要重视。

结尾可以写 400-600 字左右，是对整体论文的总概。

包括内容：项目上线及运行效果、客户评价、项目收获、项目不足和解决思路。

具体可以参考后面的万能模板来写。

6. 万能模板 (重要)

按照上面提到的步骤，我们可以总结出如下的套路模板，大家可以参考，还要用自己的项目来模仿，其中标红部分是完全复用的，任何题目都不用修改。

摘要模板 (时间+项目+项目简介+投入+历时+成功交付客户好评+结合具体题目说明本文结构)：

2021 年 3 月，我参与了某航天研究所某型号卫星的全数字仿真验证平台项目的建设，并担任系统架构设计师，负责系统架构设计工作。该系统包括虚拟目标机仿真、动力学模型仿真、同步时序控

制三大功能模块，能够模拟卫星在太空中运行所需的所有硬件及外部力学环境，从而可以在虚拟平台中充分测试卫星软件的功能及性能以提高卫星软件的可靠性。该项目总投入 565 万元人民币，历时 15 个月，于 2022 年 06 月正式交付运行至今，受到了客户的一致好评。本文结合笔者的实际工作经验就该项目的……（根据不同论文题目去简要概括本文内容）。

项目背景模板（为什么要做这个项目+项目功能和技术介绍+回应子题目并过渡到主体）：

在航天卫星的研制过程中，一颗航天卫星只有一套配套的硬件设备，无法满足一个开发团队的测试需求，同时因为航天卫星硬件设备造价十分昂贵，测试人员也无法进行一些非常规的极限测试，以防止损坏硬件设备，以上种种，将会造成对航天卫星软件的测试不充分和不彻底，有可能导致卫星研制失败。为了防止这种情况的发生，针对某重要型号的航天卫星的研制，该航天研究所领导决定使用技改经费投资建设一套全数字仿真验证平台，以纯软件的方式模拟卫星在太空中运行所需的全部硬件及外部力学环境，在虚拟平台中运行卫星软件和动力学模型并进行详细彻底的测试。

我所在的公司成功中标该项目，并于 2021 年 03 正式启动该项目的建设工作，我被任命为该项目的系统架构设计师，负责系统架构设计工作。该项目总投入 565 万元人民币，建设周期从 2021 年 3 月 10 日至 2022 年 6 月 30 日止，历时 15 个月。系统采用两台联想 Think Station P510 搭载软件的运行，考虑到对于性能及执行效率方面的要求，使用 C 语言模拟 CPU 指令集及外部设备驱动，使用 labview 图形化语言搭建地面遥测遥控终端界面。项目的主要建设内容包括三大模块：一、虚拟目标机仿真，完成对卫星软件运行所需的全部硬件环境的模拟，包括内核模拟、片内外设模拟、板级外设模拟等子系统；二、动力学模型仿真，完成对动力学模型运行所需的环境的模拟，包括动力学模型运行环境模拟、故障注入模拟、动力学同步数据控制与显示等子系统；三、同步时序控制，控制卫星软件及动力学模型的启停以及二者之间的时序同步，包括总控台监控、同步时序控制、超实时运行控制等子系统。

笔者所在的公司虽然在其他型号航天卫星的全数字化仿真验证平台项目上取得成功，但由于卫星型号及用途不同，所涉及的硬件及使用标准也不同，对应的动力学模型也完全不同，而且由于严格的保密性要求，团队成员无法获取全部的卫星型号资料，这无疑加大了项目开发的难度和风险。于是笔者决定在……（过渡段，可以自行参考论文来写，需要回应子题目并引出正文）。

项目总结模板（强调项目顺利交付运行反馈好+自己的收获或者不足之处）：

经过近 15 个月的项目开发，该型号航天卫星的全数字仿真验证系统顺利投入使用，协助客户对卫星软件进行全面的功能和性能上的测试，运行至今客户反馈良好。该系统由于保密性高，性能要求高，技术实现难度高，项目建设周期长等原因，建设过程困难重重。但由于笔者及项目团队成员十分重视项目的……（回应具体论文题目），最终保证了该项目按质按量顺利交付。

当然，在本项目中，还有一些不足之处，比如：……（自己去想一些小问题，切忌，别出现什么大问题），不过，经过我后期的纠偏，并没有对项目产生什么影响。在后续的学习和工作中，我将不断的充电学习，和同行进行交流，提升自己的专业技术水平，更好的完成系统架构设计的工作。

三 正文素材及范文

在前面，学员应该已经将自己的万能论文模板准备好了，明确了自己要写的项目和背景，接下来，就是该如何去填充正文部分了，正文部分需要写 1300 字左右，要完成两件事：

1、按照论文题目写相关内容。这一块是自己要提前准备的重点，并且一定要针对自己的项目构造几个真实的例子，不能只列举理论。

2、回应子题目。这一块，一般是写过程，有可能会问其他的，需要针对性点题，但是不会影响大局，可以看看历年真题的子题目是怎么问的，一般可以直接按子题目顺序写论文正文。

本章针对常考的论文题目，给出了思路和范文，学员在学习本章内容的时候，一定要重点关注正文部分的写作，其他项目背景等模板部分用自己的万能模板。

特别注意：

1、范文仅供参考，请勿照抄，一定要自己改写。

2、范文可能会出现字数超标，某些部分与我们课程讲的不太一样，这都是正常的，大家重点要掌握写作的思路，然后自己写的规范点。

1 第二版下篇具体应用架构

纵观历年真题考察情况，可以发现，近几年关于纯理论的题目是越来越少了，如架构风格、架构评估这些是 19 年及之前常考的，而现在常考的是具体的应用架构，如云原生架构、微服务架构、企业集成架构、系统安全架构等，在第二版教材没有更新之前，这些内容文老师称之为新技术超纲部分，是我们难以把握的。

然而，在第二版教材更新之后，大家看到下篇的章节目录，会发现，原来这些内容都更新了，让我们来看看第二版下篇内容，分别是：信息系统架构、层次式架构、云原生架构、面向服务架构、嵌入式系统架构、通信系统架构、安全架构、大数据架构。

一共八章，结合架构本身科目特性以及历年真题考察情况，**文老师预测 2023 年 11 月论文考试，很有可能考到的具体架构是：层次式架构、面向服务架构、大数据架构。**

这些具体的架构，没有完全对应的范文，但大家可以多看一些对于不同架构描述的范文主体，结合案例专题里的内容来参考修改，同时直播也会重点讲解本块内容，以下提供一些具体架构范文，供参考：

层次式架构范文

摘要：我所在的单位是国内主要的商业银行之一，作为单位的主要技术骨干，2019 年 1 月，我主持了远期结售汇系统的开发，该系统是我行综合业务系统的一个子系统，由于银行系统对安全性，可靠性，可用性和响应速度要求很高，我选择了三层 C/S 结构作为该系统的软件体系结构，在详细的设计三层结构的过程中，我采用了字符终端为表示层，CICS TRANSACTION SERVER 为中间层，DB2 UDB 7.1 为数据库层，并采用了 CICS SWITCH 组，并行批量的办法来解决设计中遇到的问题，保证了远期

结售汇系统按计划完成并顺利投产，我设计的软件三层结构得到了同事和领导的一致认同和称赞。但是，我也看到在三层结构设计中存在一些不足之处：比如中间层的负载均衡算法过于简单，容易造成系统负荷不均衡，并行批量设计不够严谨，容易造成资源冲突等。

我所在的单位是国内主要的商业银行之一。众所周知，银行的业务存在一个“二八定理”：即银行的百分之八十的利润是由百分之二十的客户所创造。为了更好地服务大客户，适应我国对外贸易的蓬勃发展态势，促进我国对外贸易的发展，2019年1月，我行开展了远期结售汇业务。所谓的远期结售汇就是企业在取得中国外汇管理局的批准后，根据对外贸易的合同等凭证与银行制定合约，银行根据制定合约当天的外汇汇率，通过远期汇率公式，计算出交割当天的外汇汇率，并在那天以该汇率进行成交的外汇买卖业务。远期结售汇系统是我行综合业务系统的一个子系统，它主要包括了联机部分、批量部分、清算部分和通兑部分，具有协议管理、合约管理、报价管理、外汇敞口管理、帐务管理、数据拆分管理、报表管理、业务缩微和事后监督等功能。我作为单位的主要技术骨干之一，主持并参与了远期结售汇系统的项目计划、需求分析、设计、编码和测试阶段的工作。

由于银行系统对安全性，可靠性，可用性和响应速度要求很高，我选择了三层C/S结构作为该系统的软件体系结构，下面，我将分层次详细介绍三层C/S软件体系结构的设计过程。

1、表示层为字符终端。我行以前一直使用IBM的VISUALGEN 2.0附带的图形用户终端来开发终端程序，但在使用的过程中，分行的业务人员反映响应速度比较慢，特别是业务量比较大的时候，速度更是难以忍受。为此，我行最近自行开发了一套字符终端CITE，它采用VISUAL BASIC作为开发语言，具有响应速度快，交互能力强，易学，编码快和功能强大的特点，在权衡了两者的优点和缺点之后，我决定选择字符终端CITE作为表示层。

2、中间层为CICS TRANSATION SERVER(CTS)。首先，我行与IBM公司一直保持着良好的合作关系，而我行的大部分技术和设备都采用了IBM公司的产品，其中包括了大型机，由于CICS在IBM的大型机上得到了广泛的应用，并在我行取得了很大的成功，为了保证与原来系统的兼容和互用性，我采用了IBM的CTS作为中间层，连接表示层和数据库层，简化系统的设计，使开发人员可以专注于表示逻辑和业务逻辑的开发工作，缩短了开发周期，减少开发费用和维护费用，提高了开发的成功率；其次，对于中间层的业务逻辑，我采用了我行一直使用的VISUALAGE FOR JAVA作为开发平台，它具有简单易用的特点，特别适合开发业务逻辑，可以使开发人员快速而准确地开发出业务逻辑，确保了远期结售汇系统的顺利完成；最后，由于采用了CTS，确保了系统的开放性和互操作性，保证了与我行原来的联机系统和其他系统的兼容，保护了我行的原有投资。

3、数据层为DB2 UDB7.1。由于DB2在大型事务处理系统中表现出色，我行一直使用DB2作为事务处理的数据库，并取得了很大的成功，在DB2数据库的使用方面积累了自己独到的经验和大量的人才，为了延续技术的连续性和保护原有投资，我选择了DB2 UDB7.1作为数据层。

但是，在设计的过程中我也遇到了一些困难，我主要采取了以下的办法来解决：

1、CICS SWITCH 组。众所周知，银行系统对于安全性，可靠性，可用性和响应速度要求很高，特别是我行最近进行了数据集中，全国只设两个数据中心，分别在XX和YY两个地方，这样对以上的要求就更高了，为了保障我行的安全生产，我采用了CTS SWITCH组技术，所谓的CICS SWITCH组，

就是一组相同的 CTS，每个 CTS 上都有相同的业务逻辑，共同作为中间层，消除了单点故障，确保了系统的高度可用性。为了简化系统的设计和缩短通讯时间，我采用了简单的负载均衡算法，比如这次分配给第 N 个 CTS，下次则分配给第 N+1 个 CTS，当到了最后一个，就从第一个开始；为了更好地实现容错，我采用了当第 N 个 CTS 失效的时候，把它正在处理的业务转到第 N+1 个上面继续处理，这样大大增加了系统的可用性，可以为客户提供更好的服务；此外，我还采用了数据库连接池的技术，大大缩短了数据库处理速度，提高了系统运行速度。

2、并行批量。银行系统每天都要处理大量的数据，为了确保白天的业务能顺利进行，有一部分的帐务处理，比如一部分内部户帐务处理，或者代理收费业务和总帐与分户帐核对等功能就要到晚上批量地去处理，但是，这部分数据在数据集中之后就显得更加庞大，我行以前采用串行提交批量作业的办法，远远不能适应数据中心亿万级的数据处理要求，在与其他技术骨干讨论之后，并经过充分的论证和试验，我决定采用了并行批量的技术，所谓的并行批量，就是在利用 IBM 的 OPC (Tivoli Operations, Planning and Control) 技术，把批量作业按时间和业务处理先后顺序由操作员统一提交的基础上，再利用 DB2 的 PARTITION 技术，把几个地区分到一个 PARTITION 里面分别处理，大大提高了银行系统的数据处理速度，确保了远期结售汇系统三层结构的先进性。在并行批量的设计过程中，我考虑到批量作业有可能因为网络错误或者资源冲突等原因而中断，这样在编写批量程序和作业的时候必须支持断点重提，以确保生产的顺利进行。

由于我软件三层结构设计得当，并采取了有效的措施去解决设计中遇到的问题，远期结售汇系统最后按照计划完成并顺利投产，不但保证了系统的开发性、开放性、可用性和互用性，取得了良好的社会效益和经济效益，而且我的软件三层结构设计得到了同事和领导的一致认同与称赞，为我行以后系统的开发打下了良好的基础。

在总结经验的同时，我也看到了我在软件三层结构设计中的不足之处：首先，负载算法过于简单，容易造成系统的负荷不均衡：由于每个业务的处理时间不一样，有的可能差距很远，简单的顺序加一负载分配算法就容易造成负载不均衡，但是如果专门设置一个分配器，则增加了一次网络通讯，使得系统的速度变慢，这样对响应速度要求很高的银行系统来说也是不可行的，于是我决定采用基于统计的分配算法，即在收到请求的时候，根据预先设定的权值，按概率，直接分配给 CTS。其次，由于批量作业顺序设计得不够严谨等各种原因，容易造成资源冲突：在远期结售汇系统运行了一段时间之后，数据中心的维护人员发现了，系统有的时候会出现资源冲突现象，在经过仔细的分析之后，我发现，由于每天各个业务的业务量大小不一样，顺序的两个作业之间访问同一个表的时候便会产生资源冲突，另外，在 OPC 作业运行的过程中，操作员提交的其他作业与这个时间的 OPC 作业产生也有可能产生资源冲突。对于第一种情况，可以在不影响业务的情况下调整作业顺序或者对于查询作业运用 DB2 的共享锁的技术，而第二种情况则要制定规范，规定在某时间断内不允许提交某些作业来解决。为了更好地开展系统分析工作，我将在以后的工作实践中不断地学习，提高自身素质和能力，为我国的软件事业贡献自己的微薄力量。

数据中心存储架构范文

摘要：笔者于 2020 年 8 月参与了某地级市市级机关电子政务系统的建设工作，该电子政务信息系统分为三个子模块，分别是办公自动化系统、政企业务信息查询系统、公共信息发布系统；笔者在该项目中担任项目经理和系统分析师一职，主要负责项目的日常刮泥和数据存储体系架构的设计工作。该电子政务系统由于数据量较大，对数据的安全性要求较高，因此笔者在经过充分的调研之后，决定采用集中式存储架构的数据中心建设方案，该方案从三个角度对数据中心展开建设工作，分别是数据存储的基本架构、以基本数据架构为基础的开放式数据调用接口、集中式数据存储的安全方案。其中数据存储的基本架构采用以中央数据库为基础的集中式方案，开放式数据调用接口采用自行开发的，以 JDBC 为基础的中间件机制，安全方案以数据容灾机制、冗余备份机制为主要的安全机制。该系统正式上线运行后，效果良好，基本实现了用户需求，然而也暴露了一些问题。

笔者于 2020 年 8 月参与了某地级市的市级机关的电子政务信息系统的建设工作，该电子政务信息系统根据 G2E、G2B、G2C 三种不同的模式分为办公自动化系统、政企业务信息查询系统、公共信息发布系统；笔者在该项目中担任项目经理和系统分析师一职，主要负责项目的日常管理和数据存储体系的建设工作。该市级机关的信息化建设工作起步较早，其主要部门大多均已建设了针对本部门的业务系统软件，并在日常工作中发挥了很大的作用，然而由于这些系统软件的建设工作起步较早，在建设当中没有充分考虑数据集中共享和处理的问题，因此在实际的使用当中，存在数据分散、共享困难等问题，给数据的集中使用和处理带来了困难，同时也加大了数据安全维护的难度，给数据的安全性带来了很大的隐患。

综上所述，笔者在对该机关的数据存储机制进行了充分调研之后，认为该机关原有的数据存储机制性能落后，扩展性差，不利于今后的业务系统扩展，并且安全性也存在很大的缺陷，因此，在经过项目组充分的讨论和策划后，我们决定采用集中存储架构的数据中心方案。我们从三个不同的角度来设计和实施集中式存储架构的数据中心方案，其中基本架构的设计采用以中央数据库为基础的集中式方案；数据的共享机制采用以 JDBC 为基础的中间件机制；安全方案以数据容灾机制、冗余备份机制为主要的安全机制。

在基本架构的设计工作中，我们采用了建立中央集中式数据库的方案框架，我们在硬件上采用了多台服务器组成的集群的基本结构，服务器之间通过光纤建立连接通道，之所以采用服务器集群机制而不采用高性能主机，是因为该机关的电子政务信息系统主要以业务流处理和并发处理为主，对数据的并发访问请求的实时处理要求较高，同时要求存储机制应具有良好的扩展性，因此服务器集群技术较高性能主机而言更适用于高并发数据访问请求和高扩展性的场合。数据库管理系统（DBMS）我们采用了 ORACLE 系统，该系统对海量数据的并发访问管理能力极为出色，非常适合用于如电子政务系统、大规模企业数据管理等场合。并且该系统支持厂商提供了丰富的数据处理支持工具，可以将异构数据库中的数据直接转换至中央数据库中，最大限度保护用户的投资。在数据的存储共享方面，我们采用了自行开发中间件的方式来实现统一的数据共享接口。通过对 JDBC 组件的二次包装和二次开发，我们扩展了数据库存储的接口，所有的对数据库的并发访问和操作均通过中间件进行统一的管理，从而在保证数据的共享机制和并发访问性能的基础上最大限度地保证了数据操作的安全性，同时该中间件组件还兼有对异构数据库进行数据集成、数据清洗、数据转换和校验等工作。通过数据库服务支

持厂商所提供的数据集成软件，我们将该机关原有的业务系统中的数据信息抽取出来，加载到集中式中央数据库中，并通过中间件的清洗、转换、整理等功能转换为集中式中央数据库所支持的格式，通过这种转换工作，我们可以将原本分散在各项业务系统中的业务数据集中到中央数据库中进行统一调用，同时通过修改各个业务系统的数据调用接口，使其可以直接同中央集中数据库通信，操作中央数据库的数据。

在中央集中式数据库的安全设计方面，我们采用了数据冗余备份机制，数据容灾机制等方案来保证数据的安全性和可恢复性。其中数据的冗余备份机制主要采用了两种方案，一种方案为心跳机制，即采用集群系统中两套数据存储主机互为备份，两套数据库主机之间通过心跳检测线连接起来，彼此互相定时向对方发送心跳检测信号，当其中主系统主机出现故障，不再响应心跳检测信号时，其备用主机在第一时间内接过系统的控制权，继续提供数据服务，以保证日常业务的正常运作，同时向 DBMS 发送告警信号，告知系统管理员系统出现故障。另一种方案为在数据存储主机上采用 RAID5 存储备份方案，利用 RAID5 优良的数据备份性能来保证数据的冗余备份。在数据的容灾方面，我们主要采用大型 UPS 不间断电源来在电力意外中断的时候保证数据库的数据备份工作和正常关机；在机房加装温度、湿度监控设备以随时监控机房的运行环境；采用基于工业化标准的机房通风建设方案；利用磁带备份机并制定数据库备份制度来保证数据库的日常脱机备份工作等。

当数据中心的建设工作基本完成后，我们对该存储架构进行了测试工作，测试工作的重点是通过模拟多并发访问的基本请求，测试数据库的并发访问性能；通过加载和转换模拟数据，测试集中式数据库对原系统的业务数据的支持程度，通过进行突发灾难的演戏，测试数据中心对突发灾难的抵御能力和数据恢复能力等。测试的结果编制为测试报告，并上报该机关的信息化建设委员会并得以通过。

该集中式存储架构的数据中心正式投入使用后，运行情况良好，基本实现了用户的需求。在项目总结大会上，我们对项目进行了总结，总结认为：该项目总体上是成功的，但是也存在一些待改进的地方，具体表现为：1、该集中存储数据库在并发数据访问量较大时，由于网络带宽不足，容易出现网络阻塞的现象；2、根据实际使用的情况来看，同时采用心跳检测和 RAID5 备份并无必要，反而增加了建设成本。关于第一个问题，笔者认为可以通过优化网络通信环境、消除网络瓶颈的方法来解决；至于第二个问题，笔者认为应该具体原因具体分析，根据实际的应用需求来选择一种合适的冗余备份方案。结束语：随着我国电子政务建设的不断开展，原有的分散性的数据存储架构日益不能满足需要，而集中式存储架构较于分散式数据存储架构而言，更能满足于大规模数据应用的需要，笔者认为，采用集中式存储架构，将会是电子政务建设工程今后发展的重点和趋势。

信息系统架构设计范文

摘要：

本文结合作者所参与研发的前台业务报表系统升级改造项目对如何设计信息系统的架构进行了论述。前台业务报表系统是国内某大型商业银行全行的通用报表平台，每日通过从各业务系统采集的数据进行报表展现，为该银行的决策者和经营管理人员提供各类系统交易的日报表信息平台。项目的主要内容是将原有的前台业务报表系统进行报表展现产品升级，对技术架构进行重构，对业务功能进行扩充，全面满足海内外的管理部门、业务部门对查看日趋复杂的大量的报表需求。本文首先说明

了作者在需求分析之后，软件设计之前为何重视架构设计的原因，并描述了通过分析本项目的规模、复杂程度、变化的因素等进行的新系统的架构设计。在此基础上依据具体的数据论述了作者采用的架构对于项目质量的效果。最后作者对本项目在架构设计的不足之处也做了简要分析，并提出了改进建议。

正文：

我在国内一家较大的商业银行的软件开发中心工作。由于我行前台业务报表从 99 年试点投产以来已经运行多年，其使用的报表产品版本对新操作系统的兼容等已经存在问题需要升级；原系统的技术架构及业务功能也已无法满足现今的业务报表要求。2021 年 2 月，总行规划并立项于 10 月前完成对业务报表进行改造，对使用的报表展现产品升级，对技术架构进行重构；根据日趋复杂的报表需求，对业务功能进行扩充。我有幸参与了该项目并担任系统架构设计和项目管理工作。

本项目的主要任务是将国内前台业务报表和海外业务报表应用已有功能进行重构，实现境内外框架一体化，支持多语言与多时区，境内外分行均使用统一应用体系架构实现各业务类报表的处理、展现及打印功能；将国内版数据处理和存储集中在各个一级分行，海外版数据统一集中存放在海外数据中心；对使用的报表产品进行升级，采用 SAP 公司提供的 CRYSTAL 报表工具，完成报表体系架构的调整及报表程序的移行升级。架构是信息系统的基石，对于信息系统项目的开发来说，一个清晰的架构是首要的，架构在软件需求与软件设计之间架起一座桥梁，着重解决软件系统的机构和需求向实现平坦地过渡的问题。架构在软件开发中为不同的人员提供了共同交流的语言，体现并尝试了系统早期的设计决策，并作为系统设计的抽象，为实现框架和构建的共享和重用、基于架构的软件开发提供了有力的支持。

本系统是一个系统改造项目，涉及国内 37 家分行，16 家境外机构，规模庞大而复杂，开发周期长，为保证项目质量，我们在需求分析之后，明确了本项目的开发任务，软件设计之前进行了详细的软件架构设计。考虑到以下几点：1、我行各分行及海外机构分布较散，另外随着 INTERNET 的迅速发展，部分报表信息需要通过网络向总行领导汇报展现；2、各分行对数据查询速度要求高，每日各网点产生的数据量很大，要求每日报表在 10 秒内展现，并能进行批量打印；3、银行内对数据的保密性要求高；4、增强系统的可扩展性，并能访问若干年前的报表数据。

典型的软件架构风格有很多。例如，设计图形用户界面常用的事件驱动风格、设计操作系统常用的层次化设计风格，设计编译程序厂用电管道与过滤风格、设计分布式应用程序常用的客户机/服务器风格等。一个实用的软件系统通常是几种典型架构风格的组合。经过分析，发现之前的前台业务报表系统 C/S 模式体系结构已显示出了他在异构的、分布式的网络环境中的不足，可维护性和发布性等较差，并不利于系统扩展，难以满足新系统的要求，基于 B/S 体系的 WEB 应用有利于系统的扩展性、维护性。C/S 一般建立在专用的网络上，小范围里的网络环境，局域网之间再通过专门服务器提供连接和数据交换服务。B/S 建立在广域网之上的，不必是专门的网络硬件环境，有比 C/S 更强的适应范围，一般只要有操作系统和浏览器就行；C/S 一般面向相对固定的用户群，对信息安全的控制能力很强。一般高度机密的信息系统采用 C/S 结构适宜，可以通过 B/S 发布部分可公开信息。B/S 建

立在广域网之上，对安全的控制能力相对弱。结合 CS 和 BS 架构的优缺点，我们最终采用 C/S 和 B/S 混合的架构设计。

一、系统支持分布式部署和集中式部署两种方式。国内采用分布式部署方式，应用环境部署在各一级分行，数据库部署两个 Oracle 实例。海外采用集中式部署方式，应用环境部署在海外数据中心，数据库部署两个 Oracle 实例，两实例间通过 DBLINK 进行参数表的同步。客户端程序国内部署于各二级分行、支行和网点；海外部署于海外分行。

二、报表文件的生成：（1）为减少传输的数据量，批量打印及离线归档采用文本方式，并且文本可以进行压缩传输，可有效减少网络传输流量。（2）所有报表数据文件都预生成，生成文本文件的批量服务器端进行自行开发，不依赖于报表工具产品，以根据数据本身的特色进行开发提高批量处理效率。例如采用 SQL 嵌入到 C 编程语言中，利用 C 语言提高对文件处理的效率。（3）在批量服务器输出文件的同时，把相关的结果也写入到数据库的 B/S 查询表中供 B/S 客户端查询展现，B/S 查询表的数据库表的结构与批量打印使用的文本文件结构一致。

