

第一章 系统工程与信息系统基础

1.1 系统工程

知识点1: 系统工程方法

- 【系统工程】是一种组织管理技术。
- 【系统工程】是为了最好的实现系统的目的，对系统的组成要素、组织结构、信息流、控制机构进行分析研究的科学方法。
- 【系统工程】从整体出发、从系统观念出发，以求【整体最优】。
- 【系统工程】利用计算机作为工具，对系统的结构、元素、信息和反馈等进行分析，以达到最优规划、最优设计、最优管理和最优控制的目的。
- 【系统工程方法】是一种现代的科学决策方法。

系统工程方法	关键点
霍尔三维结构 “硬科学”方法论	逻辑维：逻辑维即解决问题的逻辑过程。 时间维：时间维即是工作进程。 知识维：知识维即是专业科学知识。 应用场景：组织和管理大型工程建设项目。
切克兰德方法 “软科学”方法论	核心不是“最优化”，而是“比较”和“探寻”。 7步骤：认识问题、根底定义、建立概念模型、比较及探寻、选择、设计与实施、评估与反馈
并行工程方法	“制造过程”与“支持过程”并行。 强调三个方面：产品设计开发期间，最快速度按质完成；各项工作问题协调解决；适当的信息系统工具。
综合集成法	钱学森命名，【简单系统】和【巨系统】 四原则：整体论原则、相互联系原则、有序性原则、动态原则
WSR系统方法	实践准则：【懂物理】－【明事理】－【通人理】



考点1：霍尔三维结构

【2018年下】系统工程利用计算机作为工具，对系统的结构、元素、()和反馈等进行分析，以达到最优()、最优设计、最优管理和最优控制的目的。霍尔(A. D. Hall)于1969年提出了系统方法的三维结构体系，通常称为霍尔三维结构，这是系统工程方法论的基础。霍尔三维结构以时间维、()维、知识维组成的立体结构概括性地表示出系统工程的各阶段、各步骤以及所涉及的知识范围。其中时间维是系统的工作进程，对于一个具体的工程项目，可以分为7个阶段，在()阶段会做出研制方案及生产计划。

- | | | | |
|-------|-------|-------|-------|
| A. 知识 | B. 需求 | C. 文档 | D. 信息 |
| A. 战略 | B. 规划 | C. 实现 | D. 处理 |
| A. 空间 | B. 结构 | C. 组织 | D. 逻辑 |
| A. 规划 | B. 拟定 | C. 研制 | D. 生产 |

【参考答案】DBDC

1. 2信息系统生命周期

知识点1：信息系统生命周期

系统工程生命周期阶段及方法：

探索性研究→概念阶段→开发阶段→生产阶段→使用阶段→保障阶段→退役阶段

系统工程生命周期方法：

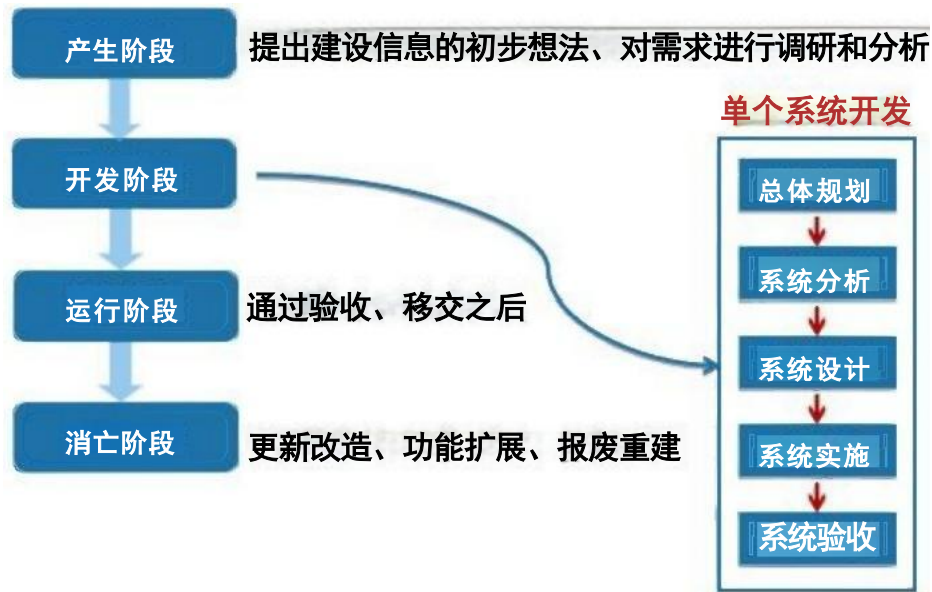
计划驱动方法：需求→设计→构建→测试→部署

渐进迭代式开发：提供连续交付以达到期望的系统

精益开发：起源于丰田，是一个动态的、知识驱动的，以客户为中心的过程

敏捷开发：更好的灵活性

信息系统生命周期：



信息系统建设原则：

高层管理人员介入原则

用户参与开发原则

自顶向下规划原则

工程化原则

其他原则

考点1：信息系统生命周期

【模拟题练习】在信息系统生命周期中，系统分析阶段的主要任务是()。

- A. 确定系统目标
- B. 进行可行性研究
- C. 制定系统实施计划
- D. 完成系统验收

【参考答案】A

【. 点拨】系统分析阶段的主要任务是对现行系统进行详细调查，彻底掌握现行系统的情况，明确用户的真实需求，并在此基础上提出新系统的逻辑模型。总结就是先设定目标再进行其他任务。

1.3 信息系统的分类

信息系统的分类	关键点
业务处理系统【TPS】	早期最初级的信息系统【20世纪50-60年代】。 功能：数据输入、数据处理【批处理、OLTP】、数据库维护、文件报表产生。
管理信息系统【MIS】	高度集成化的人机信息系统。 金字塔结构：分多个层级。
决策支持系统【DSS】	由语言系统、知识系统和问题处理系统组成。 用于辅助决策、支持决策。
专家系统【ES】	知识+推理=专家系统。人工智能的一个重要分支。
办公自动化系统【OAS】	由计算机设备、办公设备、数据通信及网络设备、软件系统组成。
企业资源计划【ERP】	打通供应链，集成，整合。

知识点1:决策支持系统【DSS】

【决策支持系统（Decision Support System, DSS）】是一个由语言系统、知识系统和问题处理系统3个互相关联的部分组成的，基于计算机的系统。

DSS 应具有的特征：

1. 数据和模型是 DSS 的主要资源。
2. DSS 用来支援用户作决策而不是代替用户作决策。
3. DSS 主要用于解决半结构化及非结构化问题。
4. DSS 的作用在于提高决策的有效性而不是提高决策的效率。

考点1：决策支持系统的特征

【模拟题练习】决策支持系统（Decision Support System, DSS）是一个由语言系统、知识系统和问题处理系统3个互相关联的部分组成的，基于计算机的系统。以下关于 DSS 的描述中错误的是（ ）。

- A. 主要资源是数据和模型
- B. 用来支援用户作决策而不是代替用户作决策

- C. 主要用于解决半结构化和非结构化的问题
- D. 可以提高决策的效率

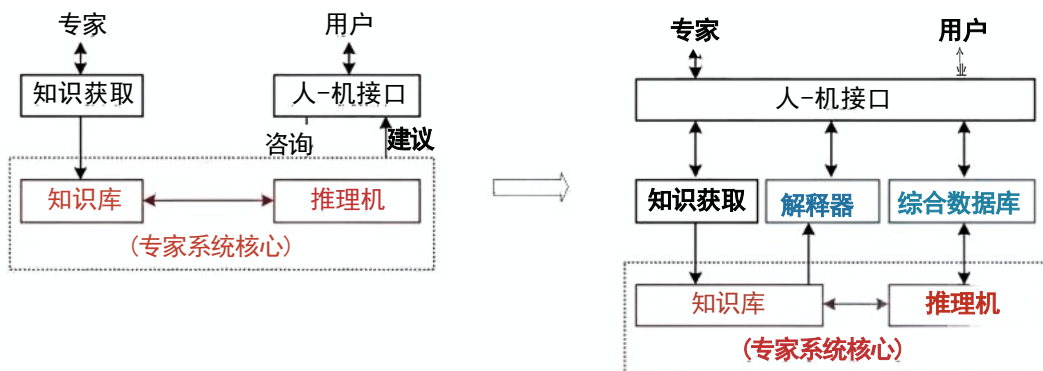
【参考答案】 D

知识点2: 专家系统【ES】

【专家系统 (Expert System,ES)】是一个智能计算机程序系统，其内部含有某个领域具有专家水平的大量知识与经验，能够利用人类专家的知识和解决问题的方法来处理该领域的问题。

系统	专家系统	一般计算机系统
功能	解决问题、解释结果、进行判断与决策	解决问题
处理能力	处理数字与符号	处理数字
处理问题种类	多属准结构性或非结构性，可处理不确定的知识，适用于特定的领域	多属结构性，处理确定的知识

组成：



知识库： 存储求解实际问题的领域知识。

综合数据库： 存储问题的状态描述、中间结果、求解过程的记录等信息。

推理机： 实质是【规则解释器】。

知识获取： 两方面功能——知识的编辑求精及知识自学习。

解释程序： 面向用户服务的。

考点1: 专家系统的组成

【模拟题练习】 以下关于专家系统【ES】的说法，不正确的是()。

- A. 专家系统不但能解决问题解释结果，还能进行判断与决策

-
- B. 专家系统中的知识库用于存储问题的状态描述、中间结果、求解过程等信息
 - C. 专家系统可用于处理不确定的知识，主要使用于特定的领域
 - D. 专家系统内部含有某个领域具有专家水平的大量知识与经验

【参考答案】 B

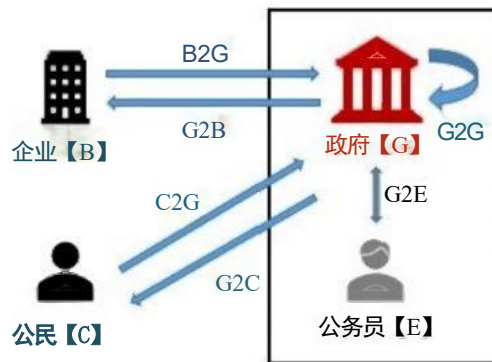
【. 点拨】 专家系统中的综合数据库用于存储问题的状态描述、中间结果、求解过程等信息。

1.4 政府信息化与电子政务

知识点1: 电子政务

电子政务主要有3类角色：政府(Government)、 企(事)业单位(Business) 及公民 (Citizen)。

如果有第4类就是公务员 (Employee)。



类型	应用
G2G	基础信息的采集、处理和利用，如：人口信息、地理信息。及政府间：计划管理、财务管理、通信系统、各级政府决策支持。
G2B	政府给企业单位颁发【各种营业执照、许可证、合格证、质量认证】。
B2G	企业向政府缴税； 企业向政府供应各种商品和服务【含竞/投标】； 企业向政府提建议，申诉。
G2C	社区公安和水、火、天灾等与公共安全有关的信息； 户口、各种证件和牌照的管理。
C2G	个人应向政府缴纳的各种税款和费用； 个人向政府反馈民意【 征求群众意见 】； 报警服务(盗贼、医疗、急救、火警等)。
G2E	政府内部管理系统。

考点1：电子政务行为主体

【2013年下】与电子政务相关的行为主体主要有三个，即（），政府的业务活动也主要围绕着这三个行为主体展开。

- A. 政府、数据及电子政务系统
- B. 政府、企（事）业单位及中介
- C. 政府、服务机构及企事业单位
- D. 政府、企（事）业单位及公民

【参考答案】 D

【点拨】与电子政务相关的行为主体包括：政府、企（事）业单位及公民。常见的电子政务形式包括：G2G、G2B、G2C，其中的G是政府、B是企（事）业单位、C是公民。

【2015年下】电子政务的主要应用模式中不包括（）。

- A. 政府对政府 (Government To Government)
- B. 政府对客户 (Government To Customer)
- C. 政府对公务员 (Government To Employee)
- D. 政府对企业 (Government To Business)

【参考答案】 B

【点拨】B 选项中：政府对客户 (Government To Customer) 不正确，应是：政府对公民 (Government To Citizen)。

考点2: 电子政务应用

【2016年下】电子政务是对现有的政府形态的一种改造，利用信息技术和其他相关技术，将其管理和服务职能进行集成，在网络上实现政府组织结构和 workflows 优化重组。与电子政务相关的行为主体有三个，即政府、() 及居民。国家和地方人口信息的采集、处理和利用，属于() 的电子政务活动。

- | | | | |
|----------|------------|----------|----------|
| A. 部门 | B. 企(事)业单位 | C. 管理机构 | D. 行政机关 |
| A. 政府对政府 | B. 政府对居民 | C. 居民对居民 | D. 居民对政府 |

【参考答案】BA

【点拨】电子政务的行为主体包括：政府、企(事)业单位及居民。

国家和地方人口信息的采集、处理和利用，属于政府对政府的电子政务活动。

【模拟题练习】电子政务主要有3类角色：政府 (Government)、企(事)业单位 (Business) 及公民 (Citizen)，政府与政府、政府与企(事)业单位以及政府与公民之间的互动构成了5种不同的、却又相互关联的领域。其中人口信息、地理信息的采集属于() 领域；颁发许可证、合格证属于() 领域；证件和牌照的管理属于() 。