三、报表查看、批量打印部分：（1）B/S 架构：报表查看结果的 HTML 页面由服务器生成，客户端本地不处理任何数据。对于 B/S 客户端方式，是一个纯集中式系统，数据的逻辑处理以及显示处理都是集中式的。对于 B/S 的报表查询需设立缓存机制。（2）C/S 架构：是一个服务器与分布式结合的报表处理系统。服务器端主要工作是数据的逻辑处理，例如报表的汇总、统计、字典转换、币种折算、数据表关联等各类数据的逻辑处理。四、版本在线自动更新功能：C/S 客户端提供版本在线自动更新功能。打开客户端程序后，客户端会根据临时文件夹及更新日志记录自行检查是否与服务器版本号相符，若发现不符合，则客户端自行下载最新的版本，并自行安装新版本，解决网点的由于报表频繁升级的安装管理维护问题。

前台业务报表升级改造项目最终在 2021 年底投产并取得了较大的成功。完全符合海内外各分行及总行管理部门的需求，得到了大家一致好评。当然，本项目的是存在一些不足之处的。我们在设计时将大部分的精力都放在了整体功能测试上，对于性能的设计上只考虑整体情况，较少考虑具体的数据库性能优化，例如表的索引设计以及锁事务处理机制我们有所忽略。导致投产后仍有少量交易因 SQL 访问效率问题出现了响应时间过长的情况。这一点也是我们今后的项目设计中需要注意改进的地方。

2 论面向服务架构设计及其应用

试题：论面向服务架构设计及其应用

企业应用集成（Enterprise Application Integration, EAI）是每个企业部必须要面对的实际问题。面向服务的企业应用集成是一种基于面向服务体系结构（Service-Oriented Architecture, SOA）的新型企业应用集成技术，强调将企业和组织内部的资源和业务功能暴露为服务，实现资源共享和系统之间的互操作性，并支持快速地将新的应用以服务的形式加入到已有的集成环境中，增强企业 IT 环境的灵活性。

请围绕“SOA 在企业集成架构设计中的应用”论题，依次从以下 3 个方面进行论述。

- 1、概要叙述你参与管理和实施的企业应用集成项目及你在其中所担任的主要工作。
- 2、具体论述 SOA 架构的内容、特点，以及你熟悉的工具和环境对 SOA 的支持，在应用中重点解决了哪些问题。
- 3、通过你的切身实践详细论述 SOA 在企业应用集成中发挥的作用和优势。

找准核心论点

问题 1 要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题 2 要点：

SOA 架构的内容、特点，以及你熟悉的工具和环境对 SOA 的支持。

在应用中重点解决了哪些问题。

问题 3 要点：

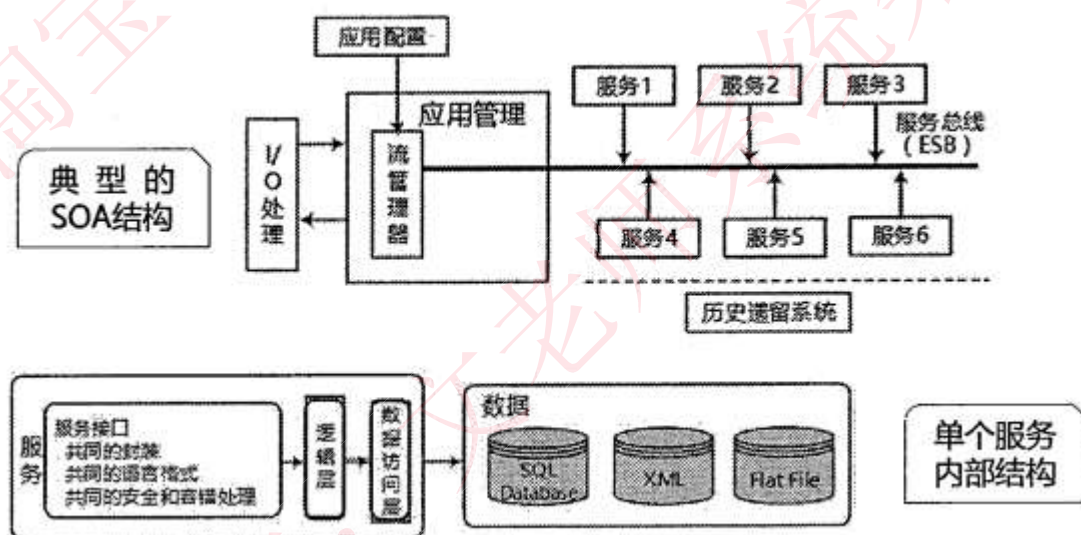
SOA 在企业应用集成中发挥的作用和优势。

理论素材准备

SOA 是一种粗粒度、松耦合服务架构，服务之间通过简单、精确定义接口进行通信，不涉及底层编程接口和通信模型。

在 SOA 中，服务是一种为了满足某项业务需求的操作、规则等的逻辑组合，它包含一系列有序活动的交互，为实现用户目标提供支持。

SOA 并不仅仅是一种开发方法，还具有管理上的优点，管理员可直接管理开发人员所构建的相同服务。多个服务通过企业服务总线提出服务请求，由应用管理来进行处理，如下：



实施 SOA 的关键目标是实现企业 IT 资产重用的最大化，在实施 SOA 过程中要牢记以下特征：可从企业外部访问、随时可用（服务请求能被及时响应）、粗粒度接口（粗粒度提供一项特定的业务功能，而细粒度服务代表了技术构件方法）、服务分级、松散耦合（服务提供者和服务使用者分离）、可

重用的服务及服务接口设计管理、标准化的接口（WSDL、SOAP、XML 是核心）、支持各种消息模式、精确定义的服务接口。

从基于对象到基于构件再到基于服务，架构越来越松散耦合，粒度越来越粗，接口越来越标准。

基于服务的构件与传统构件的区别有四点：

服务构件粗粒度，传统构件细粒度居多；

服务构件的接口是标准的，主要是 WSDL 接口，而传统构件常以具体 API 形式出现；

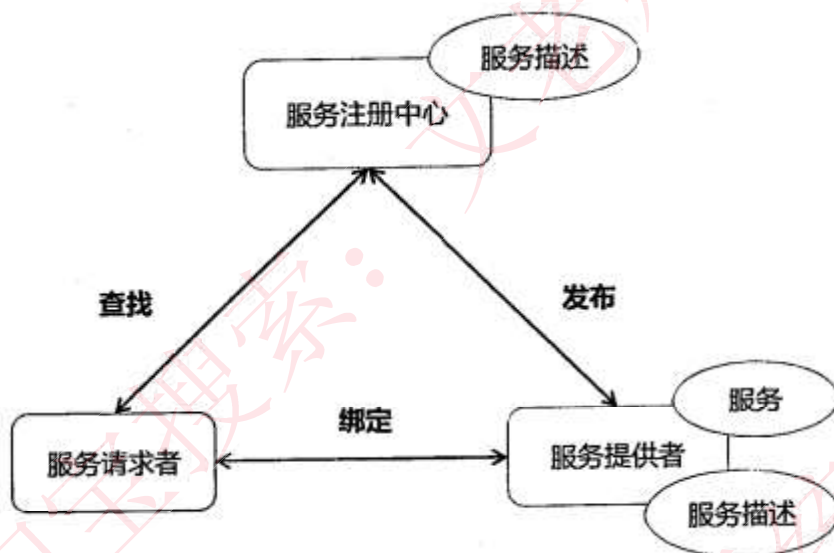
服务构件的实现与语言是无关的，而传统构件常绑定某种特定的语言；

服务构件可以通过构件容器提供 QoS 的服务，而传统构件完全由程序代码直接控制。

SOA 的实现方式

WEB Service

服务提供者、服务注册中心（中介，提供交易平台，可有可无）、服务请求者。服务提供者将服务描述发布到服务注册中心，供服务请求者查找，查找到后，服务请求者将绑定查找结果。如下图：



合格范文赏析

摘要：2016 年 8 月，我参与了胶凝砂砾石坝施工质量监控系统的开发工作，该系统旨在帮助水利工程建设法人单位、施工企业、监理单位及相关政府部门解决水利工程建设施工质量监控和工程项目管理等问题。我在该项目中担任系统分析师，主要负责该系统的系统分析及设计工作。本文以胶凝砂砾石坝施工质量监控系统为例，主要论述了 SOA 在企业集成架构设计中的具体应用。服务提供者主要完成服务的设计、描述、定义和发布等相关工作；服务注册中心保证该系统各个模块、服务的相互独立性与松耦合；服务请求者通过 WebService 技术调用服务。实践证明，通过以上技术的应用有效实现了资源共享和系统间的互操作性，提高了系统的灵活性，最终系统顺利上线，获得用户一致好评。

胶凝砂砾石坝是在面板坝和碾压混凝土重力坝基础上发展起来的一种新坝型，其特点是采用胶凝砂砾石材料筑坝，使用高效率的土石方运输机械和压实机械施工。与常规坝型相比，胶凝砂砾石坝在适用性和经济性方面具有独特的优势，可以就地、就近取材，不需设置集料筛分，施工进度快，施工工序简单高效，因而要求施工过程紧凑，高峰期筑坝效率要求高，这给施工质量控制带来了一定的困

难和风险，需要综合考虑影响施工质量的各方面因素，尽量采用自动化监控手段，加强实时质量监控力度，这使胶凝砂砾石坝施工质量监控系统应运而生。

2016年8月，我参与了胶凝砂砾石坝施工质量监控系统的开发工作，担任该系统的系统分析师，主要负责该系统的系统分析及设计工作。该系统的主要功能模块包括采料监控、运料监控、拌合监控、碾压监控和温湿度监控等。旨在帮助水利工程建设法人单位、施工企业、监理单位及相关政府部门，解决水利工程建设施工质量监控和工程项目管理等问题，通过信息技术和施工信息现场采集、实时传输、统一存储、科学分析和在线处理，及时生成质量监控报表和发布质量预警信息，提高水利工程建设管理和科学化、现代化和信息化，落实法人负责、监理控制、施工保证、政府监督等各项职能。因此，要满足该系统的需求，选择一种合适的架构技术至关重要。

SOA 是一种应用程序架构，在这种架构中，所有功能都定义为独立的服务，服务之间通过交互和协调完成业务的整体逻辑。SOA 指定了一组实体，包括服务提供者、服务消费者、服务注册表、服务条款、服务代理和服务契约，这些实体详细说明了如何提供和消费服务。服务提供者提供符合契约的服务，并将他们发布到服务代理。这些服务是自我包含的、无状态的实体，可以由多个组件组成。服务代理者作为存储库、目录库或票据交换所，产生由服务提供者发布的事先定义的标准化接口，使得服务可以提供给在任何异构平台和任何用户接口使用。这种松散耦合和跨技术实现，使各服务在交互过程中无需考虑双方的内部实现细节、实现技术、以及部署在什么平台上，服务消费者只需要提出服务请求，就可以发现并调用其他的软件服务得到答案。SOA 作为一种粗粒度、松耦合的架构，具有松散耦合、粗粒度服务、标准化的接口、位置和传输协议透明、服务的封装和重用、服务的互操作等几个特点。

该系统要求开发周期短，系统灵活性高等，结合 SOA 的特点，我们最终采用了面向服务的、基于 SOA 的企业应用集成。下面具体论述其应用过程。

1、服务提供者

服务提供者主要完成服务的设计、描述、定义和发布等相关工作。经过对水利行业施工工程及施工工艺的深入研究，通过查阅《胶凝砂砾石坝施工指南》等相关资料，根据企业应用集成的要求，对胶凝砂砾石坝施工质量监控系统的业务流程进行梳理；综合考虑服务粗粒度、松耦合、自包含和模块化等特点进行服务的设计。为了避免服务通信期间，信息量过大，服务之间交互过于频繁，尽可能的减少了服务的数量。同时，为了保证服务自身功能的完整性，尽可能的减少服务与系统之间的通信，在胶凝砂砾石坝施工质量监控系统的分析与开发过程中，先行设计，提取出了两个必要，急需的服务便于日后集成使用，其中包括拌合监控中标准拌合比对比服务和碾压监控中的碾压轨迹生成服务。

在标准拌合比对比服务中主要实现针对现有拌合配比与标准拌合比的对比，以判断现有拌合配比是否符合标准的工作。由于胶凝砂砾石坝就地、就近取材的特性，因此在不同的水利施工工地所使用的采料也不尽相同，标准拌合比对比服务预留了标准拌合比的输入接口，以适应不同的需求。在碾压轨迹生成服务中主要实现读取定位信息绘制碾压轨迹，以监控是否存在漏碾和欠碾的情况。由于受到胶凝砂砾石坝选址和机密程度的限制，定位信息可以选择 GPS 或者超宽带技术，但是两种定位的方式的数据格式并不相同，因此碾压轨迹生成服务的开发中预留了两种数据格式的接口来读取定位信息。

待完成服务设计之后，服务提供者采用 WSDL 进行服务描述，而后再利用 UDDI 技术将这些服务信息发布至服务注册中心，公布查找和定位服务的方法。

2、服务注册中心和服务请求者

在胶凝砂砾石坝施工质量监控系统采用了服务注册中心。服务注册中心并不是一个必选角色，但是为了保证该系统各个模块、服务的相互独立性与松耦合，在该系统中依然保留了服务注册中心。同时，服务注册中心的存在也使得服务请求者与服务提供者之间进一步解耦。在服务注册中心包含有已发布的标准拌合比对比服务与碾压监控中的碾压轨迹生成服务的描述信息，其描述信息主要包括服务功能描述、参数描述、接口定义、信息传递等相关信息。

服务请求者通过 WebService 技术调用服务。当服务完成发布，在服务请求者要使用已发布的服务。利用 Web Service 技术在拌合监控阶段，通过服务注册中心获取拌合监控中标准拌合比对比服务的相关功能，接口，参数及其返回值等相关服务信息；之后使用 Web Service 技术传递服务所需的标准拌合比等相关参数，进而调用该服务相关的运算、处理和分析。利用 Web Service 技术在拌合监控阶段，通过服务注册中心获取实时施工数据采集处理服务的服务定义和功能，接口，参数及其返回值等相关服务信息；之后根据施工工地的具体情况选择不同的定位方式，传递服务所需相关参数，最后实时施工数据采集处理服务运行结束返回的绘制碾压轨迹坐标点同样是利用 Web Service 技术传递至服务请求者。服务请求者接收到碾压轨迹的坐标点后最终完成碾压轨迹的绘制工作并在界面中将其呈现出来。在这期间服务请求者无需了解服务是如何对数据进行处理和分析的。

整个项目历时 10 个月开发，于 2017 年 6 月完成交付，到目前运行稳定。通过在水利施工工地等恶劣环境下的一段时间的使用，用户普遍反馈良好。总体来讲，选用 SOA 有如下优势：1、系统更易维护。当需求发生变化时，不需要修改提供业务服务接口，只需要调整业务服务流程或者修改操作即可。2、更高的可用性。该特点是在于服务提供者和服务请求者的松散耦合关系上得以发挥与体现。这种没有绑定在特定实现上、具有中立的接口定义的特征称为服务之间的松耦合。松耦合有两个明显的优势，一是它的灵活性，其独立于实现服务的硬件平台、操作系统和编程语言；二是当组成整个应用程序的每个服务的内部结构和实现逐渐地发生改变时，它能够继续存在。3、更好的伸缩性，依靠业务服务设计、开发和部署等所采用的架构模型实现伸缩性。使得服务提供者可以互相彼此独立地进行调整，以满足新的服务需求。

实践证明，SOA 技术的使用大大提高了系统开发效率，节省了开发和维护成本，使得系统具有更好的开放性、易扩展性和可维护性，从项目完工后的使用效果来看，达到了预期目的。

3 论软件系统建模方法及其应用

在 2022 年架构论文考试里出了一道软件维护和可维护性的题目，这是关于系统开发的，原来是属于系统分析师的考点，**那么在 2023 年架构论文预测里，文老师大胆预测一个有关开发阶段的论文题，就是系统分析阶段的系统建模方法，大家写法可以从结构化建模或面向对象建模这两个方向出发。**

以下是面向对象建模范文

摘要：

2018年3月1日至12月20日，我参加了“数据安全访问平台”项目的开发，担任系统分析员的工作。该项目是某行业用户“数据中心二期”建设的主要内容，目标是：建立数据统一访问接口及其使用标准，规范、约束和审计数据应用访问数据库的行为，对数据应用提供强制审计的技术手段。由于该系统是所有应用的基础平台，对系统的可靠性与性能有较高要求，同时由于没有成熟的现有系统作为参照，该项目存在较高的风险。本文结合作者实践，讨论了在项目中基于UML的需求分析。我们使用用例图描述用户与系统的交互；使用类图描述系统的核心概念；使用部署图描述系统的网络部署；使用活动图描述系统的应用流程。由于采用了UML中的多种技术，使得我们能从多个方面完整的把握需求，有效的保证到了需求工作的质量。最后，分析了需求工作中存在的问题和改进的方法。

正文：

2018年3月1日至2018年12月20日，我参加了“数据安全访问平台”项目的开发，担任系统分析员的工作。“数据安全访问平台”是某行业用户“数据中心二期”建设的主要内容。在一期建设中已建成数据的统一存储和统一分发框架。但主要存在以下问题：无法获得应用用户对数据库的操作日志；开发人员对数据库的使用不规范，查询的结果集过大，导致数据库的性能大幅下降；应用直接使用数据库的登录数据库，存在着一定的安全隐患。“数据安全访问平台”的目标是：建立数据统一访问接口及其使用标准，规范、约束和审计数据应用访问数据库的行为，对数据应用提供强制审计的技术手段。该项目具有较高的业务需求风险和技术风险。由于没有成熟系统做为参照，该项目需求不是很明确，而且系统涉及甲方多个利益相关方，各方对系统的安全和审计功能、运行维护、可靠性、性能和易用性有者不同的观点，某些观点之间还存在冲突。同时系统作为“数据中心”的基础设施之一，所有的应用系统都要通过本系统完成数据库访问。系统的可靠性和性能直接影响到应用系统的正常运行。整个系统分为6个子系统，包括JDBC驱动封装子系统、ADO.Net驱动封装子系统、WebService接口子系统、管理配置网站、存储子系统（SQL Server2005 数据库，存储配置信息）和监控子系统（数据库网络协议分析与连接控制）。

UML在需求工作中的应用 项目组采用一个精简RUP开发模型指导项目的整个开发流程。在初始阶段主要采用用户访谈、用户调查和联合讨论会捕获用户需求，进行初步分析，完成《需求规格说明书》初稿，通过用户评审；在细化阶段，对需求进行了细化，并在采用原型法验证用户需求，完成《需求规则说明书》更新稿，通过用户评审，作为项目验收的依据。项目开发中，我们采用了统一建模语言（UML），并使用了Rational Rose工具。在需求工作中，我们主要使用了UML中的用例图、类图、活动图和部署图。

用例技术的应用 整个需求开发都是围绕着用例技术开展的。首先，我们明确了系统的利益（查书）相关方、确定了系统边界和建设目标，以及主要功能特性。由于系统涉及甲方多个利益相关方，包括：应用科、维护科、网络科、信息中心领导和应用开发方，他们对系统的安全和审计功能、运行维护、可靠性、性能和易用性有者不同的观点，同时本系统与已建成的网管系统和单点登录系统存在着交互关系，这给确定系统的目标和边界带来一定的难度。项目初始阶段开始，我们先通过用户访谈、

用户调查了解了各个用户对系统的观点；随后，我们召开了联合讨论会，会上我们描述了各个用户的观点，以及之间可能的冲突，供各方进行充分的讨论，会议最后，信息中心的领导统一了各方的意见，对系统达成一致的目标，并明确了系统主要的功能特性，确定本系统与网关系统和单点登录系统的关系。其次，我们建立用例模型，并细化关键用例。明确了系统的目标和主要功能特性后，我们采用用例模型对需求进行分析。整个系统包括六个用例包：访问接口包用例包、审计信息管理用例包、用户认证信息管理用例包、数据库连接资源管理用例包、访问控制信息管理用例包、数据库连接监控管理用例包和审计日志管理用例包，共包含 55 个用例。对大部分用例我们只描述了一个基本正确流程，只对 5 个关键高风险用例进行了细化描述，包括：前置条件、后置条件、基本流、扩展流和相关约束。这项工作完成后，我们编写了《需求规格说明书》初稿，提交用户进行了审核。经过修改后，通过用户评审，作为初始化阶段的主要成果。最后，细化用例模型。在细化阶段，我们细化描述了所有的用例，完成《需求规则说明书》更新稿，通过用户评审后，作为项目验收的依据。

类图的应用 用例技术描述了系统需求的动态结构，但对于需求特性和用例中出现的概念，并没有统一的分析。我们使用类图描述系统的核心概念。本项目中涉及的核心概念主要包括：逻辑连接、用户、应用、受限角色、认证 SQL 语句、参数取值范围的约束和查询结果集的限制。在分析核心概念时，我们主要关注：概念之间的关联关系，是否具有相同的生命周期，和具体的对应关系（一对多、多对多和多对一）。

部署图与活动图的应用 由于整个系统包含多个子系统，各个子系统部署在不同的节点，需要考虑用户的网络结构是否能支持。在需求阶段，我们也描述了整个系统的部署。其中监控子系统比较特殊，由于其是通过分析 Oracle 数据库的网络数据包进行工作的，因此监控子系统必须接入核心交换机的镜像端口。我们将部署图与网络科进行了确认。我们使用活动图描述系统的应用场景。由于本系统对应用系统的开发增加了一层管理，因此应用系统的开发方与信息中心存在一个交互流程：应用开发方首先使用本系统的开发版进行本地开发，并填写基本配置数据；然后，在用户的生产环境中部署应用，并提交基本配置数据；最后，信息中心的管理员对配置数据进行修改后，应用在生产环境中才能运行。我们使用流程图描述了整个过程。用通道表示应用开发方和信息中心，并描述了各个通道内的流程以及通道之间的交互。

由于没有成熟系统做为参照，该项目具有较高的需求风险。由于在整个开发过程中使用了 UML 的用例图、类图、部署图和活动图，这使得我们能从多个方面完整的把握需求，有效的保证到了需求工作的质量。本项目的需求工作中，我们也遇到的一些问题，主要是维护 Word 文档与模型的一致性。我们使用 RationalRose 建模，使用 Word 写文档，通过截图的方式将模型插入文档，因此需要手工维护两者的一致性。这种方式较繁琐，容易出错。今后的工作中，我们准备使用 Rational 公司的 Soda 工具，自动将 Rose 中的模型插入到 Word 文档中。

4 论软件系统架构风格

论软件系统架构风格

系统架构风格（System Architecture Style）是描述某一特定应用领域中系统组织方式的惯用模式。架构风格定义了一个词汇表和一组约束，词汇表中包含一些构件和连接件类型，而这组约束指出系统是如何将这些构件和连接件组合起来的。软件系统架构风格反映了领域中众多软件系统所共有的结构和语义特性，并指导如何将各个模块和子系统有效地组织成一个完整的系统。软件系统架构风格的共有部分可以使得不同系统共享同一个实现代码，系统能够按照常用的、规范化的方式来组织，便于不同设计者很容易地理解系统架构。

请以“软件系统架构风格”论题，依次从以下三个方面进行论述：

1. 概要叙述你参与分析和开发的软件系统开发项目以及你所担任的主要工作。
2. 分析软件系统开发中常用的软件系统架构风格有哪些？详细阐述每种风格的具体含义。
3. 详细说明在你所参与的软件系统开发项目中，采用了哪种软件系统架构风格，具体实施效果如何。

找准核心论点

问题1要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题2要点：

常用的软件系统架构风格。

每种风格的具体含义。

问题3要点：

采用的哪种软件系统架构风格。

具体实施效果如何。

理论素材准备

架构设计的一个核心问题是能否达到架构级的软件复用。架构风格反映了领域中众多系统所共有的结构和语义特性，并指导如何将各个构件有效地组织成一个完整的系统。

架构风格定义了用于描述系统的术语表和一组指导构建系统的规则。

数据流风格：批处理序列、管道-过滤器

调用/返回风格：主程序/子程序、面向对象、层次结构

独立构件风格：进程通信、事件驱动系统（隐式调用）

虚拟机风格：解释器、基于规则的系统

仓库风格：数据库系统、超文本系统、黑板系统

合格范文赏析

摘要：本人于2016年1月参与浙江省某市公交集团“公交车联网一体化”项目，该系统为新能源营运车辆补贴监管、安全监控等方面提供全方位的软件支撑，在该项目组中我担任系统架构师岗位，主要负责整体架构设计与中间件选型。

本文以该车联网项目为例，主要讨论了软件架构风格在该项目中的具体应用。底层架构风格我们采用了虚拟机风格中的解释器，因该公交共有几十种不同的数据协议，使用解释器风格可以满足整车数据协议兼容性需求；中间层关于应用层的数据流转我们采用了独立构件风格中的隐式调用，这种风格主要用于减低系统间耦合度、简化软件架构，提高可修改性方面的架构属性；应用系统层我们采用了 B/S 的架构风格，统一解决公交行业性难题“实施推广难、维护难”问题。最终项目成功上线，获得用户一致好评。

随着国家十三五计划中-能源战略的深入和推广，该市公交集团自 2016 年 1 月起全面停止采购燃油机公交车，规划到 2020 年纯电公交车采购占比必须在 70%以上，同时配套将车联网方面的系统建设被列为工作重点。不管从新能源营运车辆补贴监管、安全监控或者公交公司自身的营运和机护需求，都要求有新的车联网系统对他们进行全方位的支持，而我司是该公交的主要仪表与 can 模块产品的主要供应商，全市 4000 多台车中有 3000 多辆是我司的产品，我司不仅掌握熟悉该公交整车数据而且在车联网底层 can 数据有非常明显的领域知识优势，因此 2016 年 1 月我司被该市公交集团委托建设公交集团车联网一体化项目。本项目组全体成员共有 27 人（不含业主方），我在项目中为担任系统架构师职务，架构小组共 4 人，我主要职责负责整体架构设计与中间件选型，4 月份完成架构工作，整个项目共耗时了 7 个月，2016 年 8 月顺利通过验收。

在架构工作开始阶段，我们便意识到，架构风格是一组设计原则，是能够提供抽象框架模式，可以为我们的项目提供通用解决方案的，这种能够极大提高软件设计的重用的方法加快我们的建设进程，因此在我司总工程师的建议下，我们使用了虚拟机风格、独立构件风格以及 B/S 架构风格这三种较常用风格。虚拟机风格中的解释器架构风格能够提供灵活的解析引擎，这类风格非常适用于复杂流程的处理。独立构件风格包括进程通讯风格与隐式调用风格，我们为了简化架构复杂度采用了隐式调用风格，通过消息订阅和发布控制系统间信息交互，不仅能减低系统耦合度，而且还提高架构的可修改性。B/S 架构风格是基于浏览器和服务器的软件架构，它主要使用 http 协议进行通信和交互，简化客户端的工作，最终减低了系统推广和维护的难度，以下正文将重点描述架构风格的实施过程和效果。

底层架构我们使用解释器风格来满足整车数据协议兼容性需求。解释器风格是虚拟机风格中的一种，具备良好的灵活性，在本项目中我们的架构设计需要兼容好 86 种不同 can 数据协议，一般来说这种软件编写难度非常高，代码维护难度压力也很大，因此这个解释器的设计任务便很明确了，软件设计需要高度抽象、协议的适配由配置文件来承担。具体的做法如下，我们对各个车厂的 can 数据结构进行了高度抽象，由于 can 数据由很多数据帧组成，每个数据帧容量固定并且标识和数据有明确规定，因此我们将 can 协议中的 ID 和数据进行关系建模，将整体协议标识做为一个根节点，以 canid 作为根节点下的叶子节点，使用 XML 的数据结构映射成了有整车协议链-数据帧-数据字节-数据位这 4 层的数据结构，核心的代码采用 jdom.jar 与 java 的反射机制动态生成 java 对象，搭建一套可以基于可变模板的解释器，协议模板的产生可以由公交公司提供的 excel 协议文档进行转换得到，解释器支持协议模板热部署，这种可以将透传二进制数据直接映射成 java 的可序列化对象，将数据协议的复杂度简化，后期数据协议更改不会对软件产生影响，仅仅更改协议模板文件即可，最终我们使用了 86 个协议描述文件便兼容了这些复杂的 can 数据协议，规避了 can 数据巨大差异带来的技术风险。

中间层我们使用独立构件风格中的隐式调用来简化构件间的交互复杂度，降低系统耦合度。主要的实现手段是，我们采用了一个开源的消息中间件作为连接构件，这个构件是 apache 基金会下的核心开源项目 activemq，它是一款消息服务器，其性能和稳定性久经考验。由上文提到的解释器解析出对象化数据经过 activemq 分发到各个订阅此消息的应用系统，这些应用系统包括运营指挥调度、自动化机护、新能源电池安全监控等，这种多 web 应用的情况非常适合采用消息发布与消息订阅的机制，能够有效解决耦合问题，我们在编码的过程中发现只要采用这种风格的 web 应用，整个迭代过程效率极高，错误率降低，而且我们使用的 spring 框架，消息队列的管理完全基于配置，清晰简单，维护性良好，例如整车安全主题、运营调度主题、机护维修主题等消息队列分类清晰，可以随时修改其结构也能够随时增其他主题的消息队列，不同的 web 系统监听的队列也可以随时变换组合，基于消息中间件的架构设计能够让系统的构件化思路得到良好实施，总体来说这种架构风格带来了非常清晰的数据流转架构，简化了编码难度，减低本项目的二次开发的难度。

应用系统层我们主要采用 B/S 的架构风格，主要用于解决公交推广难、维护难得问题。公交行业有一个明显的特点，公交子公司分布在全市各个地区，路途很远，且都是内网通讯，车联网络也是走的 APN 专网，一般是无法远程支持的，这给我们的系统推广以及后期维护带来了很大难题，我们可以想象如果使用 C/S 架构，更新客户端一旦遇到问题很可能需要全市各个站点跑一遍。这让我们在系统推广和维护方面面临较大压力。我们采用的 B/S 架构风格能够解决这个难题，并充分考量可现在的相关技术成熟度，例如现在的 html5 完全能够实现以前客户端的功能，项目中我们使用了大量的前端缓存技术与 websocket 技术，能够满足公交用户实时性交互等需求。这种风格中页面和逻辑处理存储在 web 服务器上，维护和软件升级只要更新服务器端即可，及时生效，用户体验较好，例如界面上需要优化，改一下 Javascript 脚本或者 CSS 文件就可以马上看到效果了。

项目于 2016 年 8 月完成验收，这 1 年内共经历了 2 次大批量新购公交车辆接入，这几次接入过程平稳顺利，其中协议解释器软件性能没有出现过问题，消息中间件的性能经过多次调优吞吐量也接近了硬盘 IO 极限，满足当前的消息交互总量，另外由于我们的项目多次紧急状态下能够快速适应 can 协议变动，得到过业主的邮件表扬。除了业主机房几次突发性的网络故障外，项目至今还未有重大的生产事故，项目组现在留 1 个开发人员和 1 个售后在维护，系统的维护量是可控的，系统运行也比较稳定。

不足之处有两个方面，第一在架构设计的过程中我们忽略 PC 配置，个别 PC 因为需要兼容老的应用软件不允许系统升级，这些电脑系统老旧，其浏览器不支持 html5，导致了系统推广障碍。第二在系统容灾方面还有待改善。针对第一种问题，我们通过技术研讨会说服可业主新购 PC，采用两台机器同时使用方式解决。针对第二种问题我方采用了服务器冗余和心跳监测等策略，在一台服务暂停的情况下，另外一台服务接管，以增加可用性。

5 论软件架构评估

论软件系统架构评估

对于软件系统，尤其是大规模的复杂软件系统来说，软件的系统架构对于确保最终系统的质量具有十分重要的意义，不恰当的系统架构将给项目开发带来高昂的代价和难以避免的灾难。对一个系统架构进行评估，是为了：分析现有架构存在的潜在风险，检验设计中提出的质量需求，在系统被构建之前分析现有系统架构对于系统质量的影响，提出系统架构的改进方案。架构评估是软件开发过程中的重要环节。请围绕“论软件系统架构评估”论题，依次从以下三个方面进行论述。

1. 概要叙述你所参与架构评估的软件系统，以及在评估过程中所担任的主要工作。
2. 分析软件系统架构评估中所普遍关注的质量属性有哪些？详细阐述每种质量属性的具体含义。
3. 详细说明你所参与的软件系统架构评估中，采用了哪种评估方法，具体实施过程和效果如何。

理论素材准备

质量属性

1、性能：指系统的响应能力，即要经过多长时间才能对某个事件做出响应，或者在某段时间内系统所能处理的事件的个数。如响应时间、吞吐量。设计策略：优先级队列、增加计算资源、减少计算开销、引入并发机制、采用资源调度等。

2、可靠性：是软件系统在应用或系统错误面前，在意外或错误使用的情况下维持软件系统的功能特性的基本能力。如MTTF、MTBF。设计策略：心跳、Ping/Echo、冗余、选举。

3、可用性：是系统能够正常运行的时间比例，经常用两次故障之间的时间长度或在出现故障时系统能够恢复正常的速度来表示。如故障间隔时间。设计策略：心跳、Ping/Echo、冗余、选举。

4、安全性：是指系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务的能力。如保密性、完整性、不可抵赖性、可控性。设计策略：入侵检测、用户认证、用户授权、追踪审计。

5、可修改性：指能够快速以较高的性能价格比对系统进行变更的能力。通常以某些具体的变更为基准，通过考察这些变更的代价衡量。设计策略：接口-实现分类、抽象、信息隐藏。

6、功能性：是系统所能完成所期望的工作的能力。一项任务的完成需要系统中许多或大多数构件的相互协作。

7、可变性：指体系结构经扩充或变更而成为新体系结构的能力。这种新体系结构应该符合预先定义的规则，在某些具体方面不同于原有的体系结构。当要将某个体系结构作为一系列相关产品的基础时，可变性是很重要的。

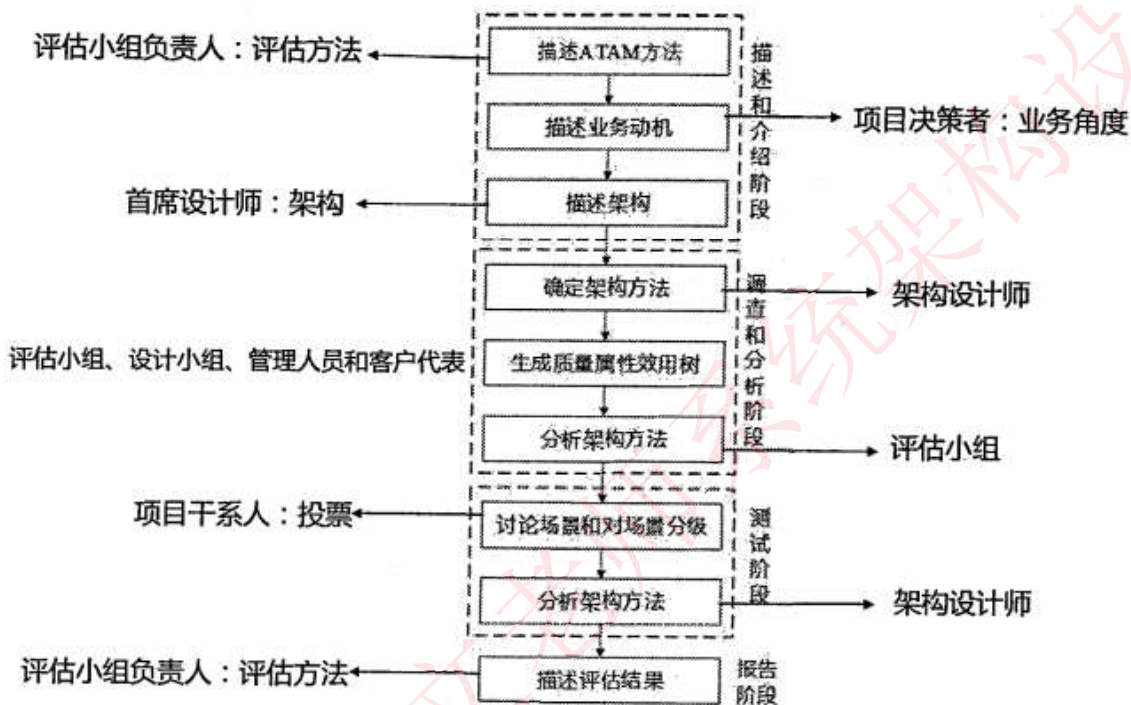
8、互操作性：作为系统组成部分的软件不是独立存在的，经常与其他系统或自身环境相互作用。为了支持互操作性，软件体系结构必须为外部可视的功能特性和数据结构提供精心设计的软件入口。程序和用其他编程语言编写的软件系统的交互作用就是互操作性的问题，也影响应用的软件体系结构。

敏感点：是指为了实现某种特定的质量属性，一个或多个构件所具有的特性。

权衡点：是影响多个质量属性的特性，是多个质量属性的敏感点。

风险点与非风险点不是以标准专业术语形式出现的，只是一个常规概念，即可能引起风险的因素，可称为风险点。某个做法如果有隐患，有可能导致一些问题，则为风险点；而如果某件事是可行的可接受的，则为非风险点。

架构权衡分析法 ATAM



合格范文赏析

2016 年 3 月，我公司承担了国家某安全中心漏洞挖掘系统的开发工作，我在该项目中承担系统架构设计师的职务，主要负责系统的架构设计。该项目的主要目的是依托大数据平台从互联网流量中挖掘未知漏洞。本文以漏洞挖掘系统为例，论述了软件系统的架构评估。首先分析了软件架构评估所普遍关注的质量属性并阐述了其性能、可用性、可修改性和安全性的具体含义。整个系统采用了面向服务 SOA 的架构设计方法。在架构设计完成之后，对 SA 评估采用了基于场景的评估方式中的体系结构权衡分析方法 ATAM，并详细描述了其评估过程，项目评估小组经过对项目的风险点、敏感点和权衡点的讨论后生成了质量效应树。目前系统已稳定运行一年多，从而验证了该项目采用 ATAM 架构评估保证了系统的顺利完成。

随着互联网的快速发展，网络上出现的安全问题越来越多，从互联网发展至今，已经爆发了众多的网络攻击事件，如网络蠕虫病毒感染、主机被控制、数据库被非法访问、非法电子银行转账等等。针对这些安全问题，很有必要开发一种 web 漏洞的发现和利用技术。2016 年 3 月我公司承接了国家某安全中心漏洞挖掘系统的开发工作。该项目通过对互联网中的流量进行特征分析，从中提取出相关的攻击内容，并将这些内容存储到大数据平台，结合大数据分析技术，对攻击者进行跟踪分析，从而捕获出未知漏洞。通过这种漏洞挖掘技术可以极大的解决大数据，大流量背景下 web 攻击入侵，帮助用户做好“事中”的安全工作，协助安全厂商对互联网攻击进行针对性过滤。

系统在整体架构上采用了面向服务的架构 SOA。前端采用了 PHP 进行开发，后台流量分析工作采用运行性较高的 c 语言在 Linux 服务器上开发，流量包存储使用了企业磁盘阵列，数据存储采用了 mysql。通过将系统拆分为多个子模块，各个子模块的构建上用服务进行了封装，它们之间通过消息进行通信。经过对客户需求的分析，我将该系统拆分为流量捕获模块（负责从互联网中捕获流量）、

pcap 文件存储模块（负责将互联网中的流量存储到大数据平台）、流量分析模块（负责对流量进行分析验证）、数据库模块（负责漏洞数据的存储）和 web 管理模块（负责下发漏洞规则和查看漏洞信息）。下面先介绍下软件架构评估的质量属性。

架构评估是软件开发过程中的重要环节，在软件架构评估中的质量属性有：性能、可用性、可修改性、安全性、可测试性、可靠性和易用性等。其中前 4 个质量属性是质量效应树的重要组成部分。性能是指系统的响应能力，即经过多长时间对事件做出响应。可用性是指系统能够正常运行的比例，通过用两次故障之间的时间长度或出现故障时系统能够恢复的速度来表示。可修改性是指系统能以较高的性价比对系统做出变更的能力。安全性是指系统能够向合法用户提供服务，同时拒绝非授权用户使用或拒绝服务的能力。

常用的架构评估方法有：基于问卷调查的评估方式、基于场景的评估方式和基于度量的评估方式。基于问卷调查的评估方式是由多个评估专家通过调查问卷的方式回答问卷中的问题，对多个评估结果进行综合，最终得到最终结果。其评价的具有主观性不太适合本项目。基于度量的评估方式虽然评价比较客观，但是需要评估者对系统的架构有精确的了解，也不太适合本项目。而基于场景的评估要求评估者对系统中了解，评价比较主观，故本项目采用了基于场景的评估方式。基于场景的评估方式又分为架构权衡分析法 ATAM，软件架构分析法 SAAM 和成本效益分析法 CBAM。本项目中根据不同质量属性使用了 ATAM 作为系统架构评估的方法。

在使用 ATAM 进行架构评估时，我们根据项目需要成立了项目评估小组。其主要成员包括：评估小组负责人、项目决策者、架构设计师、用户、开发人员、测试人员、系统部署人员等项目干系人。我在这里的身份是项目的评估小组负责人和首席架构师。架构的评估经历了描述和介绍阶段、调查和分析阶段、测试阶段和报告阶段四个阶段。下面我分别从这四个阶段进行介绍。

在描述和介绍阶段，由于项目评估成员有部分人员对 ATAM 并不熟悉，我首先介绍 ATAM 的方法。它是一种基于场景的软件架构评估方法，对系统的多个质量属性基于场景进行评估。通过该评估确认系统存在的风险，并检查各自的非功能性需求是否满足需求。客户也阐述了系统的目的和商业动机。项目是为了通过捕获互联网流量从而挖掘出有价值的漏洞信息。通过实时获取漏洞可以有效的展开防御，保证网站的安全性。客户关注系统的性能及系统能否获取高质量的漏洞信息。最后作为架构设计师的我描述了系统将要采用的 soa 架构，并将系统进行了拆分，并讲解了各个子模块的功能，初步决定系统服务端在 Linux 下使用 c 语言进行开发。

在调查分析阶段，不同的需求方基于各自的考虑都提出了各自的要求。其中客户方提出：系统的要保证其可靠性，特别是针对黑客 ip 进行跟踪的时，系统发生故障必须在 1 分钟内恢复，此优先级最高。经过自动化分析，系统对漏洞的自动识别率必须达到 90%以上，此优先级较高。系统可以对规则模块实时进行修改，其修改工作必须在 1 人天完成，以便可以根据最新的规则进行漏洞捕获。系统要确保一定的安全性。安全分析人员提出：系统需要过滤大部分正常的流量，以减轻安全分析人员的分析难度。系统必须提取出有价值的高风险 ip，无效的流量跟踪将会带来产出的低下。开发人员提出为了保证系统的开发效率及系统修改性，可以进行并行开发。经过总结我们获得了系统的质量效应树如下（考试时回简要图）。

针对这些场景我们分析了项目开发过程中的风险点、敏感点和权衡点。经过分析，该项目中存在以下风险点：黑客的 ip 如果不能实时捕获，将会丢失重要漏洞信息；系统中对消息的处理如果超过 12 小时，将会产生大量的消息积压。敏感点有：用户的加密级别、漏洞规则的修改。权衡点有：改变漏洞规则的严格程度会提升漏洞的准确率，同时带来系统性能的下降。改变系统的加密级别对系统的安全性和性能都会产生影响。

在测试阶段：经过评估小组集体讨论，确定了不同场景的优先级如下：系统的可用性最高，性能其次，可修改性及安全性优先级较低。在保证系统可用性方面，在流量捕获部分使用双机热备技术，在两个捕获系统之间设置心跳，当一台捕获系统出问题，另一台捕获设备接管。在流量自动化分析部分，采用了集群部署技术，一台分析设备出问题，不会影响整个分析系统。在保证数据安全性方面，磁盘采用企业磁盘阵列 raid5 机制。在用户数据安全性方面，采用了非对称加密及信息摘要技术。

最后形成了评估报告，经过对架构的评估，确定了系统的风险点、敏感点、权衡点和非风险点，最后以文档的形式表现。其包括的内容包括：架构分析方法文档、架构的不同场景及各自的优先级、质量效应树、风险点决策、非风险点决策及每次的评估会议记录。

该项目开发工作于 2016 年 8 月完工，系统上线后，我们的安全分析人员和客户使用该系统对互联网流量进行漏洞挖掘，一共产生了 150 种以上的 web 流量攻击流量特征和 5 个未知 web 漏洞。在国家某安全中心网研室的其他项目中起到了支撑作用，尤其是某变量覆盖漏洞、某文件写入漏洞，某 sql 注入漏洞在项目使用过程中取得了一定得效果，得到了好评。为开展互联网安全事件得防御、发现、预警和协调处置等工作提供了数据依据，更好的维护了国家公共互联网安全，保障基础信息网络和重要信息系统的安全运行。

6 论高可靠性系统中软件容错技术的应用

论高可靠性系统中软件容错技术的应用

容错技术是当前计算机领域研究的热点之一，是提高整个系统可靠性的有效途径，许多重要行业（如航空、航天、电力、银行等）对计算机系统提出了高可靠、高可用、高安全的要求，用于保障系统的连续工作，当硬件或软件发生故障后，计算机系统能快速完成故障的定位与处理，确保系统正常工作。

对于可靠性要求高的系统，在系统设计中应充分考虑系统的容错能力，通常，在硬件配置上，采用了冗余备份的方法，以便在资源上保证系统的可靠性。在软件设计上，主要考虑对错误（故障）的过滤、定位和处理，软件的容错算法是软件系统需要解决的关键技术，也是充分发挥硬件资源效率，提高系统可靠性的关键。

请围绕“高可靠性系统中软件容错技术的应用”论题，依次从以下三个方面进行论述。

1. 简述你参与设计和开发的、与容错相关的软件项目以及你所承担的主要工作。
2. 具体论述你在设计软件时，如何考虑容错问题，采用了哪几种容错技术和方法。
3. 分析你所采用的容错方法是否达到系统的可靠性和实时性要求。

找准核心论点

问题1要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题2要点：

如何考虑容错问题。

采用的哪几种容错技术和方法。

问题3要点：

采用的容错方法是否达到系统的可靠性和实时性要求。

理论素材准备

系统可靠性是系统在规定的时间内及规定的环境条件下，完成规定功能的能力，也就是系统无故障运行的概率。

系统可用性是指在某个给定时间点上系统能够按照需求执行的概率。

可靠度就是系统在规定的条件下、规定的时间内不发生失效的概率。

失效率又称风险函数，也可以称为条件失效强度，是指运行至此刻系统未出现失效的情况下，单位时间系统出现失效的概率。

冗余技术

提高系统可靠性的技术可以分为避错（排错）技术和容错技术。避错是通过技术评审、系统测试和正确性证明等技术，在系统正式运行之前避免、发现和改正错误。

容错是指系统在运行过程中发生一定的硬件故障或软件错误时，仍能保持正常工作而不影响正确结果的一种性能或措施。容错技术主要是采用冗余方法来消除故障的影响。

冗余是指在正常系统运行所需的基础上加上一定数量的资源，包括信息、时间、硬件和软件。冗余是容错技术的基础，通过冗余资源的加入，可以使系统的可靠性得到较大的提高。主要的冗余技术有结构冗余（硬件冗余和软件冗余）、信息冗余、时间冗余和冗余附加4种。

软件容错的主要方法是提供足够的冗余信息和算法程序，使系统在实际运行时能够及时发现程序设计错误，采取补救措施，以提高系统可靠性，保证整个系统的正常运行。软件容错技术主要有N版本程序设计、恢复块方法和防卫式程序设计等。

N版本程序设计：是一种静态的故障屏蔽技术，其设计思想是用N个具有相同功能的程序同时执行一项计算，结果通过多数表决来选择。其中N个版本的程序必须由不同的人独立设计，使用不同的方法、设计语言、开发环境和工具来实现，目的是减少N个版本的程序在表决点上相关错误的概率。

恢复块设计（动态冗余）：动态冗余又称为主动冗余，它是通过故障检测、故障定位及故障恢复等手段达到容错的目的。其主要方式是多重模块待机储备，当系统检测到某工作模块出现错误时，就用一个备用的模块来替代它并重新运行。各备用模块在其待机时，可与主模块一样工作，也可以不工作。前者叫热备份系统（双重系统），后者叫冷备份系统（双工系统、双份系统）。

防卫式程序设计：是一种不采用任何传统的容错技术就能实现软件容错的方法，对于程序中存在的错误和不一致性，防卫式程序设计的基本思想是通过在程序中包含错误检查代码和错误恢复代码，使得一旦发生错误，程序就能撤销错误状态，恢复到一个已知的正确状态中去。其实现策略包括错误检测、破坏估计和错误恢复三个方面。

双机容错技术：是一种软硬件结合的容错应用方案。该方案是由两台服务器和一个外接共享磁盘阵列及相应的双机软件组成。

双机容错系统采用“心跳”方法保证主系统与备用系统的联系。所谓心跳，是指主从系统之间相互按照一定的时间间隔发送通信信号，表明各自系统当前的运行状态。一旦心跳信号表明主机系统发生故障，或者备用系统无法收到主系统的心跳信号，则系统的高可用性管理软件认为主系统发生故障，立即将系统资源转移到备用系统上，备用系统替代主系统工作，以保证系统正常运行和网络服务不间断。

工作模式：双机热备模式；双机互备模式；双机双工模式。

合格范文赏析

摘要：2016 年 3 月，我所就职的国内某某知名互联网公司组织研发了一套分布式支付平台，该支付平台主要满足公司快速发展和各业务线业务流量日益增加的支付需求，用于支撑各业务线的支付功能，我有幸被定为该平台的架构设计师，主要负责架构设计工作。本文以该支付平台为例，主要论述了软件容错技术和方法在该系统中的具体应用。通过采用以集群化的形式进行应用部署；通过主备形式的数据库部署进行软件容错；通过程序设计方面进行软件的容错与避错。事实证明，以上软件容错技术的应用对于系统的可用性、安全性和可扩展性方面起到了很好的效果，满足了该系统的性能需求，并且该平台从上线到目前一直稳定运行，得到了各业务线负责人和公司高层领导的认可和赞赏。

2016 年 3 月，我所就职的国内莫某知名互联网公司组织研发了一套分布式支付平台，由于公司快速发展和各业务线业务流量的日益增加，各业务线迫切需要一个稳定健壮的支付平台，以下简称为平台，用于支撑各业务线的支付功能，公司的业务流量入口分为 pc 端、移动端、微信端和其他渠道等。我作为该平台的架构设计师，主要负责该平台的架构设计工作。

平台采用的核心架构风格为微服务的架构风格，采用 Java 语言为核心开发语言，将平台服务划分为了三类服务，核心服务，平台 web 服务、平台保障服务。其中核心服务主要分为订单服务、账户服务、网关服务、清结算对账服务、一键支付聚合服务等等，平台 web 服务主要提供与用户对接的界面等，平台保障服务主要包括例如 JOB 框架，平台报警服务，MQ 服务等等。对这些服务中的核心服务、平台 web 相关的服务必须要能满足 7*24 小时的稳定运行，对软件的可靠性要求非常高，所以在该平台中的这些核心服务就必须具备一定的容错能力，在某个服务运行时出错的情况下不能影响到整个集群中的服务，这就要求在软件架构设计中必须考虑到软件容错技术的应用。