- | | |
|---------------------|----------------|
| A. 政府对企(事)业单位 (G2B) | B. 政府与政府 (G2G) |
| C. 企业对政府 (B2G) | D. 政府对公民 (G2C) |
| A. 政府对企(事)业单位 (G2B) | B. 政府与政府 (G2G) |
| C. 企业对政府 (B2G) | D. 政府对公民 (G2C) |
| A. 政府对企(事)业单位 (G2B) | B. 政府与政府 (G2G) |
| C. 企业对政府 (B2G) | D. 政府对公民 (G2C) |

【参考答案】BAD

1.5 企业信息化与电子商务

知识点1: 信息化概念

企业信息化涉及三类创新：

【技术创新】在生产工艺设计、产品设计中使用计算机辅助设计系统，并通过互联网及时了解和掌握创新的技术信息，加快从技术向生产的转化。还有，生产技术与信息技术相结合，能够大幅度地提高技术水平和产品的竞争力。

【管理创新】按照市场发展的要求，要对企业现有的管理流程重新整合，从作为**管理核心的财务、资金管理，转向技术、物资、人力资源的管理，并延伸到企业技术创新、工艺设计、产品设计、生产制造过程的管理**，进而扩展到客户关系管理、供应链的管理乃至发展到电子商务。

【制度创新】那些不适应企业信息化的管理体制、管理机制和管理制度必须得到创新。

信息化需求的三个层次：

组织对信息化的需求是【**组织信息化的原动力**】。

一是战略需求。组织信息化的目标是提升组织的竞争能力、为组织的可持续发展提供一个支持环境。从某种意义上来说，信息化对组织不仅仅是服务的手段和实现现有战略的辅助工具；信息化可以把组织战略提升到一个新的水平，为组织带来新的发展契机。特别是对于企业，信息化战略是企业竞争的基础。

二是运作需求。组织信息化的运作需求是组织信息化需求非常重要且关键的一环，它包含三方面的内容：一是实现信息化战略目标的需要；二是运作策略的需要。三是人才培养的需要。

三是技术需求。由于系统开发时间过长等问题在信息技术层面对系统的完善、升级、集成和整合提出了需求。也有的组织，原来基本上没有大型的信息系统项目，有的也只是一些单机应用，这样的组织的信息化需求，一般是从头开发新的系统。

组织的三个层次的需求并不是相互孤立的，而是有着内在的联系。信息化需求的获取是一个自上而下的过程，需要对这些需求进行综合分析，才能把握组织对信息化建设的方向。

考点1：企业信息化的三类创新

【2013年下】企业信息化涉及对企业管理理念的创新，按照市场发展的要求，对企业现有的管理流程重新整合，管理核心从对（）的管理，转向对（）的管理，并延伸到对企业技术创新、工艺设计、产品设计、生产制造过程的管理，进而还要扩展到对（）的管理乃至发展到电子商务。

- | | |
|---------------|---------------|
| A. 人力资源和物资 | B. 信息技术和知识 |
| C. 财务和物料 | D. 业务流程和数据 |
| A. 业务流程和数据 | B. 企业信息系统和技术 |
| C. 业务流程、数据和接口 | D. 技术、物资和人力资源 |
| A. 客户关系和供应链 | B. 信息技术和知识 |
| C. 生产技术和信息技术 | D. 信息采集、存储和共享 |

【参考答案】CDA

考点2: 信息化需求的三个层次

【2021年下】政府、企业等对信息化的需求是能组织信息化的原动力，它决定了组织信息化的价值取向和成果效益水平，而需求本身又是极为复杂的，它是一个系统性的、多层次的目标体系，组织信息化需求通常包含三个层次，即()，三个层次的需求并不是相互孤立的，而是有着内在的联系。

- A. 战略需求，运作需求，功能需求
- B. 战略需求，运作需求，技术需求
- C. 市场需求，技术需求，用户需求
- D. 市场需求，技术需求，领域需求

【参考答案】B

【2017年下】组织信息化需求通常包含三个层次，其中()需求的目标是提升组织的竞争能力，为组织的可持续发展提供支持环境。()需求包含实现信息化战略目标的需求、运营策略的需求和人才培养的需求三个方面。技术需求主要强调在信息层技术层面上对系统的完善、升级、集成和整合提出的需求。

- | | | | |
|-------|-------|-------|--------|
| A. 战略 | B. 发展 | C. 人事 | D. 财 务 |
| A. 规划 | B. 运作 | C. 营销 | D. 管理 |

【参考答案】AB

知识点2: 企业信息化方法

业务流程重组方法：“彻底的、根本性的”重新设计流程。

核心业务应用方法：围绕核心业务推动信息化。

信息系统建设方法：建设信息系统作为企业信息化的重点和关键。

主题数据库方法：建立面向企业的核心业务的数据库，消除“信息孤岛”。

资源管理方法：切入点是为企业资源管理提供强大的能力。如：ERP、SCM。

人力资本投资方法：人力资本理论【注意不是人力资源管理】把一部分企业的优秀员工看作是一种资本，能够取得投资收益。

考点1: 企业信息化方法及其特点

【2012年下】企业信息化程度是国家信息化建设的基础和关键，企业信息化方法不包括()。

-
- A. 业务流程重组
 - B. 组织机构变革
 - C. 供应链管理
 - D. 人力资本投资

【参考答案】B

【点拨】企业信息化程度是国家信息化建设的基础和关键，企业信息化就是企业利用现代信息技术，通过信息资源的深入开发和广泛利用，实现企业生产过程的自动化、管理方式的网络化、决策支持的智能化和商务运营的电子化，不断提高生产、经营、管理、决策的效率和水平，进而提高企业经济效益和企业竞争力的过程。企业信息化方法主要包括业务流程重组、核心业务应用、信息系统建设、主题数据库、资源管理和人力资本投资方法。而供应链管理的简称为（SCM），属于资源管理方法，是企业信息化方法中的一种。所以答案选B 选项。

【2011年下】以下关于企业信息化方法的叙述中，正确的是（）。

- A. 业务流程重构是对企业的组织结构和工作方法进行重新设计，SCM（供应链管理）是一种重要的实现手段
- B. 在业务数量浩繁且流程错综复杂的大型企业里，主题数据库方法往往形成许多“信息孤岛”，造成大量的无效或低效投资
- C. 人力资源管理把企业的部分优秀员工看作是一种资本，能够取得投资收益
- D. 围绕核心业务应用计算机和网络技术是企业信息化建设的有效途径

【参考答案】D

【点拨】选项A 描述错误，因为 SCM 不是业务流程重构的实现手段。选项B 描述错误，因为事务型数据库容易形成信息孤岛，而主题数据库不容易形成“信息孤岛”。C 选项描述错误，因为人力资源是把所有员工看作是一种资本，而非部分员工。

知识点3: 信息系统战略规划（ISSP）

信息系统战略规划（Information System Strategic Planning, ISSP）是从企业战略出发，构建企业基本的信息架构，对企业内、外信息资源进行统一规划、管理与应用，利用信息控制企业行为，辅助企业进行决策，帮助企业实现战略目标。

ISSP 方法经历了三个主要阶段，各个阶段所使用的方法也不一样。

第一个阶段（以数据处理为核心）

主要以数据处理为核心，围绕职能部门需求的信息系统规划，主要的方法包括：

(1) 企业系统规划法 (BSP)--CU 矩阵：自上而下识别系统目标，自下而上设计信息系统，对组织机构的变动具有适应性。

(2) 关键成功因素法 (CSF)：找出实现目标的关键信息集合，从而确定开发优先次序。

(3) 战略集合转化法 (SST)：把战略目标看成“信息集合”，把战略目标转变成信息系统的战略目标。

(4) 其它方法包括：投资回收法、征费法、零线预算法、阶石法。

第二个阶段(以企业内部管理信息系统为核心)

主要以企业内部管理信息系统为核心，围绕企业整体需求进行的信息系统规划，主要的方法包括战略数据规划法 (SDP)、信息工程法 (IE) 和战略栅格法 (SG)；

第三个阶段(以集成为核心)

在综合考虑企业内外环境的情况下，以集成为核心，围绕企业战略需求进行的信息系统规划，主要的方法包括价值链分析法 (VCA) 和战略一致性模型 (SAM)。

考点1: 信息系统战略规划方法

【2017年下】用于管理信息系统规划的方法有很多，其中()将整个过程看成是一个“信息集合”，并将组织的战略目标转变为管理信息系统的战略目标。()通过自上而下地识别企业目标、企业过程和数据，然后对数据进行分析，自下而上地设计信息系统。

- | | |
|---------------|-------------|
| A. 关键成功因素法 | B. 战略目标集转化法 |
| C. 征费法 | D. 零线预算法 |
| A. 企业信息分析与集成法 | B. 投资回收法 |
| C. 企业系统规划法 | D. 阶石法 |

【参考答案】BC

【模拟题练习】信息系统规划方法中，关键成功因素法通过对关键成功因素的识别，找出实现目标所需要的关键信息集合，从而确定系统开发的()。企业系统规划法方法认为企业目标到系统目标的转化需要通过进行()分析得到。

- | | | | |
|-----------|------------|---------|---------|
| A. 系统边界 | B. 功能指标 | C. 优先次序 | D. 性能指标 |
| A. 关键成功因素 | B. 过程/数据分析 | C. 信息集合 | D. 主要矛盾 |

【参考答案】CB

【. 点拨】企业系统规划法虽然也首先强调目标，但它没有明显的目标导引过程。它通过识别企业“过程”引出了系统目标，企业目标到系统目标的转化是通过企业过程/数据类等矩阵（CU 矩阵）的分析得到的。

知识点4: 企业资源计划（ERP）

（1）发展过程



（2）ERP 结构

ERP 是将企业所有资源（企业三大流：物流、资金链、信息流）进行集成整合，全面一体化管理的管理信息系统。

包括三方面：生产控制（计划、制造）、物流管理（分销、采购、库存管理）和财务管理（会计核算、财务管理）。这三个系统本身就是一个集成体，它们相互之间有相应的接口，能够很好地整合在一起。

考点1：ERP 结构

【2012年下】ERP 中的企业资源包括（）。

- A. 物流、资金流和信息流
- B. 物流、工作流和信息流
- C. 物流、资金流和工作流
- D. 资金流、工作流和信息流

【参考答案】A

【. 点拨】ERP 中的企业资源包括企业的“三流”资源，即物流资源、资金流资源和信息流资源。

ERP 实际上就是对这“三流”资源进行全面集成管理的管理信息系统。

【2016年下】 ERP(Enterprise Resource Planning) 是建立在信息技术的基础上，利用现代企业的先进管理思想，对企业的物流、资金流和()流进行全面集成管理的管理信息系统，为企业提供决策、计划、控制与经营业绩评估的全方位和系统化的管理平台。在ERP系统中，()管理模块主要是对企业物料的进、出、存进行管理。

- | | | | |
|-------|---------|-------|-------|
| A. 产品 | B. 人力资源 | C. 信息 | D. 加工 |
| A. 库存 | B. 物料 | C. 采购 | D. 销售 |

【参考答案】 CA

【点拨】 采购与库存管理是ERP的基本模块，其中采购管理模块是对采购工作——从采购订单产生至货物收到的全过程进行组织、实施与控制，库存管理(Inventory Management, IM)模块则是对企业物料的进、出、存进行管理。

知识点5: 客户管理CRM

CRM(Customer Relationship Management) 理念：将客户看作资产；客户关怀是中心，目的是提高收入；核心思想是以客户为中心。

CRM 的主要模块：

- 销售自动化；
- 营销自动化；
- 客户服务与支持；
- 商业智能。

CRM 的价值：

- 提高工作效率，节省开支；
- 提高客户满意度；
- 提高客户的忠诚度。

考点1：CRM 的主要模块

【2012年下】 CRM 是一套先进的管理思想及技术手段，它通过将()进行有效的整合，最终为企业涉及的各个领域提供了集成环境。CRM系统的四个主要模块包括()。

- A. 员工资源、客户资源与管理技术
- B. 销售资源、信息资源与商业智能

-
- C. 销售管理、市场管理与服务管理
 - D. 人力资源、业务流程与专业技术
 - A. 电子商务支持、呼叫中心、移动设备支持、数据分析
 - B. 信息分析、网络应用支持、客户信息仓库、工作流集成
 - C. 销售自动化、营销自动化、客户服务与支持、商业智能
 - D. 销售管理、市场管理、服务管理、现场服务管理

【参考答案】 DC

【点拨】 CRM 是一套先进的管理思想及技术手段，它通过将人力资源、业务流程与专业技术进行有效的整合，最终为企业涉及的客户或者消费者的各个领域提供了完美的集成，使得企业可以更低成本、更高效地满足客户的需求，并与客户建立起基于学习性关系基础上的一对一营销模式，从而让企业可以最大程度提高客户满意度和忠诚度。CRM 系统的主要模块包括销售自动化、营销自动化、客户服务与支持、商业智能。

知识点6: 供应链管理SCM

SCM 理念：强强联合，整合与优化“三流”，打通企业间“信息孤岛”，严格的数据交换标准。将制造商、供应商、分销商、零售商，在计划（策略性）、采购、制造、配送、退货等各方面联系起来。