提高软件系统可靠性技术主要分为容错技术和避错技术，容错技术的主要方式为冗余，冗余又分为结构冗余、时间冗余、信息冗余，冗余附加。结构冗余又分为静态冗余、动态冗余和混合冗余。软件容错技术主要有 N 版本程序设计，恢复块方法，和防卫式程序设计。结合互联网软件的性质，我主

要采用了集群技术、数据库主从方式、和程序设计方面来进行软件的容错与避错处理。下面就从以上三方面详细讨论我所采用的容错技术和方法。

1、通过集群技术来容错

平台中的各服务如果在运行时部署在一台服务器上，那么当服务器发生故障，整个平台将不能再提供任何服务。所以一般非常小规模的应用才会采取这样的部署方式，像互联网应用这样的支付平台来说必须采用多机同时部署的方式，防止单台服务器宕机或者服务进程 Crash 导致整个平台不能提供服务的问题。通过多机同时部署，当一台服务出现问题时，可以很容易替换一台新服务器进行重新部署生效，通过服务客户端的软负载均衡功能，可动态剔除不可用服务机器，动态发现新加入集群的服务机器，使平台在出现故障时可平滑过渡，达到容错的目的。另外平台中各服务当首次获取到的不易变的静态数据会将其存入非本地缓存中，例如采用了 rediscluster 技术，可以很好的保证写入缓存中的数据获取的高可靠性，恰当使用缓存不但会提升平台性能，同时还可以起到容错的效果，例如当某个服务所依赖的后端存储发生了短时的故障或者网络抖动，在这个时候大量的并发请求发现存储获取失败直接从缓存中获取将数据返回给调用方，起到了很好的容错效果。

2、通过数据库主从部署方式来容错

对于该平台来说，所依赖的后端数据库存储的稳定性是非常非常重要的，所有的订单，交易、账户等等数据将直接存储到数据库中。如果数据库在运行过程中频繁宕机，那么带来的问题将是不能容忍的，因为会造成订单丢失，交易丢失，账户余额出错等等问题，所以在这里就要求数据库存储要具有非常高的可靠性，同时具有很强的容错性，在这里我主要采用了数据库的部署结构为主从式方式，要求部署在不同服务器上，在不出问题的情况下对于一些时效性要求不是很高的场合从库可以负责承担一部分读流量，当主库发生读写问题时，可快速由其他的从库升级为主库，继续服务，达到容错的效果。在这里我还采用了要求数据库宕机加报警的方式来防止宕机的主从数据库实例过多，导致在并发高的情况下没有可用的从库升级为主库提供服务，通过这样的方式也提高了整个平台的高可靠性。数据库的数据文件存储这里我也要求采用了 RAID 磁盘容错技术来防止单块磁盘损坏导致的数据文件丢失问题。

3、通过程序设计方面进行软件的容错与避错。

根据以往的架构经验，那么系统的不可靠大部分是由于程序内部的设计或者网络请求参数的配置或者连接池参数的配置不当所导致的。所以通过程序设计方面进行软件的容错是非常重要的。在程序设计方面的容错用的最普遍的就是防卫式程序设计，例如平台中的一键支付聚合服务，当在支付的过程中调用账户服务来进行账户金额扣减的时候，势必会调用账户服务传递请求对象来处理，如果说账户服务在被调用的这一刻网络抖动或者丢包的情况下，这个时候一键支付聚合服务必然会收到抛出的错误信息，如果没有通过恰当的容错处理，那么这次一键支付必然会给用户显示支付失败，不太友好，在这里我采用了 TRYCATCH 机制加 3 次重试的容错处理机制，就解决了该次支付因网络抖动导致的支付失败问题。平台采用的是微服务的架构风格，那么在服务之间的通讯过程中涉及到数据的传递，这里我采用了在数据传输协议的头部加 CRC 码来做到对错误数据处理的避错。

通过采用了以上容错技术的方法和措施后，平台从上线运行到目前为止，各服务运行状态良好，通过日志分析来看，支付成功率很高，得到了公司高层领导和各业务线负责人的赞赏和认可。但在我看来还有如下两点不足。1. 各服务间调用事物一致性问题的容错处理。针对该问题，目前只能保证事物的最终一致性，因为根据 CAP 理论，要解决该问题确实存在一定的难度，后面我准备研究下 TCC 事物处理方式尤其适合支付平台场景，争取在不损失性能的前提下最大限度解决分布式事物的一致性问题。2. 目前所采用的最大努力推送型事物服务依赖 MQ 重复消息的问题。针对该问题我采用加了一张消息处理表的方式来解决，当收到消息的时候，先查询该条消息是否已经处理，如果没有处理直接进行处理并将其进行记录，防止重复处理导致支付数据出错。

软件容错技术对软件的稳定性起着至关重要的作用，尤其是针对互联网性质的软件并发高存在流量峰值等问题，软件容错技术的应用的重要性就不言而喻了。经过这次我所采用的软件容错技术的方法和措施的实施效果后，使我也看到了自己身上的不足之处，我会在今后的架构设计过程中，不断更新自己的知识，不断完善自己的架构设计领域，设计出更好的软件架构，更好的支撑业务平台的运行，提高公司的竞争力，为公司为社会尽一份绵薄之力。

7 论信息系统的安全架构设计

论信息系统的安全架构设计

在企业信息化推进的过程中，需要建设许多的信息系统，这些系统能够实现高效率、低成本的运行，为企业提升竞争力。但在设计和实现这些信息系统时，除了针对具体业务需求进行详细的分析，保证满足具体的业务需求之外，还要加强信息系统安全方面的考虑。因为如果一个系统的安全措施没有做好，那么系统功能越强大，系统出安全事故时的危害与损失也就越大。

请围绕“信息系统的安全性及保密性”论题，依次从以下三个方面进行论述：

1. 概要叙述你参与分析设计的信息系统及你所担任的主要工作。
2. 深入讨论作者参与建设的信息系统中，面临的安全及保密性问题，以及解决该问题采用的技术方案。
3. 经过系统运行实践，客观的评价你的技术方案，并指出不足，以及解决方案。

找准核心论点

问题1要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题2要点：

参与建设的信息系统中面临的安全及保密性问题。

采用的技术方案。

问题3要点：

评价该技术方案。

指出该方案不足并提出解决方案。

理论素材准备

信息安全技术：对称加密，非对称加密，数字信封，信息摘要，数字签名，数字证书。

网络安全技术：防火墙、入侵检测、计算机病毒和木马的防护。

合格范文赏析

摘要：2015 年 3 月，我所在公司承接了一款养老管理信息平台，该系统以养老为主线，其中包括养老档案、照护计划、服务审计、状况跟踪、费用管理等方面的 60 多个业务功能模块，我在项目中担任系统架构设计师，主要负责整个系统的架构设计。本文以该养老管理信息平台为例，主要论述了针对该系统的安全及保密性问题，我们所采用的技术手段及解决方案。在网络硬件层通过设置硬件防火墙，来保证内部网络安全性；在数据层通过设置数据加密及容灾备份机制，以此保证数据抗突发风险的安全；在应用层统一采用 RBAC 的授权机制，来保证整个系统认证环节的安全性。事实证明，以上技术方案的实现对整个系统的安全和保密性方面起到了很好的效果，最终项目顺利上线，运行稳定，受到广大用户的一致好评。

目前我国已经进入到老龄化社会，老龄人口逐年增长，按照老龄办提供的数字，预计到 2020 年中国的老年人口将要达到 2.48 亿，与之增加的养老消费人均三千元左右，从整个养老产业的规模来看，估算在 2025 年要增加到五万亿规模，市场前景巨大。随着互联网的迅猛发展，各行各业都在进行着互联网+的尝试，希望搭上这个发展契机。其中，养老领域更迫切需要解决养老专业化程度低，信息化不足，健康照护水平滞后等一系列亟待解决的问题。

2015 年 3 月，我所在公司承担了全国老龄办及全国几十家养老和医疗机构合作进行的养老管理信息平台的开发工作，我在该项目中担任系统架构设计师，主要负责应用系统的软件架构设计。由于我们公司在医疗行业领域有着丰富的成功经验，同时，近些年在养老领域也成功实施过很多成熟的案例，所以，一期投资方出资 3000 万，委托我们进行这款综合性养老管理平台的开发工作。该系统以养老为主线，其中包括养老档案，照护计划，服务审计，状况跟踪，费用管理，决策支持等方面的 60 多个业务功能模块组成，系统功能相当完备。

经过前期对全国几十家养老机构和相关合作的医疗单位的调研分析，结合原有的经验，对整个系统业务进行详细规划和相关设备的初步选型，即我们内部所说的“两版三端”的方案，机构养老和社区养老两个子系统，同时能够在 PC 端，PAD 端，手机端三端进行呈现和交互。整个养老管理信息平台基于 B/A/S 模式的分层架构风格设计。同时，结合当前先进的基于租户理念搭建的一个养老云平台（SAAS）。当前信息安全问题严重，而养老云平台是一个养老大型综合管理平台，确保系统的安全性和保密性显得尤为重要，让使用我们平台的每一家机构、社区、老人及相关亲人都能够安心和放心。这样，只有最终得道了用户的信赖，系统的前景才会发展的更好。所以，我在系统的安全性和保密性方面做了充分的设计。

1、网络硬件层安全方案

首先，因为网络和硬件是整个系统运行的基础，也是很多外部攻击的主要途径，所以，在这块需要进行合理规划。我们对网络拓扑结构划分为外部网络，内部网络和 DMZ 三部分。在外部网络和内部

网络之间设置了硬件防火墙，主要是防止外部的恶意攻击。在防火墙后，为了加强对病毒入侵的防范，又设置了硬件的防毒墙，实时保证最新的病毒特征库保证有效的病毒工具。为了防止外部客户通过 DNS 服务直接访问到我们服务器的目标地址，在内部网络中又增加了反向代理服务器，对外只暴露代理服务器的虚拟 IP，更好的保护了应用服务器被攻击的可能性。应用服务器和数据库服务器做了物理隔离，内部人员也无法通过 IP 直接访问，只能通过跳板机由服务器管理员进行操作，阻断了外界通过内部客户端代理的访问对服务器造成攻击。对于一些核心的 FTP 服务器，DNS 服务器，Web 服务器都规划到 DMZ 中。

2、数据层安全方案

其次，数据是整个系统平台的核心，其中涉及大量个人隐私数据和重要信息资产，安全不容忽视。主要从数据存储，数据访问和数据容灾几方面进行了规划。由于系统主要存放的养老档案数据，其中涉及很多个人隐私部分数据，为了防止泄露，我们在存储时进行了加密处理。虽然在处理中损失些性能，但是，提高了数据的安全性和保密性。我们采购商业数据库 Oracle 作为后台存储，由于其极佳的商业口碑，对数据存储的安全提供强大保证。为了防止数据信息的泄露，加强了访问权限的限制，在数据库的访问权限上，都根据不同角色进行了详细划分，包括对数据库，表，索，记录等的增，删，改，查进行了限制，严格保证非法用户对数据库恶意破坏。同时，在数据库本身，也制定了基于全量，增量，差量的按周备份计划，保证每时每刻的数据都可以及时恢复。在物理存储上也做了本地多机房的容灾备份，充分保证数据抗突发风险的安全。

3、应用层安全方案

最后，由于我们系统架构采用的是分层架构风格，同时，又是采用“三端”的形式，即手机端，PAD 端，还有 PC 端，所以，在用户认证，接口传输等方面做了充分设计。由于系统涉及到养老机构工作人员，老人及家属等，角色众多，统一采用 RBAC 的授权机制，既增强系统安全性满足业务需求，同时，减轻由于权限信息维护负担。在登录界面设计上，不仅仅需要提供用户名+口令，同时设置了校验码，而且同时还增加了动态口令，通过短信接收验证码，以此来防止暴露非法破解登录，保证整个系统认证环节的安全性。密码提交和存储过程中，一律通过 MD5+salt 加密，即使密码泄露也不会被解密。接口设计是整个系统通信的安全保障，为了防止外部人员恶意调用和窃取信息，我们采用了令牌机制保障每次通讯的安全性，即在登录后，通过加密传输的用户信息和时间戳获取到 token 令牌，然后在后续每次通讯传输过程服务端对 token 进行校验，充分保障了接口调用的安全。

整个项目历时 10 个月开发完成，到目前运行稳定。通过用户一段时间的使用反馈，不论是 PC，还是 PAD 和手机端，在任何环境下使用系统，基本没有信息泄露的情况发生。但是，一些用户反馈，手机登录需要时间较长，查询和保存老人档案信息有时候需要等待，经过分析，登录时间较长，由于登录需要对客户端对用户和密码信息进行客户端加密处理，加密算法耗费了移动端 CPU 资源，同时也在服务端也对数据进行加密处理返回客户端 token 也消耗了不少时间。老人档案也是由于敏感信息加密的问题，耗费了 CPU 的处理时间，我们后面通过选择效率更高的加密算法改善了这个问题。

实践证明，项目能够顺利上线，并运行稳定，使用良好，与系统在安全性和保密性方面设计密不可分。系统安全是一个永久的话题，我们对系统安全性的完善是一个持续的过程，我们接下来，还会继续不断完善系统安全方面的设计缺陷和不足，使整个养老平台更加安全好用。

8 论企业应用集成

试题：论企业应用集成

企业应用集成(Enterprise Application Integration, EAI) 是完成在组织内、外的各种异构系统，应用和数据源之间共享和交换信息和协作的途径，方法学，标准和技术。企业应用集成所连接的应用包括各种电子商务系统，企业资源规划系统，客户关系管理系统，供应链管理系统，办公自动化系统，数据库系统，数据仓库等。

请围绕“企业应用集成”论题，分别从以下几个方面进行论述：

- 1、简要叙述你参与的企业应用集成项目（项目的背景、发起单位、目的、项目特点等）。
- 2、简要叙述企业应用集成的四个层次（方法）？
- 3、详细论述你参与的项目是采用了哪个层次的集成，如何实施的，效果如何？

找准核心论点

问题 1 要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题 2 要点：

企业应用集成的四个层次。

问题 3 要点：

采用了哪个层次的集成。

具体实施过程以及应用效果。

理论素材准备

1. 表示集成

表示集成也称为界面集成，这是比较原始和最浅层次的集成，但又是常用的集成。这种方法把用户界面作为公共的集成点，把原有零散的系统界面集中在一个新的界面中。表示集成是黑盒集成，无需了解程序与数据库的内部构造。常用的集成技术主要有屏幕截取和输入模拟技术。表示集成通常应用于以下几种情况：

(1) 在现有的基于终端的应用系统上配置基于 PC 的用户界面。

(2) 为用户提供一个看上去统一，但是由多个系统组成的应用系统。

(3) 当只有可能在显示界面上实现集成时。表示集成的实现是很简单的，也是很彻底的，只是做了一层“外装修”，而额外多出来的集成界面也将可能成为系统的性能瓶颈。

2. 数据集成

为了完成控制集成和业务流程集成，必须首先解决数据和数据库的集成问题。在集成之前，必须首先对数据进行标识并编成目录，另外还要确定元数据模型，保证数据在数据库系统中分布和共享。因此，数据集成是白盒集成。有很多不同的中间件工具可以用于数据集成。例如，批量文件传输，即以特定的或是预定的方式在原有系统和新开发的应用系统之间进行文件传输；用于访问不同类型数据库系统的 ODBC 标准接口；向分布式数据库提供连接的数据库访问中间件技术等。通常在以下情况下，将会使用数据集成：

(1) 需要对多种信息源产生的数据进行综合分析和决策。

(2) 要处理一些多个应用程序需要访问的公用信息库。

(3) 当需要从某数据源获得数据来更新另一个数据源时，特别是它们之间的数据格式不相同时。相对而言，数据集成比表示集成要更加灵活。但是，当业务逻辑经常发生变化时，数据集成就会面临困难。

3. 控制集成

控制集成也称为功能集成或应用集成，是在业务逻辑层上对应用系统进行集成的。控制集成的集成点存于程序代码中，集成处可能只需简单使用公开的 API 就可以访问，当然也可能需要添加附加的代码来实现。控制集成是黑盒集成。

实现控制集成时，可以借助于远程过程调用或远程方法调用、面向消息的中间件、分布式对象技术和事务处理监控器来实现。控制集成与表示集成、数据集成相比，灵活性更高。表示集成和数据集成适用的环境下，都适用于控制集成。但是，由于控制集成是在业务逻辑层进行的，其复杂度更高一些。而且，很多系统的业务逻辑部分并没有提供 API，这样，集成难度就会更大。

4. 业务流程集成

业务流程集成也称为过程集成，这种集成超越了数据和系统，它由一系列基于标准的、统一数据格式的工作流组成。当进行业务流程集成时，企业必须对各种业务信息的交换进行定义、授权和管理，以便改进操作、减少成本、提高响应速度。业务流程集成不仅要提供底层应用支撑系统之间的互连，同时要实现存在于企业内部的应用之间，本企业和其他合作伙伴之间的端到端的业务流程的管理，它包括应用集成、B2B 集成、自动化业务流程管理、人工流程管理、企业门户，以及对所有应用系统和流程的管理和监控等。

合格范文赏析

摘要：

2016 年 9 月，我国某省移动通信有限公司决定启动 VerisBilling6.0 项目，该项目实现了在线计费、离线计费、内容计费、账务处理、信控管理等子系统的整合，我作为系统分析师全程参与了项目的建设。本文以 VerisBilling6.0 项目为例，论述企业应用集成中界面集成、数据集成、应用集成在软件集成过程中的实际应用及效果。通过界面集成，实现了账务前台、产品前台、信控前台界面的整合，把几个独立系统经过集成也一个整体展现给用户。通过数据集成，将各之前各系统产生的“信息孤岛”进行了整合，数据的一致性得到有效保障。通过应用集成，从业务逻辑上为各功能系统提供

统一的数据接口，使业务逻辑更规范。通过以上集成技术的应用，项目于 2017 年 4 月成功上线，各项性能指标达到客户要求，获得省移动通信公司各级领导的好评。

正文：

近几年来某省移动用户增长至 3000 多万，随着移动数据流量资费的新一轮下调，导致 GPRS 数据流量成爆发式增长，OpenBillingNG 版系统在话单处理上瓶颈显现。16 年春节期间，GPRS 日话单达到 30 亿条，话单处理处于积压状态，直到节后两周才将积压话单追完，大量跨月的话单引发了大批用户投诉，给移动业务支撑中心带来的压力非常大；该省移动通信公司相关领导联合系统运营商遂召开会议讨论解决方案，最终决定将该省 OpenBillingNG 版升级至 VerisBilling6.0 版本，以解决 OpenBillingNG 版本遇到的瓶颈问题。作为移动通信 BOSS 业务支撑的核心，VerisBilling6.0 需支持 24x7 连续运行，满足话单的实时处理，还需要把在线计费、离线计费、内容计费、账务处理、产品管理等 OpenBillingNG 版时独立的系统进行整合。我作为系统分析师全程参与了 VerisBilling6.0 项目的建设，VerisBilling 6.0 项目由产品管理组、研发组、测试组、对账组、运维组、数据组、专家组共 120 人组成的项目团队，耗时 8 个月完成，项目从 2016 年 9 月初启动，至 2017 年 4 月 30 日上线。

作为系统分析师，我深知在 VerisBilling6.0 项目中系统集成方法对该项目的重要性。通常情况下，企业应用集成中常用的集成有界面集成、数据集成、应用集成等方法。其中界面集成通过将各系统界面进行整合，从而实现了为用户提供一个统一的系统界面的效果，增强了各系统间的交互性能。数据集成的集成点在数据访问层，通过中间件更新数据库的方式，保持数据一致；通过对数据库中通常需要同步的数据表的集成，实现各系统的数据高效同步，保证了系统数据的一致性。应用集成的集成点都在程序的内部结构中，需要根据业务的实际情况，重组结构，重新开发；是基于业务逻辑层面的集成方法，通过对系统业务逻辑进行集成，为各系统提供统一的业务接口，属于较高层次是集成方式，也是难度较大的集成。三种集成方法相辅相成，互为补充。

如何为 VerisBilling6.0 项目选择合适的集成方法呢？首先，作为 BOSS 系统的核心，Veris Billing6.0 项目是一个庞大、复杂的项目，涉及账务系统前台、产品管理系统前台、信控管理系统前台的界面集成。其次，VerisBilling6.0 项目还涉及计费产品库、账务产品库、信控系统数据库等数据库的整合。最后，项目还涉及在线计费、离线计费、内容计费等几个系统业务逻辑方面的集成。接下来我将从界面集成、数据集成、应用集成三个方面来具体阐述，在 VerisBilling6.0 项目中，我的团队是如何使用这些方式实现企业应用集成的。

界面集成的应用。在 VerisBilling6.0 项目中，我们实现了账务系统前台、产品管理前台、信控管理系统前台进行整合。首先，我们对信控系统页面重新做了布局，将账务系统前台、产品管理前台的功能页合并到信控管理系统的主界面上，并增加了相应的链接功能菜单。其次，在做界面集成时，我们充分的考虑了系统界面的用户友好性，对几个界面做了扁平化设计，并将几个系统的界面样式风格调整一致，使得集成后的系统看上去更像一个整体。最后，我们增加了系统导航功能菜单，完善了系统间的交互性能。经过界面集成，三个系统界面被集成到一个系统页面上，形成了一个整体。当用户登录信控管理系统后台就可以对账务管理系统、产品管理系统、信控管理系统进行操作。由于三个

系统都存在独立的系统表，且都实现了自身的权限管理等功能，要想实现单点登录后进行各系统功能的操作，还需要对用户数据进行集成。

数据集成的应用。在 OpenBillingNG 版本中，存在账务产品库、计费产品库、局数据产品库、信控数据库等，在以往的生产过程中，经常会因数据变动后同步不及时而产生错误引发用户投诉。VerisBilling6.0 项目要求将所有产品库进行整合，合并到一个中心数据库，首先，我们对各功能系统的数据库进行分析，将各库中的数据表进行梳理比对，并将平时需要做同步处理的表提取出来分析。接下来，将提取出来的数据表合并到公共数据库，实现了将分散的数据信息进行整理合并。最后，将那些系统间无直接关系的表直接割接到公共数据库，经过梳理加工后，完成了对数据的集成，实现了数据的统一管理，数据的一致性得到了保障。经过数据集成，可以有效避免“信息孤岛”的产生，由于系统之间直接调用公共数据表的数据，使得系统数据在任何时候都是完整的、一致的，有效避免数据同步不及时的问题。

应用集成的应用。VerisBilling6.0 项目中，大量的外围系统需要通过接口访问计费 MDB 和账务 MDB。由于这个版本中，MDB 数据表的变动非常大，为了保障 CRM 等外围系统对 MDB 的访问不受影响，首先，我们对所有的 MDB 接口进行了梳理并同外围系统研发人员进行了确认。接下来，针对每个接口，我们进行了重新定义，并将设计文档发与给外围系统研发负责人，然后按设计文档要求开发出相应的接口。最后，在系统测试过程中，要求各外围系统参与联调测试。由于接口的修改涉及到系统业务逻辑的调整，应用集成难度极大，对外围系统的影响面较广，在该集成方法的使用过程中，参与的人员最多，联调周期最长。应用集成对于各方参与研发的人员综合素质要求也比较高，需要充分考虑逻辑业务的变化对系统的性能的影响，对任何一个接口的疏忽都会产生较为严重的后果。

通过以上集成技术的应用，VerisBilling6.0 项目于 2017 年 4 月底上线，经过半年的运行，系统各项性能指标达到可以要求，并通过客户验收，获得省移动通信公司各级领导好评。在项目结束后的讨论会上，大家也指出了项目中存在一些不足。在项目中由于 MDB 发生了变动，所以需要业务逻辑的调整，我们在项目中要求所有外围系统都需要参与联调测试，但有几个接口 CRM 没有按要求进行相应的调整，导致系统上线当天出现 CRM 访问 MDB 出现异常。

通过 VerisBilling6.0 项目，使我深刻的认识到在项目实施过程中，每一个细节都需要把控好。首先，在项目中需让专家团队及时对风险进行评估，并针对每个风险点需要提供相应的应对措施，这些措施在项目出现问题时能得到有效的处置。其次，需要制定好实施计划，并严格按计划进行实施，特别是一些需要做联调测试的内容上，必须严格按照要求完成，避免出现联调测试不完整这类似问题。最后，在项目中一定要严格把控好每个细节，并实现将系统每个细节的把控落实到个人。

9 基于构件的软件开发

试题：基于构件的软件开发

软件系统的复杂性不断增长、软件人员的频繁流动和软件行业的激烈竞争迫使软件企业提高软件质量、积累和固化知识财富，并尽可能地缩短软件产品的开发周期。

集软件复用、分布式对象计算、企业级应用开发等技术为一体的“基于构件的软件开发”应运而生，这种技术以软件架构为组装蓝图，以可复用软件构件为组装模块，支持组装式软件的复用，大大提高了软件生产效率和软件质量。

请围绕“基于构件的软件开发”论题，依次从以下三个方面进行论述。

1. 简述你所参与开发的运用了构件技术的项目，以及你所担任的工作；
2. 论述你在项目中如何运用构件技术来进行软件开发；
3. 分析并讨论各种构件技术的优点、缺点，并展望构件技术的发展趋势。

找准核心论点

问题 1 要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题 2 要点：

如何运用构件技术来进行软件开发。

具体实施过程。

问题 3 要点：

各种构件技术的优点、缺点。

展望构件技术的发展趋势。

理论素材准备

构件是一个独立可交付的功能单元，外界通过接口访问其提供的服务。

构件由一组通常需要同时部署的原子构件组成。一个原子构件是一个模块和一组资源。原子构件是部署、版本控制和替换的基本单位。原子构件通常成组地部署，但是它也能够被单独部署。