信息化的三流

1. 信息流【核心】

需求信息流（需方到供方）：如客户订单、生产计划、采购合同等。

供应信息流（供方到需方）：如入库单、完工报告单、库存记录、可供销售量、提货发运单等。

2. 资金流【辅助】

3. 物流【辅助】

考点1: 信息流的分类

【2015年下】供应链中的信息流覆盖了从供应商、制造商到分销商，再到零售商等供应链中的所有环节，其信息流分为需求信息流和供应信息流，（）属于需求信息流，（）属于供应信息流。

- | | | | |
|---------|---------|----------|----------|
| A. 库存记录 | B. 生产计划 | C. 商品入库单 | D. 提货发运单 |
| A. 客户订单 | B. 采购合同 | C. 完工报告单 | D. 销售报告 |

【参考答案】 BC

【.点拨】供应链中的信息流覆盖了从供应商、制造商到分销商再到零售商等供应链中的所有环节。其信息流分为需求信息流和供应信息流，这是两个不同流向的信息流。当需求信息(如客户订单、生产计划、采购合同等)从需方向供方流动时，便引发物流。同时供应信息(如入库单、完工报告单、库存记录、可供销售量、提货发运单等)又同物料一起沿着供应链从供方向需方流动。

【模拟题练习】在供应链的管理中，涉及到信息化的三流：信息流、资金流、物流，其中()是核心。具体细化时，信息流又可分为需求信息流和供应信息流，()属于需求信息流，()属于供应信息流。

- | | | | |
|---------|----------|---------|---------|
| A.商流 | B. 信息流 | C.资金流 | D.物流 |
| A. 客户订单 | B. 可供销售量 | C.商品入库单 | D.提货发运单 |
| A. 客户订单 | B. 采购合同 | C. 库存记录 | D.生产计划 |

【参考答案】BAC

考点2:企业供应链构成

【模拟题练习】下面不属于企业供应链构成节点的是()。

- A. 制造商
- B.供应
- C. 配送中心
- D.视频会议

【参考答案】D

【.点拨】SCM 是企业的有效性管理，表现了企业在战略和战术上对业务流程的优化。整合并优化了供应商、制造商、零售商的业务效率，使商品以正确的数量、正确的品质、在正确的地点、以正确的时间、最佳的成本进行生产和销售。SCM 包括计划、采购、制造、配送、退货五大基本内容。

知识点7: 商业智能BI

(1) 过程

需求分析→数据仓库建模→数据抽取→建立 BI分析报表→用户培训和数据模拟测试→系统改进和完善

(2) 相关技术：数据仓库+数据挖掘+OLAP

(3) 用途：决策分析【分析历史数据预判未来】

(4) 数据仓库

数据库	数据仓库【特点】
面向应用：应用组织数据	面向主题：主题组织数据
零散的：一个应用对应一个数据库	集 成 的：整个企业对应一个数据仓库
CRUD:增删改查是常态	相对稳定的(非易失的): 查询为主、基本无修改与删除
解决当下应用问题	反映历史变化(时变的): 各个阶段信息都有，并可预测未来趋势

(5) 数据挖掘：分类：

关联分析：挖掘出隐藏在数据间的相互关系。

序列模式分析：侧重点是分析数据间的前后关系(因果关系)。

分类分析：为每一个记录赋予一个标记再按标记分类。

聚类分析：分类分析法的逆过程。

(6) 数据湖

概念：数据湖是一个存储企业的各种各样原始数据的大型仓库，其中的数据可供存取、处理、分析及传输。

特点：数据湖从企业的多个数据源获取原始数据，并且针对不同的目的，同一份原始数据还可能有多种满足特定内部模型格式的数据副本。因此，数据湖中被处理的数据可能是任意类型的信息，从结构化数据到完全非结构化数据。

区别：数据仓库仅支持分析处理，数据湖既支持分析处理，也支持事务处理。

比较：

维度	数据仓库	数据湖
数据	清洗过的数据 结构化的数据	原始数据 结构化，半结构化数据
模式	数据存储之前定义数据模式 数据集成之前完成大量工作 数据的价值提前明确	数据存储之后定义数据模式 提供敏捷，简单的数据集成 数据的价值尚未明确
存取方法	标准SQL接口	应用程序，类SQL的程序
优势	多数据源集成 干净，安全的数据 转换一次，多次使用	无限扩展性 并行执行 支持编程框架 数据经济

考点1：商业智能相关技术

【2014年下】商业智能是企业对商业数据的搜集、管理和分析的系统过程，主要技术包括（）。

- A. 数据仓库、联机分析和数据挖掘
- B. 数据采集、数据清洗和数据挖掘
- C. 联机分析、多维度分析和跨维度分析
- D. 数据仓库、数据挖掘和业务优化重组

【参考答案】 A

考点2：数据挖掘

【2013年下】数据挖掘是从数据库的大量数据中揭示出隐含的、先前未知的并有潜在价值的信息的非平凡过程，主要任务有（）。

- A. 聚类分析、联机分析、信息检索等
- B. 信息检索、聚类分析、分类分析等
- C. 聚类分析、分类分析、关联规则挖掘等
- D. 分类分析、联机分析、关联规则挖掘等

【参考答案】 C

【. 点拨】数据挖掘的任务有关联分析、聚类分析、分类分析、异常分析、特异群组分析和演变分析等等。

知识点8:企业应用集成

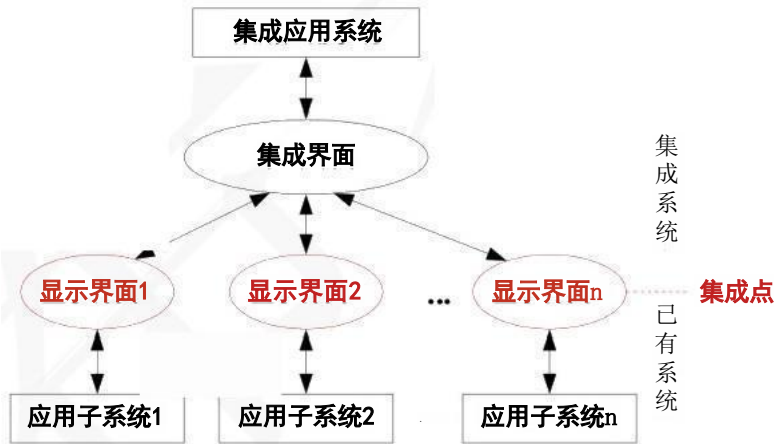
企业集成分类：

按集成点分：

	集成点	效果	解题关键点
界面集成	界面	统一入口，产生“整体”感觉	“整体”感觉 最小代价实现一体化操作
数据集成	数据	不同来源的数据逻辑或物理上“集中”	其他集成方法的基础
控制集成	应用逻辑	调用其他系统已有方法，达到集成效果	
业务流程集成 (过程集成)	应用逻辑	跨企业，或优化流程而非直接调用	企业之间的信息共享能力
门户集成		将内部系统对接到互联网上	发布到互联网上