构件和原子构件之间的区别在于，大多数原子构件永远都不会被单独部署，尽管它们可以被单独部署。相反，大多数原子构件都属于一个构件家族，一次部署往往涉及整个家族。

一个模块是不带单独资源的原子构件（在这个严格定义下，Java 包不是模块——在 Java 中部署的原子单元是类文件。

一个单独的包被编译成多个单独的类文件——每个公共类都有一个）。

模块是一组类和可能的非面向对象的结构体，比如过程或者函数。

构件和对象

构件的特性是：（1）独立部署单元；（2）作为第三方的组装单元；（3）没有（外部的）可见状态。一个构件可以包含多个类元素，但是一个类元素只能属于一个构件。将一个类拆分进行部署通常没什么意义。

对象的特性是：（1）一个实例单元，具有唯一的标志。（2）可能具有状态，此状态外部可见。（3）封装了自己的状态和行为。

构件接口

接口标准化是对接口中消息的格式、模式和协议的标准化。它不是要将接口格式化为参数化操作的集合，而是关注输入输出的消息的标准化，它强调当机器在网络中互连时，标准的消息模式、格式、协议的重要性。

面向构件的编程（COP）

关注于如何支持建立面向构件的解决方案。“面向构件的编程需要下列基本的支持：

- 多态性（可替代性）；
- 模块封装性（高层次信息的隐藏）；
- 后期的绑定和装载（部署独立性）；
- 安全性（类型和模块安全性）。

构件技术就是利用某种编程手段，将一些人们所关心的，但又不便于让最终用户去直接操作的细节进行了封装，同时对各种业务逻辑规则进行了实现，用于处理用户的内部操作细节。

目前，国际上常用的构件标准主要有三大流派，分别是 COM/DCOM/COM+、CORBA 和 EJB。

EJB(Enterprise Java Bean)规范由 Sun 公司制定，有三种类型的 EJB，分别是会话 Bean(Session Bean)、实体 Bean(Entity Bean)和消息驱动 Bean(Message-driven Bean)。

EJB 实现应用中关键的逻辑，创建基于构件的企业级应用程序。EJB 在应用服务器的 EJB 容器内运行，由容器提供所有基本的中间层服务，如事务管理、安全、远程客户连接、生命周期管理和数据库连接缓冲等。

COM、DCOM、COM+：COM 是微软公司的。DCOM 是 COM 的进一步扩展，具有位置独立性和语言无关性。COM+并不是 COM 的新版本，是 COM 的新发展或是更高层次的应用。

COBRA 标准主要分为三个层次：对象请求代理、公共对象服务和公共设施。

最底层是对象请求代理 ORB，规定了分布对象的定义（接口）和语言映射，实现对象间的通讯和互操作，是分布对象系统中的“软总线”；

在 ORB 之上定义了很多公共服务，可以提供诸如并发服务、名字服务、事务（交易）服务、安全服务等各种各样的服务；

最上层的公共设施则定义了组件框架，提供可直接为业务对象使用的服务，规定业务对象有效协作所需的协定规则。

合格范文赏析

摘要：本文以我主持开发的某公司生产经营管理系统为例，探讨了基于构件的软件开发问题。该系统是一个集原料采购、生产管理、物流管控等七大功能于一体的综合信息系统，在该系统的开发过程中，我担任系统架构师角色，主要负责需求分析、系统建模和方案设计三个方面的工作。本文首先简要分析了 CORBA、EJB、COM/DCOM 三种构件技术的特点，然后着重论述了采用构件技术进行软件开发的过程。在构件的获取阶段，我们采用了三种构件获取方式来解决用户提出的 3 类不同的需求；在构件的开发阶段，我们统一采用一个查询构件进行了封装，以实现将同一功能的不同表现封装到一个独立的构件中；在构件组装阶段，采用了 3 种构件组装方式完成了构件的组装。最终项目顺利上线并运行稳定，获得用户一致好评。

2013年5月，我所在公司承接了某大型粮食加工企业某公司生产经营综合管理系统的开发工作。该系统是某公司在国家粮食案例政策的指导下，结合自身经营管理需求提出建设的。其目的是对内加强管理，对外提升服务，以实现提升品牌形象、保护消费者利益的战略目标。系统整体上分为两个部分，一是经营管理WEB平台，二是手机APP应用。系统采用了基于服务的层次架构，共分为三层。其中用户界面层使用Extjs、senchatouch和phonegap框架实现，业务服务层使用.Net平台实现，数据层使用IBMDB2V9.5。该系统于2013年5月开始建设，2014年6月上线运行，历时一年。在系统的开发过程中，我担任系统架构师角色，负责需求的获取和分析、系统建模和总体方案设计工作。在系统的实现过程中，我和我的团队使用了基于构件的软件开发技术，收到了很好的效果。

基于构件的软件开发是在面向对象技术的基础上发展起来的，它是软件危机问题日益突出形势下的产物，它有效地解决了软件系统复杂度、成本、质量、效率等难以控制的问题，受到业界的广泛推崇。当前主流的软件构件技术有三大流派，分别是OMG的CORBA、SUN的EJB和微软的COM技术。上述三种技术中，CORBA实现的是最为漂亮的，它分为对象请示代理、公共对象服务和公共设施三个层次，其特点是大而全，互操作性和开放性特别好，其缺点是庞大且复杂，相关技术和标准更新缓慢。相比之下，EJB的发展要更加迅速，它是在JAVA语言基础上建立的服务器端组件模型，具有优秀的跨平台性。EJB框架提供了远程访问、安全、持久化和生命周期等多种支持分布式计算的服务，目前JAVA和CORBA有融合的趋势。最后是COM技术，它是微软公司的独家产品，基于windows平台，功能强大、效率很高，且有一系列相应的开发工具支持，其最大的弱点是跨平台性较差，很多人认为CORBA将比COM技术走地更远。基于构件的软件开发技术能有效地简化设计、提高效率、保证质量，某公司生产经营管理系统的开发项目时间紧、任务重，我们使用基于构件的软件开发技术解决了软件开发中的各种问题，取得了很好的效果。

具体在实践过程中，我们开展了模块划分、构件标识、构件获取、构件组装与测试、构件管理等活动，这里重点谈一下“构件获取”、“构件开发”和“构件组装”三个方面的问题。

一、构件获取

在本系统的开发过程中，用户提出的需求从实现方式上可分为三类。第一类需要修改当前的系统来实现，比如用户提出，需要将某公司当前在用的两套生产控制系统的支行状态集成到目标系统中；第二类是直接使用我公司现成的构件来实现。比如用户登录、角色权限管理、日志记录、数据库访问等基本功能；第三类需要集成第三方的服务来实现，比如用户要求APP软件需要具有消息推送、GPS定位功能，可通过集成第三方运营商的产品来实现，无需另行开发。针对上述三种类型的需求，我们采用了三种构件获取方式。首先是通过改造现有系统获得构件，我们与生产控制系统的开发商联系，请他们使用COM技术对软件的工作状态进行封装，并开放接口。其次是使用构件库中现成的构件，我公司在长期的软件项目建设过程中，积累了大量可重用的软件构件，如文件序列化、数据库连接等模块，这些构件可直接在新项目中使用。最后是集成第三方的构件，针对用户提出的消息推送和GPS定位功能，我们经过对比和测试，选择了个百度的产品进行集成。此外我们还根据需求，重新开发了一些功能构件，以支撑用户的需求。

二、构件开发

构件的优势体现在其粗粒度的重用性，因此在构件的设计过程中，应尽可能将同一功能的不同表现封装到一个独立的构件中，以保持其高内聚、低耦合的特性，本系统的构件设计就很好地遵循了这一原则。以数据查询构件为例，整个系统中用户需要查询的数据多种多样，对分页显示的要求不尽相同，数据返回格式也不完全一致，可能是 XML，也可能是 JSON，针对所有这些查询需求，我们统一用一个查询构件进行了封装，开发人员只需要构造好 SQL 语句，再配合一些特定的参数，就能得到自己想要的结果，这样最大限度地保证了构件的可重用性和重用粒度。为了实现这一目标，有时还需要用到一些经典的设计模式。比如我们使用的数据库连接构件，就用到了抽象工作方法，构件可根据配置文件的内容，在运行过程中建立对不同数据库、不同方式的连接，很好地体现了构件的优势。

三、构件的组装

不同的构件类型，需要采用不同的组装方式。在本系统的开发过程中，我们用到了以下三种方式。首先是 DCOM 构件的组装，采用了远程调用方式，该方式主要用于对生产控制系统构件的组装。其实现比较复杂，需要在控制系统上部署 DCOM 服务，还得在 WEB 平台的 IIS 上配置相应的权限。其次是构件库中构件的组装，该方式比较简单，直接引用构件库中的产品文件，可能是 DLL 格式，或者是 C# 的源码文件。最后是 SOA 服务调用组装，主要针对第三方的软件服务，其实现也很简单，使用基于 http 的 webservice 访问即可。需要注意的是在这种组装方式中，由于数据需要经过第三方的软件平台进行传输，需要采用一定的数据加密措施，以提高系统的安全性。

由于使用了基于构件的软件开发方式，某公司生产经营管理系统的开发工作进行地十分顺利，系统按期上线，并得到了用户的肯定。该项目的成功让我认识到好的软件设计思想和技术在软件开发过程中的作用和价值，坚定了我对基于构件的软件开发技术的信心。从软件行业的发展来看，从汇编语言、基于过程的软件开发、面向对象的软件开发，直到现在基于构件、面向服务的软件开发，我们可以认识到软件元素在两个维度上的进化趋势，即内部功能越来越强大、全面，对外的接口却越来越简单、标准，终有一日各领域的软件必将能在一个统一的标准下进行无缝的组装，届时面向协作的软件开发、基于职能的软件开发等新技术都可能会出现，上层应用功能的实现也将变得异常简单，计算机软件可能会无所不在，数字化生活、智能地球等等现在还比较超前的概念，将在那里得以实现。我想这一天值得每一位软件从业人员为之努力、期待。

10 论企业集成平台的技术与应用

论企业集成平台的技术与应用

企业集成平台是一个支持复杂信息环境下信息系统开发、集成和协同运行的软件支撑环境。它基于各种企业经营业务的信息特征，在异构分布环境（操作系统、网络、数据库）下为应用提供一致的信息访问和交互手段，对其上运行的应用进行管理，为应用提供服务，并支持企业信息环境下各特定领域的应用系统的集成。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。

请以“企业集成平台的技术与应用”为题，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企业集成平台相关的软件项目以及你在其中所担任的主要工作。

2. 简要说明企业集成平台的基本功能及企业集成的关键技术，并结合项目实际情况，阐述该项目所选择的关键技术及其原因。

3. 结合你具体参与管理和开发的实际项目，举例说明所采用的企业集成架构设计技术的具体实施方式及过程，并详细分析其实现效果。

找准核心论点

问题1要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题2要点：

企业集成平台的基本功能及企业集成的关键技术。

结合项目实际情况，阐述该项目所选择的关键技术及其原因。

问题3要点：

采用的企业集成架构设计技术的具体实施方式及过程。

详细分析其实现效果。

理论素材准备

企业集成分类

(1) 表示集成：即界面集成，是最原始的集成，黑盒集成。将多个信息系统的界面集成在一起，统一入口，为用户提供一个看上去统一，但是由多个系统组成的应用系统的集成，例如桌面。

(2) 数据集成：白盒集成，把不同来源、格式、特点性质的数据在逻辑上或者物理上有机的集中，从而为企业提供全面的数据共享。如数据仓库。

(3) 控制集成（功能集成、应用集成）：黑盒集成，业务逻辑层次的集成，可以借助于远程过程调用或远程方法调用、面向消息的中间件等技术，将多个应用系统功能进行绑定，使之像一个实时运行的系统一样接受信息输入和产生数据输出，实现多个系统功能的叠加。如钉钉。

(4) 业务流程集成：即过程集成，最彻底的、综合的集成，这种集成超越了数据和系统，由一系列基于标准的、统一数据格式的工作流组成。当进行业务流程集成时，企业必须对各种业务信息的交换进行定义、授权和管理，以便于改进操作、减少成本、提高响应速度。它包括应用集成、B2B集成、自动化业务流程管理、人工流程管理、企业门户，以及对所有应用系统和流程的管理和监控等。如电子购物网站-第三方支付平台-银行-物流等流程集成。

应用集成数据交换方式

共享数据库：在应用集成时，让多个应用系统通过直接共享数据库的方式，来进行数据交换，实时性强，可以频繁交互，属于同步方式；但是安全性、并发控制、死锁等问题突出。

消息传递：消息是软件对象之间进行交互和通信时所使用的一种数据结构，可以独立于软件平台而存在，适用于数据量小、但要求频繁、立即、可靠、异步地数据交换场合。

文件传输：是指在进行数据交换时，直接将数据文件传送到相应位置，让目标系统直接读取数据，可以一次性传送大量信息，但不适合频繁进行数据传送。适用于数据量大、交换频度小、即时性要求低的情况。

企业集成平台

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 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。

合格范文赏析

摘要：2015 年 2 月，本人所在的某家商业银行启动了零售 CRM 项目建设，该项目主要实现客户管理、客户分析、营销管理、绩效管理、数据 ETL 处理、多渠道数据服务等功能，在此项目中，我担任架构师，负责项目总体架构设计工作。本文以该零售 CRM 项目建设为例，主要论述了企业集成架构设计技术在该系统中的具体应用。操作型 CRM 和分析型 CRM 两个子系统通过零售 CRM 门户实现界面集成；数据服务系统通过银行已有 ESB 与各请求系统实现应用集成；利用数据总线 and 数据仓库实现数据集成，建立仓内零售 CRM 数据集市，为客户分析、绩效管理、数据服务提供数据支持。通过以上集成

架构技术将项目中多个子系统进行整合，形成有机整体，最终项目顺利上线，至今系统运行稳定，受到用户一致好评。

本人所在的某家商业银行分支机构遍布全国省会和重点城市，零售客户数量近 6 千万，客户经理数量约 5 千。客户经营维系情况参差不齐，客户经理经营维系客户的策略、方法、工具缺乏统一，抬高了经营成本，降低了客户体验。营销活动的有效性和目标性不强，往往营销成本较大但效果不明显，针对这些问题，同时为了适应竞争激烈的市场环境，该银行制定了零售 CRM 业务战略，并于 2015 年 2 月启动了零售 CRM 项目建设，主要实现客户管理、客户分析、营销管理、绩效管理、数据 ETL 处理、多渠道数据服务等功能，该项目旨在建立全行统一的零售客户经营管理体系和零售数据分析环境，充分发挥长期积累数据的业务价值，科学、有效的指导客户获取、营销、服务、挽留等全生命周期活动，提高客户经营和服务水平，增强客户体验，提升客户价值，从而提高零售业务的利润回报。

在此项目中，我担任架构师，负责项目总体架构设计。项目涉及范围广泛，我采用“分而治之”策略，根据应用类型、用户角色、数据处理特征的不同，将项目分为多个（子）系统，并研究企业集成的关键技术，根据项目实际情况，选择合适的集成技术并进行架构设计。

企业集成是指使用应用服务器、中间件等平台和技术，连接企业内的各应用系统，实现异构系统之间的交互和协作，以及数据交换和共享。企业集成模式与技术主要有以下四种：1、界面集成，把各个应用系统的界面集中在一个界面之中，常用技术为企业门户。2、过程集成，使各个应用系统连接起来支持完整的业务流程，常用技术为工作流、企业门户。3、应用集成，为两个以上应用系统中的数据和程序提供接近实时的集成，常用技术为远程过程调用、消息中间件、服务总线（ESB）、Web 服务等。4、数据集成，解决的是信息系统之间数据同步（包括主数据系统向副本系统同步，源系统向数据仓库同步）和时效性（包括实时、批量）问题，常用技术为适配器、消息中间件、数据总线、数据仓库。下面我将以通过企业门户、服务总线、数据总线、数据仓库等技术或平台，来实现界面、应用、数据集成来具体论述其实现过程。

1、操作型 CRM 和分析型 CRM 两个子系统通过零售 CRM 门户实现界面集成。

客户管理、营销管理、客户分析、绩效管理四部分功能，属于面向银行内部用户的管理分析类需求，适宜采用 BS 架构搭建零售 CRM 系统。为了降低耦合，将零售 CRM 系统划分为操作型 CRM（实现客户管理和营销管理功能）和分析型 CRM（实现客户分析、绩效管理功能）两个子系统。但对于用户使用便捷性却不能降低，需要将两个子系统的界面集中在一起，这要使用界面集成技术，于是开发零售 CRM 门户进行界面集成。零售 CRM 门户作为零售用户的统一工作台，为两个子系统提供统一登录界面和访问入口、统一用户认证和授权、统一界面风格、统一功能菜单和待办事项，使用代理技术实现门户系统向两个子系统页面的路由和跳转，提高了用户体验，使得子系统划分对于用户透明。门户系统采用 J2EE 开源框架（Spring、MyBatis）进行开发，技术架构分为页面展现、接入控制、业务逻辑、数据访问四层。

2、数据服务系统通过银行已有 ESB 与各请求系统实现应用集成。

多渠道数据服务功能，既面向客户经理通过零售 CRM 系统 Web 界面查询客户交易数据，也面向客户通过网上银行和手机银行系统查询交易数据。经分析，查询功能可以定义为标准的服务，支持复用

且面向多个请求系统提供接口访问，各系统技术异构，适宜采用 SOA 架构风格搭建数据服务系统，并使用应用集成技术，利用银行已有服务总线（ESB）向多个系统提供查询服务。ESB 包括服务管理、协议转换、格式转换、服务路由、消息处理等功能。将数据服务系统的查询服务在 ESB 中注册并发布，各请求系统向 ESB 发起服务请求，ESB 接入请求，通过适配器技术进行通讯协议和报文格式的转换，经服务路由，向数据服务系统提交查询请求。

3、利用数据总线 and 数据仓库实现数据集成。

数据 ETL 处理，指将银行前台存款、贷款、信用卡、理财等上百个、技术异构、数据标准不统一的交易系统的数据进行采集、转换、清洗、存储、整合、加工，为客户分析、绩效管理、数据服务提供数据支持，这正是数据总线 and 数据仓库要解决的问题，属于数据集成范畴。数据总线设计过程为：通过数据采集器（IBMII 产品），每日日终自动从源系统采集数据并将编码转换为文本格式，通过数据处理器（IBMDaStage 产品），进行数据质量检查（要定义检查规则）、增量数据生成、数据标准转换等（要定义数据标准）处理，通过数据传输器，将数据文件传输给目标系统（包括数据仓库）。数据仓库设计过程为：接收数据总线传输来的数据，加载至数据仓库系统（要选择具有海量数据处理能力且能水平扩展的产品），对数据按照主题模型（比如客户、产品、账户、交易、渠道等）进行归类、整合，并存放长期历史数据，之上建立零售 CRM 数据集市，根据本次需求的数据加工规则，对数据进行关联、聚合等加工处理，生成数据文件，传输并加载至分析型 CRM 子系统、数据服务系统，为客户分析、绩效管理、交易查询提供数据支持。

项目经过一年时间的开发测试，于 2016 年 2 月上线运行，至今系统运行稳定，使用效果良好，有力地支持了零售数据分析和客户经营管理。项目总结会上，领导对于系统划分方法及采用的界面、应用、数据集成方式表示高度认同。

近来，业务提出呼叫中心客服坐席要查询客户历史交易明细，由于采用了 ESB 应用集成技术，所以可以方便接入呼叫中心系统。面对客户分析和绩效管理的数据维度和指标经常发生变化且要重算，由于数据仓库积累了长期历史数据，所以方便应对数据变更需求。但面对客户提出通过网上银行和手机银行系统查询当日发生的交易明细需求，由于目前数据总线只支持数据 T+1 时效，所以要在数据总线中增加实现数据准实时采集机制，将交易数据准实时同步至数据服务系统，以提高数据服务时效性。

11 论软件设计模式及其应用

论软件设计模式及其应用

软件设计模式（Software Design Pattern）是一套被反复使用的、多数人知晓的、经过分类编目的代码设计经验的总结。使用设计模式是为了重用代码以提高编码效率、增加代码的可理解性、保证代码的可靠性。软件设计模式是软件开发中的最佳实践之一，它经常被软件开发人员在面向对象软件开发过程中所采用。项目中合理地运用设计模式可以完美地解决很多问题，每种模式在实际应用中都有相应的原型与之相对，每种模式都描述了一个在软件开发中不断重复发生的问题，以及对应该原型问题的核心解决方案。

请围绕“论软件设计模式及其应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与分析和开发的软件系统，以及你在项目中所担任的主要工作。
2. 说明常用的软件设计模式有哪几类？阐述每种类型特点及其所包含的设计模式。
3. 详细说明你所参与的软件系统开发项目中，采用了哪些软件设计模式，具体实施效果如何。

找准核心论点

问题 1 要点：

软件系统的概要：系统的背景、发起单位、目的、开发周期、交付的产品等。

“我”的角色和担任的主要工作。

问题 2 要点：

常用的软件设计模式的分类。

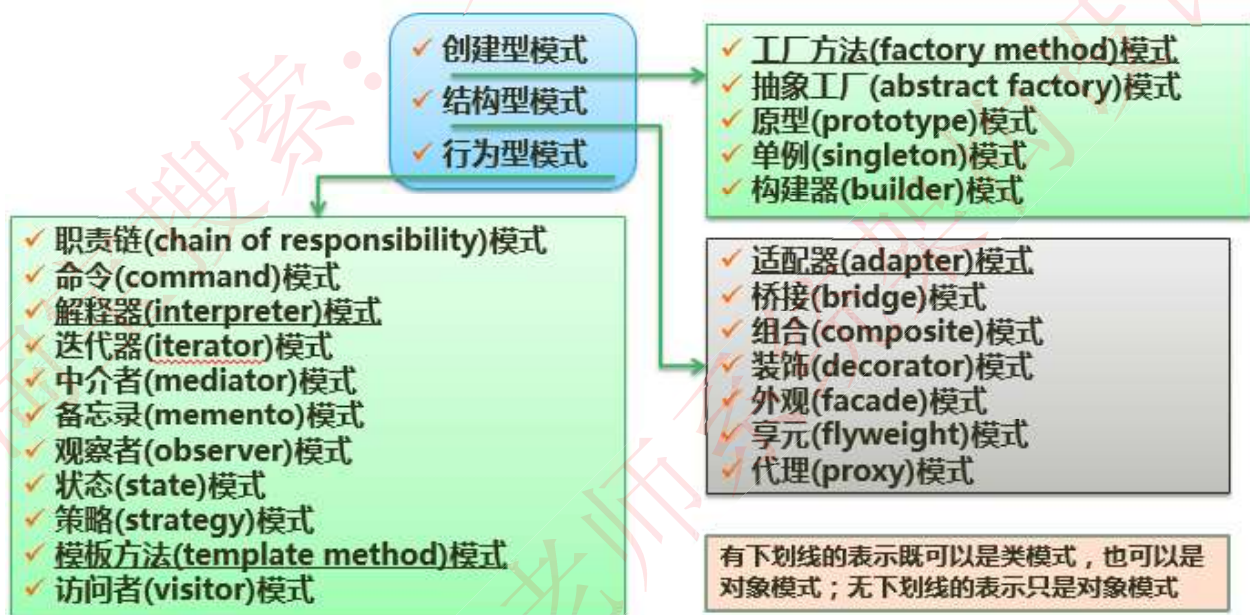
说明每种类型特点及其所包含的设计模式。

问题 3 要点：

采用了哪些软件设计模式，说明其具体实施步骤。

最终效果如何。

理论素材准备



合格范文赏析

摘要：2014 年 3 月，本人所在公司承担了一项农业系统平台的开发项目，该项目主要是实现农业系统各项内部业务，以及各项农业项目的审批工作，并提供外部用户通过 web 服务进行信息访问。我在该项目中担任系统架构设计师一职，负责系统的架构设计和软件开发的部分设计工作。本文以该农业系统平台的开发项目为例，主要论述了软件设计模式在该系统开发中的具体应用。在农产品标准化模块中，针对不同的农产品质量指标我们采用了责任链模式；在数据访问中我们采用了工厂模式，

以实现对不同数据格式的转换；在验证码生成中我们采用了策略模式，以实现算法的灵活替换。通过使用这些设计模式，提高了软件的设计质量和开发效率，最终项目顺利上线，并获得用户一致好评。

2014 年 3 月，本人所在公司承担了某市农委系统的系统平台的开发项目，该项目是农业系统的工作平台。不仅承担着农委系统的内部业务工作，包括：生产处、环能处、经管处、农村处、生态处等多个处室。通过实施该系统，可以实现不同处室的业务信息的共享和交流，消除信息孤岛，提高办事效率和质量。另一方面，通过这个平台，可以为农产品加工企业、合作社农户等涉农群体，提供信息公开、在线审批、政策查询、留言信箱、技术推广等农业服务，实现与农产品加工企业、合作社农户等社会群体的网上在线交互，提高服务三农的质量和水平。在该项目中，本人担任系统架构设计师，负责项目的架构设计以及软件开发的部分工作。

由于传统的结构化的软件设计方法不符合面向对象的设计原则，无法很好的实现高内聚和低耦合的要求。模块之间过于紧密，给软件扩展和维护带来很多困难。在这种情况下，设计模式的出现和广泛应用给问题的解决提供了一种有效方法。通过利用设计模式，可以帮助开发者复用已有的设计方法，设计出结构合理、易于复用和可维护的软件，当用户需要发生改变时，可以通过修改少量代码或不修改原有代码即可满足新的需求，增强了系统的可修改性和稳定性，降低系统开发成本。

一般而言，一个设计模式具有模式名称、适应场景、解决方案和效果四个方面的基本要素。设计模式依据其目的可以分为创建型、结构型、行为型三种类型。创建型模式，主要负责对象的创建工作，程序在确定需要创建对象时，可以获得更大的灵活性。常用的创建型设计模式有：单例模式、工厂方法、原型、构造器、抽象工厂等 5 种模式。结构型模式，负责处理类或对象之间的关系，用于构件结构更加复杂庞大的系统。常用的结构型设计模式有适配器、桥接模式、享元模式、组合模式、外观模式、代理模式等 7 种模式。行为型模式，主要任务是对类或对象如何交互以及为类和对象分配具体职责进行描述。常用的行为型模式有观察者、状态、策略模式、备忘录、命令、责任链、中介者等 11 种模式。这些设计方法都是经过反复使用的成熟方法，对优化软件结构，提高软件质量具有重要的指导意义。