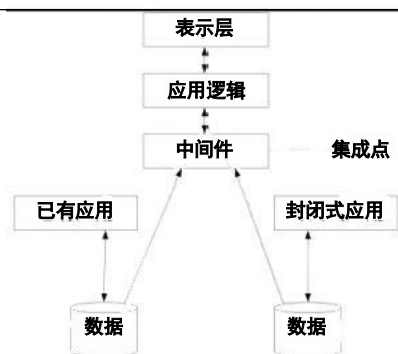
(1) 表示集成（界面集成）

把各应用系统的界面集成起来，统一入口，产生“整体”感觉。



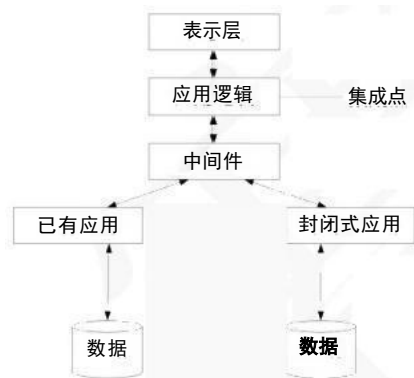
(2) 数据集成

数据集成是应用集成和业务流程集成的基础。把不同来源、格式、特点性质的数据在逻辑上或物理上有机地集中，从而为企业提供全面的数据共享。 ETL、 数据仓库、 联邦数据库都可视为数据集成。



(3) 控制集成(功能集成、应用集成、 API 集成)

业务逻辑层次集成，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。



(4) 业务流程集成(过程集成， B2B)

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。

按传输方式分：

	特点
消息集成	数据量小，交互频繁，立即地，异步
共享数据库	交互频繁，立即地，同步
文件传输	数据量大，交互频度小，即时性要求低(月末，年末)

EAI 提供4个层次的服务：

EAI提供4个层次的服务	功能
流程控制服务	解决人工参与的长期的工作流程控制问题
应用连接服务	应用连接适配器将应用接口连接至EAI平台
信息传递与转化服务	负责传递消息和转化消息
通讯服务	通过通讯中间件进行消息的路由

企业集成技术的架构层次：

企业集成技术的架构层次	说明
会聚集成	集成化运行
应用集成	语用互操作 模式：集成适配器、集成信使、集成面板和集成代理4种
数据集成	语义互通 模式：数据联邦、数据复制、基于接口的数据集成
网络集成	语法互联

企业门户：

企业信息门户 (Enterprise Information Portal,EIP): 使员工/合作伙伴/客户/供应商都能够访问企业内部网络和因特网存储的各种自己所需的信息。 **【统一访问入口】**

企业知识门户 (Enterprise Knowledge Portal,EKP): 企业网站的基础上增加知识性内容。 **【企业知识库】**

企业应用门户 (Enterprise Application Portal,EAP): 以商业流程和企业应用为核心，把商业流程中功能不同的应用模块通过门户技术集成在一起。 **【企业信息系统的网上集成界面】**

垂直门户：为某一特定的行业服务的，传送的信息只属于人们感兴趣的领域。

考点1: 企业集成的特点

【2013年下】企业信息集成按照组织范围分为企业内部的信息集成和外部的信息集成。在企业内部的信息集成中，()实现了不同系统之间的互操作，使得不同系统之间能够实现数据和方法的共享；()实现了不同应用系统之间的连接、协调运作和信息共享。

- A. 技术平台集成 B. 数据集成 C. 应用系统集成 D. 业务过程集成
A. 技术平台集成 B. 数据集成 C. 应用系统集成 D. 业务过程集成

【参考答案】CD

【2019年下】系统应用集成构建统一标准的基础平台，在各个应用系统的接口之间共享数据和功能，基本原则是保证应用程序的()。系统应用集成提供了4个不同层次的服务，最上层服务是()服务。

- A. 独立性 B. 相关性 C. 互操作性 D. 排他性
A. 通信 B. 信息传递与转化 C. 应用连接 D. 流程控制

【参考答案】AD

【点拨】EAI 构建统一标准的基础平台，将进程、软件、标准和硬件联合起来，连接具有不同功能和目的而又独自运行的企业内部的应用系统，以达到信息和流程的共享，使企业相关应用整合在一起。

EAI 就是在各个应用系统的接口之间共享数据和功能。EAI 的基本原则就是集成多个系统并保证系统互不干扰，也就是独立性。因此第一空选择A 选项独立性。排他性一般理解为不能共存，因此在这里并不合适。

EAI 的终极目标就是将多个企业和企业内部的多个应用集成到一个虚拟的、统一的应用系统中。因此实施 EAI 必须遵循如下原则：应用程序的独立性；面向商业流程；独立于技术；平台无关。

EAI 提供4个层次的服务，从下至上依次为通信服务、信息传递与转化服务、应用连接服务、流程控制服务，最上层是流程控制服务，第二空选择D 选项。

【2024年下】在企业集成技术架构层次中，自下而上依次为（）。其中，数据集成主要有三种模式，分别为（）。

- | | |
|-----------------------|-----------------------|
| A.网络集成、数据集成、应用集成和会聚集成 | B.网络集成、会聚集成、数据集成和应用集成 |
| C.网络集成、数据集成、会聚集成和服务集成 | D.网络集成、会聚集成、数据集成和服务集成 |
| A.数据复制、数据转换和数据融合 | B.数据复制、数据融合和数据抽取 |
| C.数据联邦、数据复制和基于接口的数据集成 | D.数据抽取、数据复制和数据测试 |

【参考答案】AC

【点拨】企业集成技术的架构层次：网络集成（语法互联）、数据集成（语义互通）、应用集成（语用互操作）、会聚集成（集成化运行）。

其中数据集成主要有以下三种模式：数据联邦、数据复制和基于接口的数据集成。

（1）数据联邦

数据联邦是指不同的应用共同访问一个全局虚拟数据库，通过全局虚拟数据库管理系统为不同的应用提供全局信息服务，实现不同的应用和数据源之间的信息共享和数据交换，其具体实现由客户端应用、全局信息服务和若干个局部数据源三部分组成。

（2）数据复制模式

在数据复制模式中，通过底层应用数据源之间的一致性复制来实现（访问不同数据库的）不同应用之间的信息共享和互操作，其实现的关键是必须能够提供在两个或多个数据库系统之间实现数据转换和传输的基础结构（以屏蔽不同数据库间数据模型的差异）。