在农业信息平台的开发过程中，我们综合使用了多种设计模式，本文着重对责任链模式、工厂方法、策略模式等 3 种设计模式在该项目中的具体应用进行介绍。

一、责任链模式

我们在信息平台的开发过程中，需要完成对农产品质量进行标准化评选，从低到高评选无公害农产品、绿色产品、有机食品、地理标志认证 4 种认证，其中，无公害农产品的认定数量较多，标准较低，由农业生产处进行认定。在认定过程中，我们采用了责任链的设计模式。首先，定义了农产品对象 `fproducts`，该对象中保存有农产质量的各项指标，包括水、空气、土壤等环境质量指标，及耕地净化、品种优质高抗、投入品无害化等生产技术。能够全面反映农产品质量水平。其次，我们定义了接口类 `deal`，接口中持有一个农产品对象和自身的接口，以及处理函数 `processrequest`。对外提供对农产品进行分类，并存入不同的信息数据库。随后，我们定义了无公害处理类、绿色食品、有机食品和地理标志 4 个实现类。对农产品对象 `fproducts` 的处理，按照由高到低的顺序，依次进行处理，

直到符合某个标准为止，并完成信息处理，将对象信息按照审核的分档标准，存入信息库。通过这个方法，可以实现农产品对象，与处理方法的分离。

二、工厂方法

在农业产业化管理过程中，需要对各区市数据进行采集，由于不同类型的数据导入算法不同，在程序设计过程中，设计者需要定义若干类分别实现导入 excel、xml、sqldata 等类型的数据的算法，而且用户导入的数据类型存在不确定性，设计者无法确定应该实例化哪一个类。为解决这一问题，我们使用工厂方法模式。首先，定义一个数据访问接口类 import。同时，针对不同的数据类型，还定义了 ImportExcel、ImportXML、ImportSQLDATA 具体产品类，实现了 import 所声明的公共接口，其主要功能是封装了不同类型的数据导入到数据库的具体算法。Importcreator 是抽象工厂类，持有一个接口产品类 import 的对象。Importexcelcreator、importXMLCreator、importSQLdata 是具体工程类，主要功能是生产具体产品实体，直接在客户端的调用下创建产品实例。通过工厂方法模式的引入，可以有效解决客户需要变化对设计的影响，设计者无需知道那个子类被实例化，子类会根据具体情况自己决定实例化哪一个类，而且创建具体产品的细节也有着很好的封装，符合高内聚、低耦合的设计和原则。当需要在系统中添加新的产品时，也不需要修改抽象工厂和抽象产品的接口，以及其他具体工厂和具体产品，具有很好的可扩展性。

三、策略模式

在系统的安全性方面，我们采用了用户名—密码—自动验证码相结合的办法，以保证系统访问安全性。根据验证码的使用环境，一般分为数字验证码、汉字验证码、英文验证码 3 种类型。而生成不同类型验证码的算法存在巨大差异，为此需要定义不同的生成验证码的算法。为解决此问题，可以利用策略模式将不同的算法封装起来，并使他们可以相互替换，使得算法独立于使用它的客户而变化。在设计策略模式中，我们定义了 3 个角色。环境角色：持有一个抽象策略角色 StrategyVerifyCode 接口的引用，并通过 StrategyVerifyCode 接口，来实现一个具体的策略算法。抽象策略角色：定义所有的具体策略类所需的统一访问接口；具体策略角色：包装了相关的算法或行为；在该项目中，我们按照数字、文字、字符 3 种类型，分别定义了 shuzi_verify、zifu_verify、wenzi_verify 3 个具体的策略类。通过使用策略算法，将生成验证码的算法封装在一个个独立的策略类中，用户可以根据自己的需求从不同策略中进行选择，有效的避免了使用条件转移语句不易维护的缺点。而且策略模式利用组合代替继承，将算法的实现与算法的选择分离开来，降低了程序之间的耦合度，增强了代码的可扩展性和可维护性。

以上设计模式的选用基本达到了预期的效果。首先是，这些设计模式都是一些常用的设计方法，在架构设计师、系统架构设计师、开发人员之间，形成了良好的沟通桥梁，大家很容易进行交流和沟通。其次，在使用设计模式过程中，软件的开发效率较高，能够节省开发成本。最重要的是，这些模式都是一些经过反复使用的成熟设计方案，符合面向对象中设计规范，比如：面向接口编程、里氏替换原则、单一职责原则、依赖倒转等设计原则，最大限度的提高软件的标准化的，为日后的系统维护打下了很好的基础。

当然，我们在设计过程中，也存在一些问题和不足，不少开发人员在设计过程中，有时还是习惯于原有的设计方法，对模式的使用有些抵触。而且，这些设计模式在应用过程中，往往不是单独使用，需要对多个模式进行综合运用。这方面，我们还缺少相关的经验。所以，在以后的项目设计中，我们将继续应用各种设计模式，做到融会贯通，不拘一格的目标，争取能设计出更多的高质量软件项目。

四 历年论文写作题目

2022 年下半年试题 1

试题一 论基于构件的软件开发方法及其应用

基于构件的软件开发（Component-Based Software Development, CBSD）是一种基于分布对象技术、强调通过可复用构件设计与构造软件系统的软件复用途径。基于构件的软件系统中的构件可以是COTS（Commercial-Off-the-Shelf）构件，也可以是通过其它途径获得的构件（如自行开发）。CBSD将软件开发的重点从程序编写转移到了基于已有构件的组装，以更快地构造系统，减轻用来支持和升级大型系统所需要的维护负担，从而降低软件开发的费用。

问题内容：

请围绕“基于构件的软件开发方法及其应用”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 详细论述基于构件的软件开发方法的主要过程。
- 3 结合你具体参与管理和开发的实际项目，请说明具体实施过程以及碰到的主要问题。

解析：这个题目真的是可惜了，最后押题的时候我是准备把这个题目放进去的，以前连续放了一年，这次犹豫了。不过这个在我们论文总复习资料里有，如果往下写是能写到的，还是大家都没往下学啊。

2022 年下半年试题 2

试题二 论软件维护方法及其应用

软件维护是指在软件交付使用后，直至软件被淘汰的整个时间范围内，为了改正错误或满足新的需求而修改软件的活动。在软件系统运行过程中，软件需要维护的原因是多种多样的，根据维护的原因不同，可以将软件维护分为改正性维护、适应性维护、完善性维护和预防性维护。在维护的过程中，

也需要对软件的可维护性进行度量。在软件外部，一般采用 MTIR 来度量软件的可维护性；在软件内部，可以通过度量软件的复杂性来间接度量软件的可维护性。

据统计，软件维护阶段占整个软件生命周期 60% 以上的时间。因此，分析影响软件维护的因素，度量和提高软件的可维护性，就显得十分重要。

问题内容：

请围绕“软件维护方法及其应用”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 详细论述影响软件维护工作的因素有哪些。
- 3 结合你具体参与管理和开发的实际项目，说明在具体维护过程中，如何度量软件的可维护性，说明具体的软件维护工作类型。

解析：这个题目是可以选的嘛，比较容易写，就回答问题 2, 3 的要求，从软件维护的类型来写，例如完善性、适应性、预防性等，一个一段，有很多东西可以写的。

2022 年下半年试题 3

试题三 论区块链技术及应用

区块链作为一种分布式记账技术，目前已经被应用到了资产管理、物联网、医疗管理、政务监管等多个领域。从网络层面来讲，区块链是一个对等网络（Peer to Peer, P2P），网络中的节点地位对等，每个节点都保存完整的账本数据，系统的运行不依赖、中心化节点，因此避免了中心化带来的单点故障问题。同时，区块链作为一个拜占庭容错的分布式系统，在存在少量恶意节点情况下可以作为一个整体对外提供稳定的服务。

问题内容：

请围绕“区块链技术及应用”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 区块链包含多种核心技术，请简要描述区块链的 3 种核心技术。
- 3 具体阐述你参与管理和开发的项目是如何应用区块链技术进行设计与实现。

2022 年下半年试题 4

试题四 论湖仓一体架构及其应用

随着 5G、大数据、人工智能、物联网等技术的不断成熟，各行各业的业务场景日益复杂，企业数据呈现出大规模、多样性的特点，特别是非结构化数据呈现出爆发式增长趋势。在这一背景下，企业数据管理不再局限于传统的结构化 OLTP (On-Line Transaction Processing) 数据交易过程，而是提出了多样化、异质性数据的实时处理要求。传统的数据湖 (Data Lake) 在事务致性及实时处理方面有所欠缺，而数据仓库 (Data Warehouse) 也无法应对高并发、多数据类型的处理。因此，支持事务一致性、提供高并发实时处理及分析能力的湖仓一体 (Lake House) 架构应运而生。湖仓一体架构在成本、灵活性、统一数据存储、多元数据分析等多方面具备优势，正逐步转化为下一代数据管理系统的核心竞争力。

问题内容：

请围绕“湖仓一体架构及其应用”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、采用湖仓一体架构的软件项目以及你在其中所承担的主要工作。
- 2 请对湖仓一体架构进行总结与分析，给出其中四类关键特征，并简要对这四类关键特征的内涵进行阐述。
- 3，具体阐述你参与管理和开发的项目是如何采用湖仓一体架构的，并围绕上述四类关键特征，详细论述在项目设计与实现过程中遇到了哪些实际问题，是如何解决的。

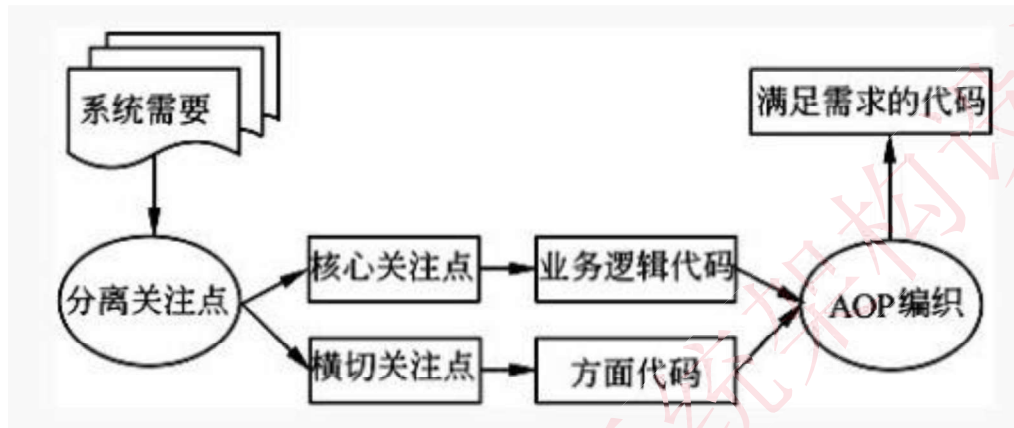
2021 年下半年试题 1

论面向方面的编程技术及其应用

针对应用开发所面临的规模不断扩大、复杂度不断提升的问题，面向方面的编程 (Aspect Oriented Programming, AOP) 技术提供了一种有效的程序开发方法。为了理解和完成一个复杂的程序，通常要把程序进行功能划分和封装。一般系统中的某些通用功能，如安全性、持续性、日志记录等等，其代码是分散的，较难实现模块化，不利于程序演变、维护和更新。AOP 技术将逻辑上关系松散的代码封装到一个具有某种公共行为的可重用模块，并将其命名为方面 (Aspect)。

请围绕“面向方面的编程技术及其应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与实施的应用 AOP 技术的软件项目以及你在其中所担任的主要工作。
2. 叙述在软件项目实践过程使用 AOP 技术开发的具体步骤。
3. 结合项目内容，论述该项目使用 AOP 技术的原因，开发过程中存在的问题和解决方法，以及使用 AOP 技术带来的实际应用效果。



解析：

AOP 包括三个开发步骤，分别是方面分解、关注点实现和方面的重新组合。

(1) 方面分解。分解需求提取出横切关注点和核心关注点。把核心模块级关注点和系统级的横切关注点进行分离。例如，对于一个信用卡系统，可以分解出三个关注点：核心的信用卡处理、日志和验证。

(2) 关注点实现。各自独立地实现这些关注点，用 OOP（面向对象的程序设计）实现核心关注点，用 AOP 实现横切关注点。例如，可以用 OOP 实现信用卡处理单元，而用 AOP 实现日志单元和验证单元。

(3) 方面的重新组合。方面集成器通过创建一个模块单元（方面）来制定重组的规则，重组过程也称为编织。

2021 年下半年试题 2

论系统安全架构设计及其应用

随着社会信息化进程的加快，计算机及网络已经被各行各业广泛应用，信息安全问题也变得愈来愈重要。它具有机密性、完整性、可用性、可控性和不可抵赖性等特征。信息系统的安全保障是以风险和策略为基础，在信息系统的整个生命周期中提供包括技术、管理、人员和工程过程的整体安全，以保障信息的安全特征。

请围绕“系统安全架构设计及其应用”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涉及安全架构设计的软件项目以及承担的主要工作。
2. 请详细论述安全架构设计中鉴别框架和访问控制框架设计的内容，并论述鉴别和访问控制所面临的主要威胁有哪些，说明其危害。
3. 请简要说明在你所参与项目的开发过程中，在鉴别框架和访问控制框架设计中存在的实际问题，以及是如何解决这些问题的。

解析：鉴别（Authentication）的基本目的，就是防止其他实体占用和独立操作被鉴别实体的身份。鉴别提供了实体声称其身份的保证，只有在主体和验证者的关系背景下，鉴别才是有意义的。鉴

别有两种重要的关系背景：一是实体由申请者来代表，申请者与验证者之间存在着特定的通信关系（如实体鉴别）；二是实体为验证者提供数据项来源。

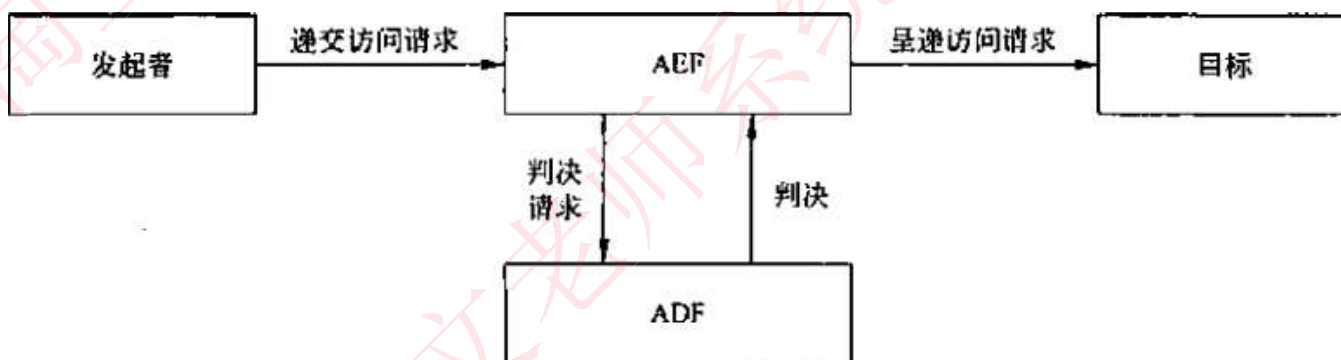
鉴别的方式主要基于以下 5 种。

- (1) 已知的，如一个秘密的口令。
- (2) 拥有的，如 IC 卡、令牌等。
- (3) 不改变的特性，如生物特征。
- (4) 相信可靠的第三方建立的鉴别（递推）。
- (5) 环境（如主机地址等）。

鉴别服务分为以下阶段：安装阶段；修改鉴别信息阶段；分发阶段；获取阶段；传送阶段；验证阶段；停活阶段；重新激活阶段；取消安装阶段。

在安装阶段，定义申请 AI 和验证 AI。修改鉴别信息阶段，实体或管理者申请 AI 和验证 AI 变更（如修改口令）。在分发阶段，为了验证交换 AI，把验证 AI 分发到各实体（如申请者或验证者）以供使用。在获取阶段，申请者或验证者可得到为鉴别实例生成特定交换 AI 所需的信息，通过与可信第三方进行交互或鉴别实体间的信息交换可得到交换 AI。例如，当使用联机密钥分配中心时，申请者或验证者可从密钥分配中心得到一些信息，如鉴别证书。在传送阶段，在申请者与验证者之间传送交换 AI。在验证阶段，用验证 AI 核对交换 AI。在停活阶段，将建立一种状态，使得以前能被鉴别的实体暂时不能被鉴别。在重新激活阶段，使在停活阶段建立的状态将被终止。在取消安装阶段，实体从实体集合中被拆除。

访问控制（AccessControl）决定开放系统环境中允许使用哪些资源、在什么地方适合阻止未经授权访问的过程。在访问控制实例中，访问可以是对一个系统（即对一个系统通信部分的一个实体）或对一个系统内部进行的。



基本访问控制功能示意图

ACI（访问控制信息）是用于访问控制目的的任何信息，其中包括上下文信息。ADI（访问控制判决信息）是在做出一个特定的访问控制判决时可供 ADF 使用的部分（或全部）ACI。ADF（访问控制判决功能）是一种特定功能，它通过对访问请求、ADI 以及该访问请求的上下文使用访问控制策

略规则而做出访问控制判决。AEF（访问控制实施功能）确保只有对目标允许的访问才由发起者执行。

涉及访问控制的有发起者、AEF、ADF 和目标。发起者代表访问或试图访问目标的人和基于计算机的实体。目标代表被试图访问或由发起者访问的，基于计算机或通信的实体。例如，目标可能是 OSI 实体、文件或者系统。访问请求代表构成试图访问部分的操作和操作数。

当发起者请求对目标进行特殊访问时，AEF 就通知 ADF 需要一个判决来做出决定。为了作出判决，给 ADF 提供了访问请求（作为判决请求的一部分）和下列几种访问控制判决信息（ADI）。

2021 年下半年试题 3

论企业集成平台的理解与应用

企业集成平台（Enterprise Integration Platform, EIP）是支持企业信息集成的像环境，其主要功能是企业中的数据、系统和应用等多种对象的协同行提供各种公共服务及运行时的支撑环境。企业集成平台能够根据业务模型的变化快速地进行信息系统的配置和调整，保证不同系统、应用、服务或操作人员之间顺畅地相互操作，进而提高企业适应市场变化的能力，使企业能够在复杂多变的市场环境中生存。

请围绕“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. 概要叙述你参与管理和开发的、采用企业集成平台进行企业信息集成的软件项目以及你在其中所承担的主要工作。
2. 请给出至少 4 种企业集成平台应具有的基本功能，并对这 4 种功能的内涵进行简要阐述。
3. 具体阐述你参与管理和开发的项目是如何使用企业集成平台进行企业信息集成的，并围绕上述 4 种功能，详细论述在集成过程中遇到了哪些实际问题，是如何解决的。

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，其基本功能要如下。

（1）通信服务

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

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

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，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

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

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。

2021 年下半年试题 4

论微服务架构及其应用

微服务架构（Microservices Architecture）是一种架构风格，它将一个复杂的应用拆分成多个独立自治的服务，服务与服务间通过松耦合的形式交互，在微服务架构中，服务是细粒度的，协议是轻量级的。这些服务通常按业务能力组织，有自身的技术堆栈。

请围绕“微服务架构及其应用”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、采用微服务架构的软件项目以及你在其中所承担的主要工作。
2. 请简要描述微服务架构的优点。
3. 具体阐述你参与管理和开发的项目是如何基于微服务架构进行设计实现的。

解析：

微服务好处：

高异构性，高性能，高弹性，高扩展，易部署，可组合性，可替代性

微服务优点：

- 通过应用“分而治之”的原则，持续交付和部署大型，复杂的应用程序
- 通过更易于理解，开发和测试系统来提高模块化
- 通过每个微服务具有较小的代码库来降低复杂性
- 允许更新功能，而对系统的其余部分没有影响或影响极小
- 使架构变得高度可扩展

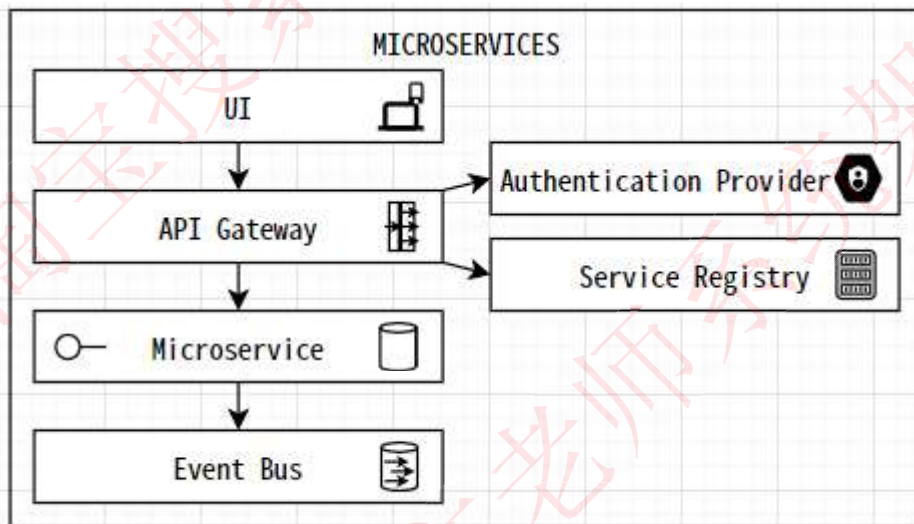
- 大大减少了破坏系统无关部分的机会
- 可以独立交付和部署服务，而不必等待整个系统发布
- 允许部署到多个云和本地基础设施环境
- 在持续发展现有系统的同时持续融入和利用最新的技术
- 使同一时间在同一系统上工作的一组开发人员间的协作更可控
- 允许新的团队成员更快地提高生产力，他们可以开发新功能而不必学习整个系统

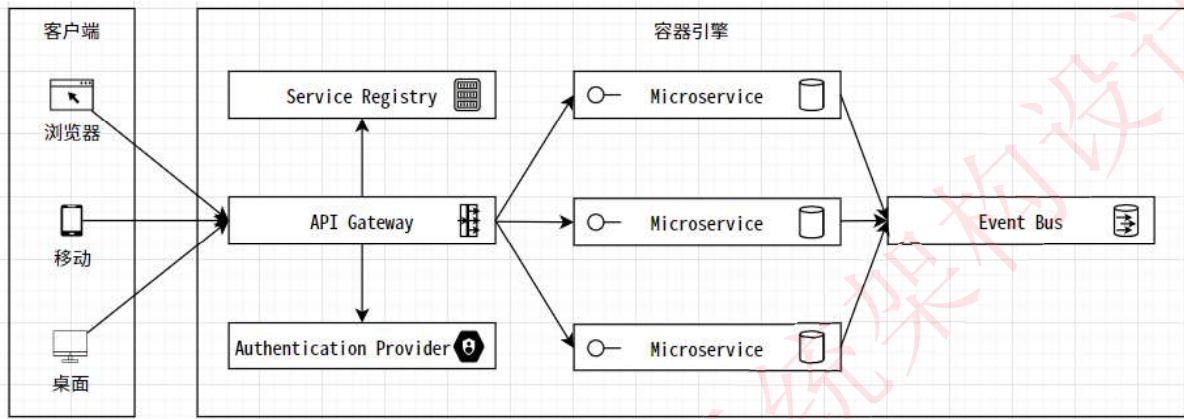
基于微服务的系统设计实现：

设计原则

- 围绕业务概念建模
- 实现自动化
- 隐藏内部实现细节
- 一切去中心化
- 独立部署
- 隔离失败
- 高度可观察

设计实现





微服务 RESTful API :

业务服务及通用服务

服务网关 API Gateway:

客户端到微服务通信

服务注册 Service Registry:

微服务注册，发现中心

事件总线 Event Bus :

微服务到微服务通信

安全保护 Auth Provider:

认证授权提供服务

2020 年下半年试题 1

企业集成架构 (Enterprise Integration Architecture, EIA) 是企业集成平台的核心，也是解决企业信息孤岛问题的关键。企业集成架构设计包括了企业信息、业务过程、应用系统集成架构的设计。实现企业集成的技术多种多样，早期的集成方式是通过在不同的应用之间开发一对一的专用接口来实现应用之间的数据集成，即采用点到点的集成方式；后来提出了利用集成平台的方式来实现企业集成，可以将分散的信息系统通过一个统一的接口，以可管理、可重复的方式实现单点集成。企业集成架构设计技术方案按照要解决的问题类型可以分为数据集成、应用集成和企业集成。

请围绕“论企业集成架构设计及应用”论题，依次从以下三个方面进行论述：

1. 概要叙述你参与的软件开发项目以及承担的主要工作。
2. 详细说明三类企业集成架构设计技术分别要解决的问题及其含义，并阐述每种技术具体包含了哪些集成模式。
3. 根据你所参与的项目，说明采用了哪些企业集成架构设计技术，其实施效果如何。

写作要点

一、简要描述所参与的软件系统开发项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、详细说明三类企业集成架构设计技术分别要解决的问题及其含义，并阐述每种技术具体包含了哪些集成模式。

1. 数据集成。

数据集成是为了解决不同应用和系统间的数据共享和交换需求，具体包括共享信息管理、共享模型管理和数据操作管理三个部分。共享信息管理通过定义统一的集成服务模型和共享信息访问机制，完成对集成平台运行过程中产生数据信息的共享、分发和存储管理；共享模型管理则提供数据资源配置管理、集成资源关系管理、资源运行生命周期管理及相应的业务数据协同监控管理等功能；数据操作管理则为集成平台用户提供数据操作服务，包括多通道的异构模型之间的数据转换、数据映射、数据传递和数据操作等功能服务。

数据集成的模式包括：数据联邦、数据复制模式、基于结构的数据集成模式。

2. 应用集成。

应用集成是指两个或多个应用系统根据业务逻辑的需要而进行的功能之间的互相调用和互操作。应用集成需要在数据集成的基础上完成。应用集成在底层的网络集成和数据集成的基础上实现异构应用系统之间应用层次上的互操作。它们共同构成了实现企业集成化运行最顶层集成所需要的技术层次上的支持。

应用集成的模式包括：集成适配器模式、集成信使模式、集成面板模式和集成代理模式。

3. 企业集成。

企业应用软件系统从功能逻辑上可以分为表示、业务逻辑和数据三个层次。其中表示层负责完成系统与用户交互的接口定义；业务逻辑层主要根据具体业务规则完成相应业务数据的处理；数据层负责存储由业务逻辑层处理所产生的业务数据，它是系统中相对稳定的一部分。支持企业间应用集成和交互的集成平台通常采用多层结构，其目的是在最大程度上提高系统的柔性。在集成平台的具体设计开发中，还需要按照功能的通用程度对系统实现模块进行分层。

企业集成的模式包括：前端集成模式、后端集成模式和混合集成模式。

三、针对考生本人所参与的项目中使用的企业集成架构设计技术，说明实施过程和具体实施效果。

2020年下半年试题2

软件缺陷指的是计算机软件或程序中存在的某种破坏正常运行能力的问题、错误，或者隐藏的功能缺陷。缺陷的存在会导致软件产品在某种程度上不能满足用户的需要。在目前的软件开发过程中，缺陷是不可避免的。软件测试是发现缺陷的主要手段，其核心目标就是尽可能多地找出软件代码中存在的缺陷，进而保证软件质量。软件缺陷管理是软件质量管理的一个重要组成部分。

请围绕“论软件测试中缺陷管理及其应用”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承担的主要工作。

2. 详细论述常见的缺陷种类和级别，论述缺陷管理的基本流程。

3. 结合你具体参与管理和开发的实际项目，说明是如何进行缺陷管理的，请说明具体实施过程以及应用效果。

写作要点

一、简要叙述所参与管理和开发的软件项目，并明确指出在其中承担的主要任务和开展的主要工作。

二、根据 IEEE 标准，软件测试中所发现的缺陷主要包括：输入/输出错误；逻辑错误；计算错误；接口错误；数据错误等；从软件测试角度还可以将缺陷分为五类：功能缺陷；系统缺陷；加工缺陷；数据缺陷；代码缺陷。不同企业的缺陷分类往往不同。

根据缺陷后果的严重程度，可以将缺陷分为多个不同的级别，例如 Beizer 将缺陷分为十级：轻微、中等、使人不悦、影响使用、严重、非常严重、极为严重、无法容忍、灾难性、传染性等。

缺陷管理是对软件测试环节中缺陷状态的完整跟踪和管理，确保每个被发现的缺陷都得到妥善处理。缺陷管理的目的是对各个阶段测试发现的缺陷进行跟踪管理，以保证各级缺陷的修复率达到标准，主要实现以下目标：保证信息的一致性；保证缺陷得到有效的跟踪；缩短沟通时间，解决问题更高效；收集缺陷数据并进行数据分析，作为缺陷度量的依据。

缺陷管理基本的流程如下：

(1) 缺陷提交：测试人员发现缺陷后提交缺陷报告。

(2) 缺陷审查：确定缺陷问题、种类和级别。

(3) 修复流程：缺陷审查通过后进入修复流程，缺陷报告会转发给相应的软件开发人员进行修复。

(4) 验证流程：开发人员提交修复后的代码，进入验证流程。通过回归测试等方法验证缺陷问题已经修复。

(5) 缺陷关闭：在确认缺陷已完全解决后，关闭该缺陷。

部分缺陷管理流程中，还包括对缺陷状态的跟踪。

三、考生需结合自身参与项目的实际状况，指出其参与管理和开发的项目中所进行的缺陷管理活动，说明缺陷管理的具体实施过程，并对实际应用效果进行分析。

2020 年下半年试题 3

近年来，随着数字化转型不断深入，科技创新与业务发展不断融合，各行各业正在从大工业时代的固化范式进化成面向创新型组织与灵活型业务的崭新模式。在这一背景下，以容器和微服务架构为代表的云原生技术作为云计算服务的新模式，已经逐渐成为企业持续发展的主流选择。云原生架构是基于云原生技术的一组架构原则和设计模式的集合，旨在将云应用中的非业务代码部分进行最大化剥离，从而让云设施接管应用中原有的大量非功能特性（如弹性、韧性、安全、可观测性、灰度等），使业务不再有非功能性业务中断困扰的同时，具备轻量、敏捷、高度自动化的特点。云原生架构有

利于各组织在公有云、私有云和混合云等新型动态环境中，构建和运行可弹性扩展的应用，其代表技术包括容器、服务网格、微服务、不可变基础设施和声明式 API 等。

请围绕“论云原生架构及其应用”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承担的主要工作。
2. 服务化、弹性、可观测、韧性和自动化是云原生架构重要的设计原则。请简要对这些设计原则的内涵进行阐述。
3. 具体阐述你参与管理和开发的项目是如何采用云原生架构的，并围绕上述四类设计原则，详细论述在项目设计与实现过程中遇到了哪些实际问题，是如何解决的。

写作要点

一、简要叙述所参与管理和开发的软件项目，需要明确指出在其中承担的主要任务和开展的主要工作。

二、云原生架构的设计原则具体描述如下：

(1) 服务化原则。当代码规模超出小团队的合作范围时，就有必要进行服务化拆分，包括拆分为微服务架构、小服务（mini service）架构，通过服务化架构把不同生命周期的模块分离出来，分别进行业务迭代，避免迭代频繁模块被慢速模块拖慢，从而加快整体的进度和稳定性。同时服务化架构以面向接口编程，服务内部的功能高度内聚，模块间通过公共功能模块的提取增加软件的复用程度。

(2) 弹性原则。大部分系统部署上线需要根据业务量的估算，准备一定规模的机器，传统上线过程中需要经历采购申请、供应商洽谈、机器部署上电、软件部署、性能压测等阶段，周期很长，重新调整也非常困难。针对这种情况，弹性原则是指系统的部署规模可以随着业务量的变化自动伸缩，无须根据事先的容量规划准备固定的硬件和软件资源，从而提高资源利用率，降低成本。

(3) 可观测原则。可观测性原则是指主动通过日志、链路跟踪和度量等手段，每次业务请求背后的多次服务调用的耗时、返回值和参数都清晰可见，甚至可以下钻到三方软件调用、SQL 请求、节点拓扑、网络响应等。具备可观测能力可以使运维、开发和业务人员实时掌握软件运行情况，并结合多个维度的数据指标，获得前所未有的关联分析能力，不断对业务健康度和用户体验进行数字化衡量和持续优化。

(4) 韧性原则。韧性原则是指当软件所依赖的软硬件组件出现各种异常时，软件需要表现出抵御能力，这些异常通常包括硬件故障、硬件资源瓶颈、业务流量超出软件设计能力、故障和灾难、软件 bug、黑客攻击等业务可用性带来致命影响的因素。韧性从多个维度诠释了软件持续提供业务服务的能力，核心目标是提升软件的 MTBF（Mean Time Between Failure, 平均无故障时间）。

(5) 自动化原则。自动化原则是指通过多种技术手段和自动化交付工具，一方面标准化企业内部的软件交付过程，另一方面在标准化的基础上进行自动化，通过配置数据自描述和面向终态的交付过程，让自动化工具理解交付目标和环境差异，实现整个软件交付和运维的自动化。

三、论文中需要结合项目实际工作，详细论述在项目中是如何采用云原生架构进行系统的设计与实现的，并围绕云原生架构的设计原则，论述遇到了哪些实际问题，是采用何种方法解决的。

2020 年下半年试题 4

数据分片就是按照一定的规则，将数据集划分成相互独立、正交的数据子集，然后将数据子集分布到不同的节点上。通过设计合理的数据分片规则，可将系统中的数据分布在不同的物理数据库中，达到提升应用系统数据处理速度的目的。

请围绕“论数据分片技术及其应用”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承担的主要工作。

2. Hash 分片、一致性 Hash（Consistent Hash）分片和按照数据范围（Range Based）分片是三种常用的数据分片方式。请简要阐述三种分片方式的原理。

3. 具体阐述你参与管理和开发的项目采用了哪些分片方式，并具体说明其实现过程和应用效果。

写作要点

一、简要叙述所参与管理和开发的软件项目，需要明确指出在其中承担的主要任务和开展的主要工作。

二、三种分片方式的具体描述如下：

1. Hash 方式。

数据分片的 Hash 方式是基于哈希表的思想，即按照数据的某一特征（key）来计算哈希值，并将哈希值与系统中的节点建立映射关系，从而将哈希值不同的数据分布到不同的节点上。

按照 Hash 方式做数据分片，优点是映射关系非常简单，需要管理的元数据也非常之少，只需要记录节点的数目以及 Hash 方式即可。但是，Hash 方式的缺点也非常明显。首先，当加入或者删除一个节点的时候，大量的数据需要移动。其次，Hash 方式很难解决数据不均衡的问题，如原始数据的特征值分布不均匀，导致大量的数据集中到一个物理节点上；或者对于可修改的记录数据，单条记录的数据变大。

2. 一致性 Hash 方式。

一致性 hash 是将数据按照特征值映射到一个首尾相接的 Hash 环上，同时也将节点（按照 IP 地址或者机器名 Hash）映射到这个环上。对于数据，从数据在环上的位置开始，顺时针找到的第一个节点即为数据的存储节点。可以看到相比于 Hash 方式，一致性 Hash 方式需要维护的元数据额外包含了节点在环上的位置，但这个数据量是较小的。同时，一致性 Hash 在增加或者删除节点的时候，受到影响的数据是比较有限的，只会影响到 Hash 环上相应的节点，不会发生大规模的数据迁移。

3. 按照数据范围方式。

按照数据范围（Range Based）方式是按照关键值划分成不同的区间，每个物理节点负责一个或者多个区间。按照数据范围方式跟一致性 Hash 有相似之处，可以理解为物理节点在 Hash 环上的位置是动态变化的。在按照数据范围方式中，区间的大小不是固定的，每个数据区间的数据量与区间的大小也是没有关系的。比如说，一部分数据非常集中，那么区间大小应该会比较小的，即以数据量的大小为片段标准。在实际工程中，一个节点往往负责多个区间，每个区间成为一个块，每个块有

一个阈值，当达到这个阈值之后就会分裂成两个块。这样做的目的在于当有节点加入的时候快速达到均衡。

三、论文中需要结合项目实际工作，详细论述项目采用了哪些分片方式，并具体说明其实现过程和应用效果。

2019 年下半年试题 1

论软件设计方法及其应用

软件设计（Software Design，SD）根据软件需求规格说明书设计软件系统的整体结构、划分功能模块、确定每个模块的实现算法以及程序流程等，形成软件的具体设计方案。软件设计把许多事物和问题按不同的层次和角度进行抽象，将问题或事物进行模块化分解，以便更容易解决问题。分解得越细，模块数量也就越多，设计者需要考虑模块之间的耦合度。

请围绕“论软件设计方法及其应用”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. 详细阐述有哪些不同的软件设计方法，并说明每种方法的适用场景。
3. 详细说明你所参与的软件开发项目中，使用了哪种软件设计方法，具体实施效果如何。

文老师点评：

这个题目可以选，虽然不是当年押中的，不过不难写，就针对软件设计这个步骤，有三个大家都知的方向。

一是结构化设计，可以写概要设计、详细设计。

二是面向对象设计，可以写面向对象的设计原则、设计模式。

三是软件架构设计，可以写架构风格这些，不过这个可能不是太贴合题意，建议前两个任选一个。

其他参考资料如下：

软件设计方法包括：

（1）模型驱动设计。

模型驱动设计是一种系统设计方法，强调通过绘制图形化系统模型描述系统的技术和实现。通常从模型驱动分析中开发的逻辑模型导出系统设计模型，最终，系统设计模型将作为构造和实现新系统的蓝图。

（2）结构化设计。

结构化设计是一种面向过程的系统设计技术，它将系统过程分解成一个容易实现和维护的计算机程序模块。把一个程序设计成一个自顶向下的模块层次，一个模块就是一组指令：一个程序片段、程序块、子程序或者子过程，这些模块自顶向下按照各种设计规则和设计指南进行开发，模块需要满足高度内聚和松散耦合的特征。

（3）信息工程。

信息工程是一种用来计划、分析和设计信息系统的模型驱动的、以数据为中心的但对过程敏感的技术。信息工程模型是一些说明和同步系统的数据和过程的图形。信息工程的主要工具是数据模型图（物理实体关系图）。

（4）原型设计。

原型化方法是一种反复迭代过程，它需要设计人员和用户之间保持紧密的工作关系，通过构造一个预期系统的小规模的、不完整的但可工作的示例来与用户交互设计结果。原型设计方法鼓励并要求最终用户主动参与，这增加了最终用户对项目的信心和支持。原型更好地适应最终用户总是想改变想法的自然情况。原型是主动的模型，最终用户可以看到并与之交互。

（5）面向对象设计。

面向对象设计是一种新的设计策略，用于精炼早期面向对象分析阶段确定的对象需求定义，并定义新的与设计相关的对象。面向对象设计是面向对象分析的延伸，有利于消除“数据”和“过程”的分离。

（6）快速应用开发。

快速应用开发是一种系统设计方法，是各种结构化技术（特别是数据驱动的信息工程）与原型化技术和联合应用开发技术的结合，用以加速系统开发。快速应用开发要求反复地使用结构化技术和原型化技术来定义用户的需求并设计最终系统。

2019 年下半年试题 2

论软件系统架构评估及其应用

对于软件系统，尤其是大规模复杂软件系统而言，软件系统架构对于确保最终系统的质量具有十分重要的意义。在系统架构设计结束后，为保证架构设计的合理性、完整性和针对性，保证系统质量，降低成本及投资风险，需要对设计好的系统架构进行评估。架构评估是软件开发过程中的重要环节。

请围绕“软件系统架构评估及其应用”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. 详细阐述有哪些不同的软件系统架构评估方法，并从评估目标、质量属性和评估活动等方面论述其区别。
3. 详细说明你所参与的软件开发项目中，使用了哪种评估方法，具体实施过程和效果如何。

文老师点评：

这个题目毫无疑问应该选，是当年完全押中的论文题，而且也很好写，架构评估，可以写质量属性，引出质量属性效用树，然后任选一种架构评估方法，按步骤来写，上课时我们讲的是 ATAM 方法。

2019 年下半年试题 3

论数据湖技术及其应用

近年来，随着移动互联网、物联网、工业互联网等技术的不断发展，企业级应用面临的数据规模不断增大，数据类型异常复杂。针对这一问题，业界提出“数据湖（Data Lake）”这一新型的企业数据管理技术。数据湖是一个存储企业各种原始数据的大型仓库，支持对任意规模的结构化、半结构化和非结构化数据进行集中式存储，数据按照原有结构进行存储，无须进行结构化处理；数据湖中的数据可供存取、处理、分析及传输，支撑大数据处理、实时分析、机器学习、数据可视化等多种应用，最终支持企业的智能决策过程。

请围绕“数据湖技术及其应用”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. 详细阐述数据湖技术，并从主要数据来源、数据模式（Schema）转换时机、数据存储成本、数据质量、面对用户和主要支撑应用类型等 5 个方面详细论述数据湖技术与数据仓库技术的差异。
3. 详细说明你所参与的软件开发项目中，如何采用数据湖技术进行企业数据管理，并说明具体实施过程及应用效果。

文老师点评：

不建议选。这个完全是新技术的，不是专门从事这个行业的，扯不出一篇论文的，不选即可。相关资料如下：

数据仓库是一个优化的数据库，用于分析来自事务系统和业务线应用程序的关系数据。数据仓库技术需要事先定义数据结构和数据模式（Schema）以优化快速 SQL 查询，其中结果通常用于操作报告和分析。数据经过了清理、丰富和转换，因此可以充当用户可信任的“单一信息源”。与数据仓库不同，数据湖能够同时存储来自业务线应用程序的关系数据，以及来自移动应用程序、物联网设备和社交媒体的非关系数据。在进行数据捕获时，无须定义数据结构或数据模式（Schema）。数据湖支持用户对数据使用不同类型的分析（如 SQL 查询、大数据分析、全文搜索、实时分析和机器学习等），为企业智能决策提供支撑。

下面从主要数据来源、数据模式转换时机、数据存储成本、数据质量、面对用户和主要支撑应用类型等六个方面对数据湖技术和数据仓库技术进行比较：

特性	数据湖	数据仓库
主要数据来源	来自物联网设备、互联网、移动应用程序、社交媒体和企业应用程序的结构化、半结构化和非结构化数据	来自事务系统、运营数据库和业务线应用程序的结构化数据
数据模式转换时机	数据进入数据湖时不进行模式转换, 在进行实际数据分析时才进行模式转换	在进入数据仓库之前(需要提前设计数据仓库的 Schema)
数据存储成本	通常基于非关系型数据库, 数据存储成本相对较低	通常基于关系型数据库, 数据存储成本高
数据质量	原始的、未经处理的数据	可作为重要事实依据的高质量数据
面对用户	业务分析师、应用开发人员和数据科学家	业务分析师
主要支撑应用类型	机器学习、预测分析、数据发现和分析	批处理报告、商务智能 (BI) 和数据可视化

2019 年下半年试题 4

论负载均衡技术在 Web 系统中的应用

负载均衡技术是提升 Web 系统性能的重要方法。利用负载均衡技术, 可将负载(工作任务)进行平衡、分摊到多个操作单元上执行, 从而协同完成工作任务, 达到提升性能的目的。

请围绕“负载均衡技术在 Web 系统中的应用”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. 详细阐述常见的三种负载均衡算法, 说明算法的基本原理。
3. 详细说明你所参与的软件开发项目中, 如何基于负载均衡算法实现 Web 应用系统的负载均衡。

文老师点评:

不建议选。当年考试也有不少人选了这个题目, 这个题目不算新技术, 也是上课讲过的, 但是上课讲的主要是负载均衡的不同技术, 应用层、传输层如何实现负载均衡, 而不是实现负载均衡的算法, 有明显的区别, 所以选这一题的基本都是低分跑题了, 相关资料如下:

现有的负载均衡算法主要分为静态和动态两类。静态负载均衡算法以固定的概率分配任务, 不考虑服务器的状态信息, 如轮转算法、随机法等; 动态负载均衡算法以服务器的实时负载状态信息来决定任务的分配, 如最小连接法等。

(1) 轮询法。

轮询法就是将用户的请求轮流分配给服务器, 就像是挨个数数, 轮流分配。这种算法比较简单, 具有绝对均衡的优点, 但是也正是因为绝对均衡, 它必须付出很大的代价, 例如它无法保证分配任务的合理性, 无法根据服务器承受能力来分配任务。

(2) 随机法。

随机法是随机选择一台服务器来分配任务。它保证了请求的分散性达到了均衡的目的。同时它是没有状态的，不需要维持上次的选择状态和均衡因子。但是随着任务量的增大，它的效果趋向轮询后也会具有轮询法的部分缺点。

（3）最小连接法。

最小连接法将任务分配给此时具有最小连接数的节点，因此它是动态负载均衡算法。一个结点收到一个任务后连接数就会加 1，当结点发生故障时就将结点权值设置 0，不再给结点分配任务。最小连接法适用于各个结点处理的性能相似的情形。任务分发单元会将任务平滑分配给服务器。但当服务器性能差距较大时，就无法达到预期的效果。因为此时连接数并不能准确表明处理能力，连接数小而自身性能很差的服务器可能不及连接数大而自身性能极好的服务器。所以在这个时候就会导致任务无法准确地分配到剩余处理能力强的机器上。

2018 年下半年试题 1

论软件开发过程 RUP 及其应用

RUP (Rational Unified Process) 是 IBM 公司一款软件开发过程产品，它提出了一整套以 UML 为基础的开发准则，用以指导软件开发人员以 UML 为基础进行软件开发。RUP 汲取了各种面向对象分析与设计方法的精华，提供了一个普遍的软件过程框架，可以适应不同的软件系统、应用领域、组织类型和项目规模。

请围绕“论软件开发过程 RUP 及其应用”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担任的主要工作。
2. 详细论述软件开发过程产品 RUP 所包含的 4 个阶段以及 RUP 的基本特征。
3. 结合你所参与管理和开发的软件项目，详细阐述 RUP 在该项目中的具体实施内容，包括核心 workflows 的选择、制品的确定、各个阶段之间的演进及迭代计划以及 workflow 内部结构的规划等。

2018 年下半年试题 2

论软件体系结构的演化

软件体系结构的演化是在构件开发过程中或软件开发完毕投入运行后，由于用户需求发生变化，就必须相应地修改原有软件体系结构，以满足新的变化了的软件需求的过程。体系结构的演化是一个复杂的、难以管理的问题。

请围绕“论软件体系结构的演化”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. 软件体系结构的演化是使用系统演化步骤去修改系统，以满足新的需求。简要论述系统演化的 6 个步骤。
3. 具体阐述你参与管理和开发的项目是如何基于系统演化的 6 个步骤完成软件体系结构演化的。

2018 年下半年试题 3

论面向服务架构设计及其应用

面向服务架构（Service-Oriented Architecture, SOA）是一种应用框架，将日常的业务应用划分为单独的业务功能服务和流程，通过采用良好定义的接口和标准协议将这些服务关联起来。通过实施基于 SOA 的系统架构，用户可以构建、部署和整合服务，无需依赖应用程序及其运行平台，从而提高业务流程的灵活性，帮助企业加快发展速度，降低企业开发成本，改善企业业务流程的组织和资产重用。

请围绕“论面向服务架构设计及其应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与分析和开发的软件系统开发项目以及你所担任的主要工作。
2. 说明面向服务架构的主要技术和标准，详细阐述每种技术和标准的具体内容。
3. 详细说明你所参与的软件系统开发项目中，构建 SOA 架构时遇到了哪些问题，具体实施效果如何。

2018 年下半年试题 4

论 NoSQL 数据库技术及其应用

随着互联网 web2.0 网站的兴起，传统关系数据库在应对 web2.0 网站，特别是超大规模和高并发的 web2.0 纯动态 SNS 网站上已经显得力不从心，暴露了很多难以克服的问题，而非关系型的数据库则由于其本身的特点得到了非常迅速的发展。

NoSQL (Not only SQL) 的产生就是为了解决大规模数据集合及多种数据类型带来的挑战，尤其是大数据应用难题。目前 NoSQL 数据库并没有一个统一的架构，根据其所采用的数据模型可以分为 4 类：键值（Key-Value）存储数据库、列存储数据库、文档型数据库和图（Graph）数据库。

请围绕“NoSQL 数据库技术及其应用”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担任的主要工作。
2. 详细论述常见的 NoSQL 数据库技术及其所包含的主要内容，并说明 NoSQL 数据库的主要适用场景。
3. 结合你具体参与管理和开发的实际项目，说明具体采用哪种 NoSQL 数据库技术，并说明架构设计过程及其应用效果。

2017 年下半年试题 1

论软件系统建模方法及其应用

软件系统建模 (Software System Modeling) 是软件开发中的重要环节，通过构建软件系统模型可以帮助系统开发人员理解系统、抽取业务过程和管理系统的复杂性，也可以方便各类人员之间的交

流。软件系统建模是在系统需求分析和系统实现之间架起的一座桥梁，系统开发人员按照软件系统模型开发出符合设计目标的软件系统，并基于该模型进行软件的维护和改进。请围绕“论软件系统建模方法及其应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与的软件系统开发项目以及你所担任的主要工作。
2. 说明软件系统开发中常用的建模方法有哪几类？阐述每种方法的特点及其适用范围。
3. 详细说明你所参与的软件系统开发项目中，采用了哪些软件系统建模方法，具体实施效果如何。

2017 年下半年试题 2

论软件架构风格

软件体系结构风格是描述某一特定应用领域中系统组织方式的惯用模式。体系结构风格定义一个系统家族，即一个体系结构定义一个词汇表和一组约束。词汇表中包含一些构件和连接件类型，而这组约束指出系统是如何将这些构件和连接件组合起来的。体系结构风格反应了领域中众多系统所共有的结构和语义特性，并指导如何将各个模块和子系统有效地组织成一个完整的系统。请围绕“论软件架构风格”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与分析和设计的软件系统开发项目以及你所担任的主要工作。
2. 软件系统开发中常用的软件架构风格有哪些？详细阐述每种风格的具体含义。
3. 详细说明你所参与分析和设计的软件系统是采用什么软件架构风格的，并分析采用该架构风格设计的原因。

2017 年下半年试题 3

论无服务器架构及其应用

近年来，随着信息技术的迅猛发展和应用需求的快速更迭，传统的多层企业应用系统架构面临越来越多的挑战，已经难以适应这种变化。在这一背景下，无服务器架构(Serverless Architecture)逐渐流行，它强调业务逻辑由事件触发，具有短暂的生命周期，运行于无状态的轻量级容器中，并且由第三方代为管理。采用无服务器架构，业务逻辑以功能即服务(Function As a Service, FAAS)的方式形成多个相互独立的功能组件，以标准接口的形式向外提供服务；同时，不同功能组件间的逻辑组织代码将存储在通用的基础设施管理平台中，业务代码仅在调用时才激活运行，当响应结束后占用的资源便会释放。

请围绕“无服务器架构及其应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与分析和设计的软件系统开发项目以及你所担任的主要工作。
2. 与传统的企业应用系统相比较，基于无服务器架构的应用系统具有哪些特点，请例举至少 3 个特点，并进行解释。

3. 结合你具体参与分析和设计的软件开发项目, 描述该软件的架构, 说明该架构是如何是采用无服务器架构模式的, 并说明在采用无服务器架构后软件开发过程中遇到的实际问题和解决方案。