（3）基于接口的数据集成模式

在基于接口的数据集成模式中，不同的应用系统之间利用适配器(或接口代理)提供的应用程序接口来实现相互调用。应用适配器或接口代理通过其开放或私有接口将业务信息从其所封装的具体应用系统中提取出来，进而实现不同的应用系统之间业务数据的共享与互交换。接口调用的方式可以采用同步调用方法，也可以采用基于消息中间件的异步方法来实现。

考点2:企业门户

【2010年下】某大型公司欲开发一个门户系统，该系统以商业流程和企业应用为核心，将商业流程中不同的功能模块通过门户集成在一起，以提高公司的集中贸易能力、协同能力和信息管理能力。根据这种需求，采用企业()门户解决方案最为合适。

- A. 信息
- B. 知识
- C. 应用
- D. 垂直

【参考答案】 C

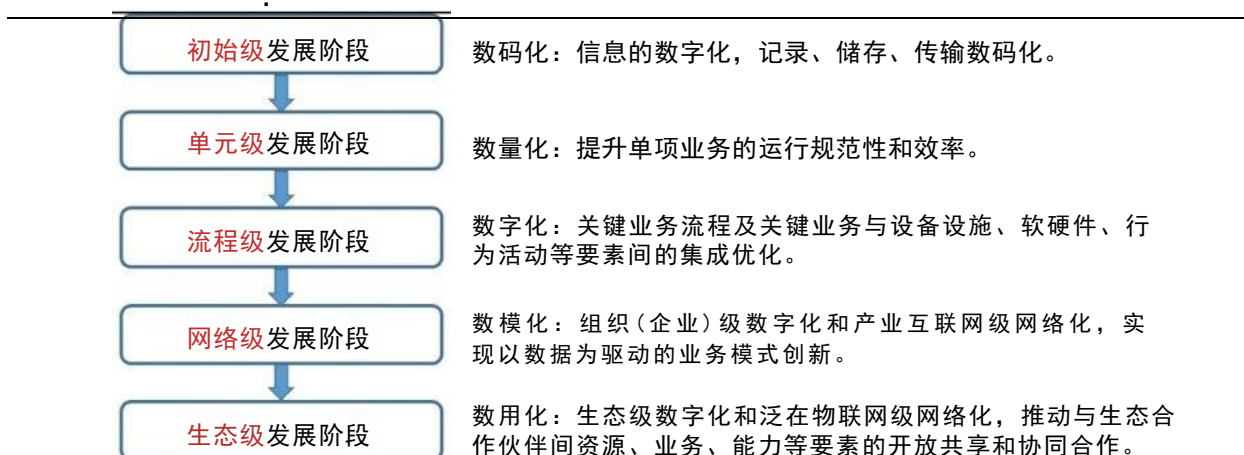
【点拨】企业门户是一个信息技术平台，这个平台可以提供个性化的信息服务，为企业提供单一访问企业各种信息资源和应用程序的入口。现有的企业门户大致可以分为企业信息门户、企业知识门户和企业应用门户三种。其中企业信息门户重点强调为访问结构数据和无结构数据提供统一入口，实现收集、访问、管理和无缝集成。企业知识门户提供了一个创造、搜集和传播企业知识的平台，通过企业知识门户，员工可以与工作团队中的其他成员取得联系，寻找能够提供帮助的专家。企业应用门户是一个用来提高企业的集中贸易能力、协同能力和信息管理能力的平台。它以商业流程和企业应用为核心，将商业流程中功能不同的应用模块通过门户集成在一起，提高公司的集中贸易能力、协同能力和信息管理能力。

1.6数字化转型智能制造体系

知识点1:企业数字化转型与智能制造体系

【数字化】是新一代信息技术真正的实现推动整个【商业模式的变革】，推动产业链的重构，推动改进企业与消费者之间的关系，以及企业与合作伙伴之间的关系。

企业数字化转型的五个发展阶段分别为：初始级发展阶段、单元级发展阶段、流程级发展阶段、网络级发展阶段、生态级发展阶段。



智能制造体系架构中，系统层级是指与企业生产活动相关的组织结构的层级划分，包括设备层、单元层、车间层、企业层和协同层。

(1) 设备层是指企业利用传感器、仪器仪表、机器、装置等，实现实际物理流程并感知和操控物理流程的层级；

(2) 单元层是指用于企业内处理信息、实现监测和控制物理流程的层级；

(3) 车间层是实现面向工厂或车间的生产管理的层级；

(4) 企业层是实现面向企业经营管理的层级；

(5) 协同层是企业实现其内部和外部信息互联和共享，实现跨企业间业务协同的层级。

考点1：企业数字化转型的发展阶段

【2021年下】企业数字化转型的五个发展阶段依次是（）。

A.初始级发展阶段、单元级发展阶段、流程级发展阶段、网络级发展阶段、生态级发展阶段

B.初始级发展阶段、单元级发展阶段、系统级发展阶段、网络级发展阶段、生态级发展阶段

C.初始级发展阶段、单元级发展阶段、流程级发展阶段、网络级发展阶段、优化级发展阶段

D.初始级发展阶段、流程级发展阶段、系统级发展阶段、网络级发展阶段、生态级发展阶段

【参考答案】 A

考点2：智能制造体系

【模拟题练习】智能制造系统架构从生命周期、系统层级和智能特征3个维度对智能制造所涉及的活动、装备、特征等内容进行描述，其中，系统层级是对与企业生产活动相关的组织结构的层级划分，包括（）。

A. 设备层、单元层、车间层、企业层和应用层

-
- B. 设备层、单元层、传输层、企业层和应用层
 - C. 设备层、单元层、传输层、企业层和协同层
 - D. 设备层、单元层、车间层、企业层和协同层

【参考答案】 D

第二章项目管理

2.1 立项管理—盈亏平衡管理

知识点1: 盈亏平衡管理

销售额=固定成本+可变成本+税费+利润【正常情况下】

销售额=固定成本+可变成本+税费【盈亏平衡时】

考点1: 盈亏平衡点

【2020年下】某厂生产的某种电视机，销售价为每台2500元，去年的总销售量为25000台，固定成本总额为250万元，可变成本总额为4000万元，税率为16%，则该产品年销售量的盈亏平衡点为（）台（只有在年销售量超过它时才能盈利）。

A.5000

B.10000

C.15000

D.20000

【参考答案】A

【点拨】本题考查的是盈亏平衡点计算问题。

盈亏平衡点也称为零利润点或保本点，是全部销售收入等于全部成本时的产量。当销售收入高于盈亏平衡点时，表示企业是盈利的状态；当销售收入低于盈亏平衡点时，表示企业是亏损的状态。