2017 年下半年试题 4

论软件质量保证及其应用

软件质量保证 (Software Quality Assurance, SQA) 是指为保证软件系统或软件产品充分满足用户要求的质量而进行的有计划、有组织的活动, 这些活动贯穿于软件生产的整个生命周期。质量保证人员负责质量保证的计划、监督、记录、分析及报告工作, 辅助软件开发人员得到高质量的最终产品。

请围绕“软件质量保证及其应用”论题, 依次从以下三个方面进行论述。概要叙述你参与管理和开发的软件项目以及你在其中所担任的主要工作。详细论述软件质量保证中常见的活动有哪些? 阐述每个活动的主要内容。结合你具体参与管理和开发的实际项目, 说明是如何实施软件质量保证的各项活动, 说明其实施过程及应用效果。

2016 年下半年试题 1

论软件系统架构评估

对于软件系统, 尤其是大规模的复杂软件系统来说, 软件的系统架构对于确保最终系统的质量具有十分重要的意义, 不恰当的系统架构将给项目开发带来高昂的代价和难以避免的灾难。对一个系统架构进行评估, 是为了: 分析现有架构存在的潜在风险, 检验设计中提出的质量需求, 在系统被构建之前分析现有系统架构对于系统质量的影响, 提出系统架构的改进方案。架构评估是软件开发过程中的重要环节。请围绕“论软件系统架构评估”论题, 依次从以下三个方面进行论述。

1. 概要叙述你所参与架构评估的软件系统, 以及在评估过程中所担任的主要工作。
2. 分析软件系统架构评估中所普遍关注的质量属性有哪些? 详细阐述每种质量属性的具体含义。
3. 详细说明你所参与的软件系统架构评估中, 采用了哪种评估方法, 具体实施过程和效果如何。

2016 年下半年试题 2

论软件设计模式及其应用

软件设计模式 (Software Design Pattern) 是一套被反复使用的、多数人知晓的、经过分类编目的代码设计经验的总结。使用设计模式是为了重用代码以提高编码效率、增加代码的可理解性、保证代码的可靠性。软件设计模式是软件开发中的最佳实践之一, 它经常被软件开发人员在面向对象软件开发过程中所采用。项目中合理地运用设计模式可以完美地解决很多问题, 每种模式在实际

应用中都有相应的原型与之相对，每种模式都描述了一个在软件开发中不断重复发生的问题，以及对应该原型问题的核心解决方案。请围绕“论软件设计模式及其应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与分析和开发的软件系统，以及你在项目中所担任的主要工作。
2. 说明常用的软件设计模式有哪几类？阐述每种类型特点及其所包含的设计模式。
3. 详细说明你所参与的软件系统开发项目中，采用了哪些软件设计模式，具体实施效果如何。

2016 年下半年试题 3

论数据访问层设计技术及其应用 在信息系统的开发与建设中，分层设计是一种常见的架构设计方法，区分层次的目的是为了实现在“高内聚低耦合”的思想。分层设计能有效简化系统复杂性，使设计结构清晰，便于提高复用能力和产品维护能力。一种常见的层次划分模型是将信息系统分为表现层、业务逻辑层和数据访问层。信息系统一般以数据为中心，数据访问层的设计是系统设计中的重要内容。数据访问层需要针对需求，提供对数据源读写的访问接口；在保障性能的前提下，数据访问层应具有良好的封装性、可移植性，以及数据库无关性。请围绕“论数据访问层设计技术及其应用”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与数据访问层设计有关的软件项目，以及你在其中所担任的主要工作。
2. 详细论述常见的数据访问层设计技术及其所包含的主要内容。
3. 结合你参与管理和开发的实际项目，具体说明采用了哪种数据访问层设计技术，并叙述具体实施过程以及应用效果。

2016 年下半年试题 4

论微服务架构及其应用

近年来，随着互联网行业的迅猛发展，公司或组织业务的不断扩张，需求的快速变化以及用户量的不断增加，传统的单块（Monolithic）软件架构面临着越来越多的挑战，已逐渐无法适应互联网时代对软件的要求。在这一背景下，微服务架构模式（Microservice Architecture Pattern）逐渐流行，它强调将单一业务功能开发成微服务的形式，每个微服务运行在一个进程中；采用 HTTP 等通用协议和轻量级 API 实现微服务之间的协作与通信。这些微服务可以使用不同的开发语言以及不同数据存储技术，能够通过自动化部署工具独立发布，并保持最低限制的集中式管理。

请围绕“论微服务架构及其应用”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、采用微服务架构的软件开发项目及在其中所担任的主要工作。
2. 与单块架构相比较，微服务架构有哪些特点？请列举至少 4 个特点并进行说明。

3. 结合你参与管理和开发的软件开发项目，描述该软件的架构，说明该架构是如何采用微服务架构模式的，并说明在采用微服务架构后，在软件开发过程中遇到的实际问题和解决方案。

2015 年下半年试题 1

论应用服务器基础软件

应用服务器是在当今基于互联网的企业级应用迅速发展，电子商务应用出现并快速膨胀的需求下产生的一种新技术。在分布式、多层结构及基于组件和服务端程序设计的企业级应用开发中，应用服务器提供的是一个开发、部署、运行和管理、维护的平台，提供软件“集群”功能，可以让多个不同的异构服务器协同工作、相互备份，以满足企业级应用所需要的高可用性、高性能、高可靠性和可伸缩性等实际需求。应用服务器技术的出现，能够加快应用的开发速度，减少应用的开发量。通过隔离底层细节，便于商业逻辑的实现与扩展，同时也为企业应用提供现成的、稳定的、灵活的、成熟的基础架构。

请以“应用服务器基础软件”为题，依次从以下三个方面进行论述：

1. 概要叙述你参与分析和开发的软件系统开发项目以及你所担任的主要工作。
2. 论述并分析应用服务器在软件设计、开发、部署、运行和管理阶段，应该提供哪些核心功能？
3. 详细说明你所参与的软件系统开发项目，采用了哪种应用服务器，在软件开发、部署和运行阶段，具体实施效果如何。

2015 年下半年试题 2

论软件系统架构风格

系统架构风格（System Architecture Style）是描述某一特定应用领域中系统组织方式的惯用模式。架构风格定义了一个词汇表和一组约束，词汇表中包含一些构件和连接件类型，而这组约束指出系统是如何将这些构件和连接件组合起来的。软件系统架构风格反映了领域中众多软件系统所共有的结构和语义特性，并指导如何将各个模块和子系统有效地组织成一个完整的系统。软件系统架构风格的共有部分可以使得不同系统共享同一个实现代码，系统能够按照常用的、规范化的方式来组织，便于不同设计者很容易地理解系统架构。

请以“软件系统架构风格”论题，依次从以下三个方面进行论述：

1. 概要叙述你参与分析和开发的软件系统开发项目以及你所担任的主要工作。
2. 分析软件系统开发中常用的软件系统架构风格有哪些？详细阐述每种风格的具体含义。
3. 详细说明在你所参与的软件系统开发项目中，采用了哪种软件系统架构风格，具体实施效果如何。

2015 年下半年试题 3

论面向服务的架构及其应用

面向服务的架构（Service-Oriented Architecture, SOA）是一种组件模型，把应用程序中的不同功能单元（即服务）通过这些服务之间定义良好的接口和契约联系起来，使得这些系统中的服务能够以一种统一和通用的方式进行交互。从应用角度看，SOA 是一种应用框架，它关注企业日常的业务应用，将其划分为单独的业务功能和流程，并抽象为服务，用户和系统开发人员可以构建、部署和整合这些服务，无需依赖特定的应用程序及应用平台，从而提高企业业务流程的灵活性。SOA 有助于实现更多的信息资产重用、更轻松地管理和更快地应用开发与部署。

请以“面向服务的架构及其应用”为题，依次从以下三个方面进行论述：

1. 概要叙述你参与实施的、基于面向服务架构的软件开发项目以及所担任的主要工作。
2. 指出 SOA 技术参考架构中都包含哪些服务类别，并对每类服务的定义和作用进行简要说明。
3. 详细阐述你的项目是如何以面向服务的架构为指导进行实施的，在实施过程中遇到了哪些问题，是如何解决的。

2015 年下半年试题 4

论企业集成平台的技术与应用 企业集成平台是一个支持复杂信息环境下信息系统开发、集成和协同运行的软件支撑环境。它基于各种企业经营业务的信息特征，在异构分布环境（操作系统、网络、数据库）下为应用提供一致的信息访问和交互手段，对其上运行的应用进行管理，为应用提供服务，并支持企业信息环境下各特定领域的应用系统的集成。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。

请以“企业集成平台的技术与应用”为题，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企业集成平台相关的软件项目以及你在其中所担任的主要工作。
2. 简要说明企业集成平台的基本功能及企业集成的关键技术，并结合项目实际情况，阐述该项目所选择的关键技术及其原因。
3. 结合你具体参与管理和开发的实际项目，举例说明所采用的企业集成架构设计技术的具体实施方式及过程，并详细分析其实现效果。

2014 年下半年试题 1

论软件需求管理

软件需求管理是一个对系统需求变更了解和控制的过程。需求管理过程与需求开发过程相互关联，初始需求导出的同时就要形成需求管理规划，一旦启动了软件开发过程，需求管理活动就紧密相伴。

需求管理过程中主要包含变更控制、版本控制、需求跟踪和需求状态跟踪等 4 项活动，其目标是为项目管理人员建立一个软件需求基线，并保持软件计划、产品和活动与软件需求的一致性。

请以“软件需求管理”为题，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担任的主要工作。
2. 详细描述需求管理过程中各个活动中的主要工作。
3. 详细说明你所参与的软件开发项目中，是如何进行软件需求管理的，实施的具体效果如何。

2014 年下半年试题 2

论非功能性需求对企业应用架构设计的影响

企业应用架构(Enterprise Application Architecture) 描述了企业 IT 系统的功能和技术实现内容，它在企业信息化建设中起到了统一规划、承上启下的作用，向上承接了企业战略发展方向和业务模式，向下规划和指导企业各 IT 系统的定位和功能。企业应用架构包括了企业的应用架构蓝图、架构标准、系统的边界和定义、系统间的关联关系等。其中非功能性需求是进行企业应用架构设计时需要重点考虑的因素，不同类型的非功能性需求从不同侧面影响应用系统的架构设计。

请以“非功能性需求对企业应用架构设计的影响”为题，依次从以下三个方面进行论述。

1. 概要叙述你参与分析和开发的企业应用系统项目以及你所担任的主要工作。
2. 分析在企业应用架构设计中应该考虑哪些非功能性需求，详细阐述这些非功能性需求是如何影响架构设计的。
3. 详细说明你所参与的企业应用系统项目中，在进行系统架构设计时，考虑了哪些非功能性需求，如何通过架构设计满足了系统的这些非功能性需求。

2014 年下半年试题 3

论软件的可靠性设计 现代军事和商用系统中，随着系统中软件成分的不断增加，系统对软件的依赖性越来越强。软件可靠性已成为软件设计过程中不可或缺的重要组成部分。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，由此提出了可靠性设计的概念。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，使程序设计在兼顾用户的功能和性能需求的同时，全面满足软件的可靠性要求。

请以“软件的可靠性设计”为题，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担任的主要工作。
2. 简要说明目前比较主流的软件可靠性设计技术，结合项目实际情况，阐述所选择的可靠性设计技术及其原因。
3. 结合你具体参与管理和开发的实际项目，举例说明所选取的软件可靠性技术的具体实施过程，并详细分析实施效果。

2014 年下半年试题 4

论网络安全体系设计

随着社会信息化的普及，计算机网络已经在各行各业得到了广泛的应用。目前，绝大多数业务处理几乎完全依赖计算机和网络执行，各种重要数据如政府文件、工资档案、财务账目和人事档案等均依赖计算机和网络进行存储与传输。另一方面，针对计算机和网络的攻击活动日益猖獗，网络安全已经成为当前社会的主要安全问题之一。在上述背景下，国家标准《信息处理系统工程开放系统互联基本参考模型——第二部分：安全体系结构》(GB/T 9387.2-1995)定义了基于 OSI 参考模型 7 层协议之上的信息安全体系，其核心内容是：为了保证异构计算机进程与进程之间远距离交换信息的安全，定义了认证服务、访问控制服务、数据机密性服务、数据完整性服务和抗抵赖性服务等 5 大类安全服务，以及提供这些服务的 8 类安全机制及相应的 OSI 安全管理，并根据具体系统适当配置于 OSI 模型的 7 层协议之中。

请以“网络安全体系设计”为题，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承担的主要工作，并详细阐述该软件系统在网络安全方面的要求。
2. 请对 GB/T 9387.2-1995 中定义的 5 大类安全服务进行描述，阐述每类安全服务的定义和主要实现手段。
3. 请结合项目实际，具体阐述你在项目中实现了上述 5 大类安全服务中的哪些服务，具体运用了哪些实现手段。

2013 年下半年试题 1

论软件架构建模技术与应用

软件架构用来处理软件高层次结构的设计和实现，它以精心选择的形式将若干结构元素进行装配，从而满足系统的主要功能和性能需求。软件架构设计的首要问题是如何表示软件架构，即如何对软件架构建模。根据建模的侧重点不同，可以将软件架构模型分为结构模型、框架模型、动态模型、过程模型和功能模型。Kruchten 在 1995 年提出了“4+1”视图模型，将 5 种模型有机地统一在了一起。

请围绕“软件架构建模技术与应用”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. 简要叙述“4+1”视图模型的主要内容。结合你参与项目的实际情况，详细说明该项目需求及所涉及的软件架构（包括使用到的视图模型、创建的架构模型及使用的建模工具等）。
3. 说明该项目软件架构的实施效果，分析其是否满足了项目的需求并说明原因。

2013 年下半年试题 2

论企业应用系统的分层架构风格

软件架构风格是描述一类特定应用领域中系统组织方式的惯用模式，反映了领域中诸多系统所共有的结构特征和语义特征，并指导如何将各个模块和子系统有效组织成一个完整的系统。分层架构是一种常见的软件架构风格，能够有效简化设计，使得设计的系统结构清晰，便于提高复用能力和产品维护能力。由于大量企业应用系统都由界面呈现、业务逻辑、数据存储三类功能构成，因此广泛采用分层架构风格进行系统设计。

请围绕“企业应用系统的分层架构风格”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企业应用系统建设项目以及你在其中所承担的主要工作。
2. 请结合项目实际情况，指出应用系统都有哪些层次以及每个层次的主要功能。
3. 请结合项目实际情况，指出设计每个层次时需要注意的问题及相应的解决方案。

2013 年下半年试题 3

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。

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。

2013 年下半年试题 4

论分布式存储系统架构设计

分布式存储系统（Distributed Storage System）通常将数据分散存储在多台独立的设备上。传统的网络存储系统采用集中的存储服务器存放所有数据，存储服务器成为系统性能的瓶颈，也是可靠性和安全性的焦点，不能满足大规模存储应用的需要。分布式存储系统采用可扩展的系统结构，

利用多台存储服务器分担存储负荷，利用位置服务器定位存储信息，它不但提高了系统的可靠性、可用性和存取效率，还易于扩展。

请围绕“分布式存储系统架构设计”论题，依次从以下三个方面进行论述。

- 1、概要叙述你参与分析和开发的分布式存储系统项目以及你所承担的主要工作。
- 2、简要说明在分布式存储系统架构设计中所使用的分布式存储技术及其实现机制，详细叙述你在具体项目中选用了哪种分布式存储技术，说明其原因和实施效果。
- 3、冗余是提高分布式存储系统可靠性的主要方法，通常在分布式存储系统设计中可采用哪些冗余技术来提升系统的可靠性？你在具体项目中选用了哪种冗余技术？说明其原因和实施效果。

2012年下半年试题 1

论企业信息化规划的实施与应用

企业信息化建设是一项长期而艰巨的任务，不可能在短时间内完成。信息化规划是企业信息化建设的纲领和向导，是信息系统设计和实施的前提和依据。信息化规划以整个企业的发展目标和战略、企业各部门的目标与功能为基础，同时结合行业信息化方面的实践和对信息技术发展趋势的掌握，制定出企业信息化远景、目标和发展战略，从而达到全面、系统地指导企业信息化建设的目的。请围绕“企业信息化规划的实施与应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与的企业信息化规划项目以及你在其中所担任的主要工作。
2. 简要叙述企业信息化规划的主要内容。结合你参与的项目的实际情况，详细分析有关企业的信息化规划目标及规划的具体内容。
3. 说明你所参与实施的企业信息化规划的步骤及效果，介绍其是否达到了预期的目标并分析原因。

2012年下半年试题 2

论决策支持系统的开发与应用

决策支持系统（Decision Support Systems, DSS）是以管理科学、运筹学、控制论和行为科学为基础，以计算机技术、仿真技术和信息技术为手段，以人机交互方式进行半结构化和非结构化决策的信息系统。它调用各种信息资源，并提供各种分析工具，为决策者提供分析问题、建立模型、模拟决策过程和方案的环境，帮助决策者提高决策水平和质量。决策支持系统在许多领域得到了广泛的应用，已成为许多行业经营管理中一个不可缺少的现代化支持工具。

请围绕“决策支持系统的开发与应用”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决策支持系统项目以及在其中所担任的主要工作。
2. 简要叙述决策支持系统包含的典型组成部件及对应的基本功能。说明在建立决策支持系统时需解决的一般关键问题。

3. 说明你所参与管理和开发的决策支持系统的应用场合以及对决策结果的要求，具体阐述在开发过程中所采用的关键技术、实施过程和实际应用的效果。

2012 年下半年试题 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。

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。

2012 年下半年试题 4

论基于架构的软件设计方法及应用

基于架构的软件设计（Architecture-Based Software Design, ABSD）方法以构成软件架构的商业、质量和功能需求等要素来驱动整个软件开发过程。ABSD 是一个自顶向下，递归细化的软件开发方法，它以软件系统功能的分解为基础，通过选择架构风格实现质量和商业需求，并强调在架构设计过程中使用软件架构模板。采用 ABSD 方法，设计活动可以从项目总体功能框架明确后就开始，因此该方法特别适用于开发一些不能预先决定所有需求的软件系统，如软件产品线系统或长生命周期系统等，也可为需求不能在短时间内明确的软件项目提供指导。

请围绕“基于架构的软件开发方法及应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与开发的、采用 ABSD 方法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，详细说明采用 ABSD 方法进行软件开发时，需要经历哪些开发阶段？每个阶段包括哪些主要活动？
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。

2011 年下半年试题 1

论模型驱动架构在系统开发中的应用

模型驱动架构(Model Driven Architecture, MDA)是对象管理组织提出的软件体系架构方法学，它基于 UML 以及一系列工业标准，能够支持基于可视化模型驱动的软件设计、内容存储与交换。MDA 核心思想是抽象出与实现技术无关、完整描述业务功能的核心平台无关模型 (PIM)，然后针对不同实现技术制定多个映射规则，通过映射规则和辅助工具将 PIM 转换成与具体实现技术有关的平台相关模型 (PSM)，最后完成 PSM 到代码的转换。通过 PIM 和 PSM，MDA 分离业务建模与底层实现技术，降低技术变迁对业务模型带来的影响。

请围绕“模型驱动架构在系统开发中的应用”论题，依次从以下三个方面进行论述。

- (1) 简要叙述你参与管理和开发的、与 MDA 相关的软件开发项目以及你所担任的主要工作。
- (2) 简要分析模型驱动架构能够为软件开发带来哪些好处，详细论述采用模型驱动架构进行开发的过程。
- (3) 具体阐述你参与管理和开发的项目中使用模型驱动架构的情况与实际开发效果。

2011 年下半年试题 2

论企业集成平台的架构设计

企业集成平台是一个支持复杂信息环境下信息系统开发、集成和协同运行的软件支撑环境，它基于企业各种经营业务的信息特征，在异构分布环境（操作系统、网络、数据库）下为应用提供一致的信息访问和交互手段，对其上运行的应用进行管理，为应用提供服务，并支持各种特定领域应用系统的集成。

请围绕“企业集成平台的架构设计”论题，依次从以下三个方面进行论述。

- 1、简要叙述你参与管理和开发的企业集成平台项目以及你在其中所承担的主要工作。
- 2、请说明企业集成平台的基本功能，并结合项目实际，详细说明所设计的企业集成平台的架构，以及实现时用到了哪些关键技术。
- 3、具体说明所设计的企业集成平台的使用情况，最终实施效果如何。

2011 年下半年试题 3

论企业架构管理与应用

企业架构管理 (Enterprise Architecture Management, EAM) 从功能、应用、信息和技术四个层面定义了企业应用系统的结构，并通过业务需求驱动开发过程，为企业应用系统的开发提供标准和指导。EAM 将企业的业务和技术需求联系在一起，以管理业务变更为核心，强调业务与技术对齐，构建一个高内聚、动态的企业应用解决方案。EAM 能够帮助企业识别可以提高运营效率的潜在领域，

有助于企业建立从战略到解决方案交付的各种关系，识别技术解决方案中最优的业务成果，能够在业务重组、兼并、收购和其他业务变更计划中为企业最大化地节约成本，降低相关风险。

请围绕“企业架构管理与应用”论题，依次从以下三个方面进行论述。

(1) 简要叙述你参与实施的企业应用系统的开发背景与总体需求、系统所采用的技术体制、实施企业软件架构管理的动机与期望以及你所承担的实际工作。

(2) 结合项目实际，简要阐述企业架构管理包含哪些方面的内容，每个方面包括哪些主要活动。

(3) 阐述你在实施企业架构管理的过程中都遇到了哪些实际问题，以及解决这些问题的方法和过程。

2011年下半年试题4

论软件需求获取技术及应用

软件需求是指用户对新系统在功能、行为、性能、设计约束等方面的期望。软件需求获取是一个确定和理解不同的项目干系人的需求和约束的过程。需求获取是否科学、准备充分，对获取的结果影响很大，这是因为大部分用户无法完整地描述需求，而且也不可能看到系统的全貌。因此，掌握各种不同的需求获取技术，并且熟练地在实践中运用它，并与用户有效合作，是十分重要的。

请围绕“需求获取技术及应用”论题，依次从以下三个方面进行论述。

1、简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、详细说明目前有哪些比较常用的需求获取技术？说明每种需求获取技术的基本方法。

3、详细论述在你参与分析和开发的软件项目中所采取的需求获取技术以及选取这些技术的原因，并说明需求获取的具体实施步骤。

2010年下半年试题1

论软件的静态演化和动态演化及其应用

软件演化 (Software Evolution) 是指软件在其生命周期内的更新行为和过程。演化是一系列贯穿软件生命周期始终的活动，系统需求改变、功能实现增强、新功能加入、软件架构改变、软件缺陷修复、运行环境改变均要求软件系统能够快速适应变化，具有较强的演化能力。软件静态演化 (Static Evolution) 和动态演化 (Dynamic Evolution) 是目前软件演化的两种重要类型。

请围绕“软件的静态演化和动态演化及其应用”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担任的主要工作。

2. 请分别对软件静态演化和动态演化的特点进行论述，说明两种软件演化类型各自的优缺点及其应用场合，并举例说明各自的常见演化技术手段。

3. 具体阐述你参与管理和开发的项目中所进行的软件演化活动的特点、演化的类型，以及所采取的对应演化技术手段，说明具体实施过程以及实际应用的效果。

2010 年下半年试题 2

论数据挖掘技术的应用

随着信息技术的高速发展，各组织机构积累的数据量急剧增长。如何从海量的数据中提取有用的知识成为当务之急。数据挖掘（Data Mining）就是为顺应这种需要应运而生发展起来的数据处理技术，是知识发现的关键步骤。数据挖掘就是从大量的、不完全的、有噪声的、模糊的、随机的实际应用数据中，提取隐含在其中的、人们事先不知道的、但又是潜在有用的信息和知识的过程。请围绕“论数据挖掘技术的应用”论题，依次对以下三个方面进行论述。

1. 概要叙述你参与分析和开发的软件项目以及你所担任的主要工作。
2. 数据挖掘的主要任务是什么？具体论述你在项目中使用数据挖掘技术所解决的问题。
3. 数据挖掘的方法主要有哪些？分析并讨论你所选择的数据挖掘方法，简述其具体实现过程和实际应用效果。

2010 年下半年试题 3

论大规模分布式系统缓存设计策略

大规模分布式系统通常需要利用缓存技术减轻服务器负载、降低网络拥塞、增强系统可扩展性。缓存技术的基本思想是将客户最近经常访问的内容在缓存服务器中存放一个副本，当该内容下次被访

问时，不必建立新的数据请求，而是直接由缓存提供。良好的缓存设计，是一个大规模分布式系统能够正常、高效运行的必要前提。在进行大规模分布式系统开发时，必须从一开始就针对应用需求和场景对系统的缓存机制进行全面考虑，设计一个可伸缩的系统缓存架构。

请围绕“大规模分布式系统缓存设计策略”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与实施的大规模分布式系统开发项目以及你所担任的主要工作。
2. 从不同的用途和应用场景考虑，请详细阐述至少两种常见的缓存工作模式，并说明每种工作模式的适应场景。
3. 阐述你在设计大规模分布式系统的缓存机制时遇到了哪些问题，如何解决。

2010 年下半年试题 4

论软件可靠性评价

软件可靠性评价是指选用和建立合适的可靠性数学模型，运用统计技术和其他手段，对软件可靠性测试和系统运行期间的软件失效数据（也可能包含软件生命周期内其他可靠性数据）进行处理，并评估和预测软件可靠性的过程。软件可靠性评价是软件可靠性活动的重要组成部分，既可在软件

开发过程实施，也可针对最终软件系统实施。软件可靠性评价的难点在于软件可靠性模型的选择和软件可靠性数据的收集与处理。

请围绕“软件可靠性评价”论题，依次从以下三个方面进行论述。

1. 简要概述你参与实施的软件开发项目以及你承担的主要工作。
2. 说明你在项目实施过程中所选择的软件可靠性模型，并论述在软件可靠性模型选择时应该考虑的主要因素。
3. 收集软件可靠性数据时经常遇到的问题有哪些？简述你收集软件可靠性数据时所遇到的具体问题及解决的方法。