去年卖了25000台电视机，每台售价2500元，固定成本250万，可变成本4000万，税率16%。

总营收：25000*2500=6250万

固定成本：250万

可变成本：4000万，占营收比例：64%（4000万/6250万=64%）。

税不属于成本，但与可变成本性质相似，会随销量变化。

设盈亏平衡时的销售量为X台。则有：

$2500000 + X * 2500 * 64\% + X * 2500 * 16\% = X * 2500$

$500X = 2500000$

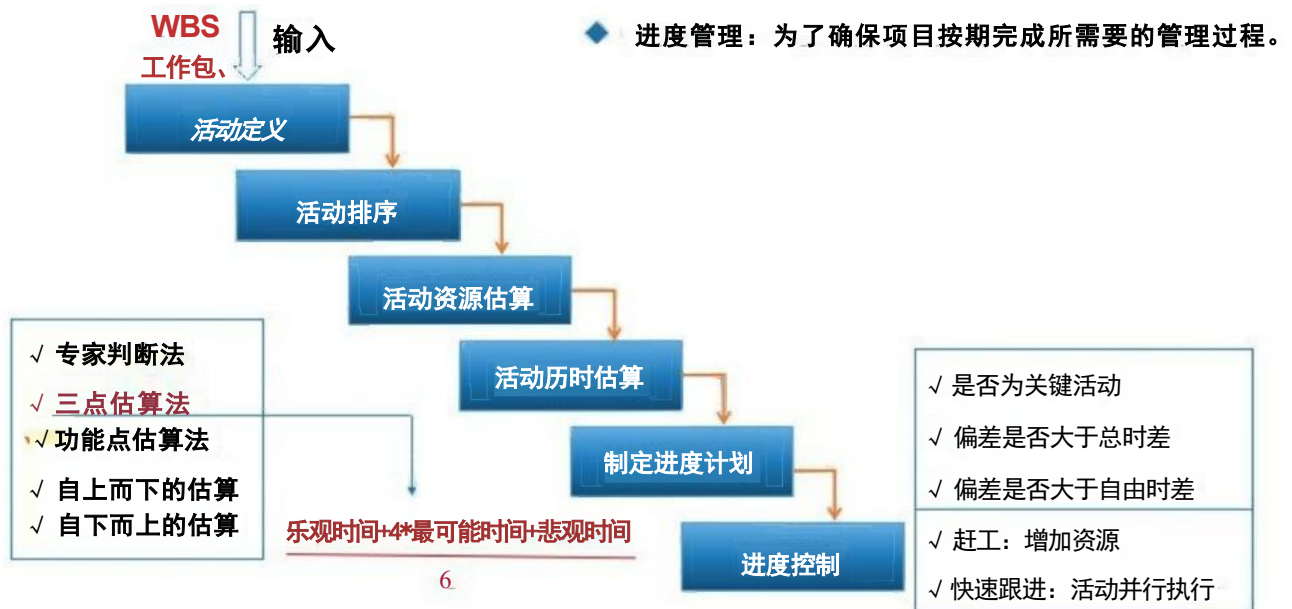
解得：X=5000

2.2 进度管理（时间管理）

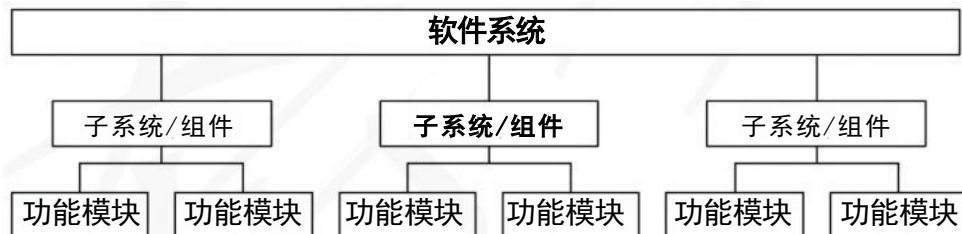
知识点1: 进度管理

进度管理：为了确保项目按期完成所需要的管理过程。

过程



工作分解结构【WBS】



【WBS 分解的基本要求】

- WBS 的工作包是可控和可管理的，不能过于复杂。
- 任务分解也不能过细，一般原则WBS 的树形结构不超过6层。
- 每个工作包要有一个交付成果。
- 每个任务必须有明确定义的完成标准。
- WBS 必须有利于责任分配。

历时估算方法

- 专家判断法
- 三点估算法
- 功能点估算法

d) 自上而下的估算

e) 自下而上的估算

进度控制

可能出现的问题：

- 是否为关键活动
- 偏差是否大于总时差
- 偏差是否大于自由时差

采取的手段：

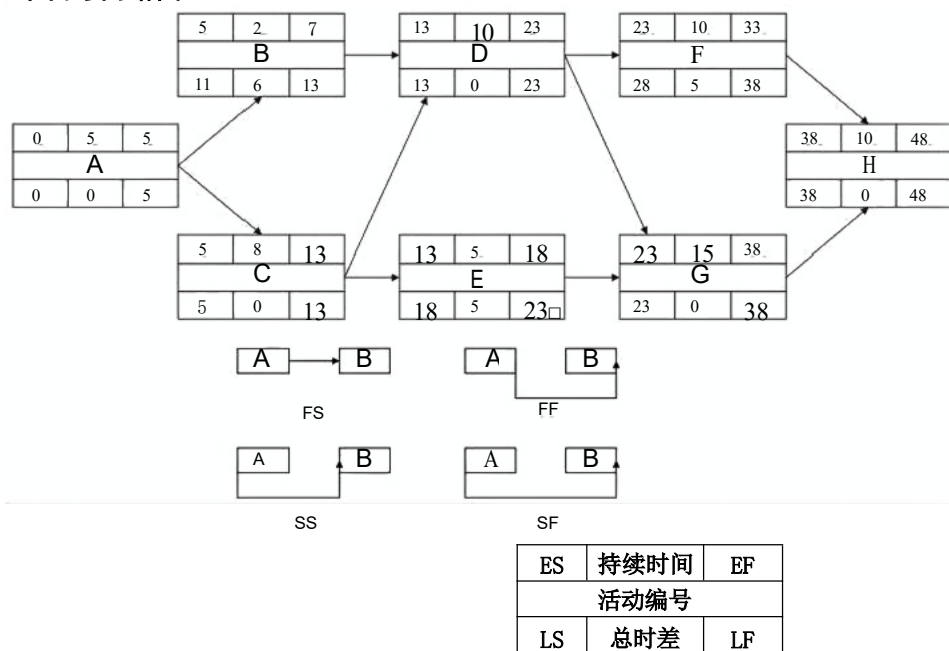
赶工：增加资源、加班

快速跟进：活动并行执行

进度网络图-关键路径法 (PERT)

关键路径法是在制订进度计划时使用的一种进度网络分析技术。关键路线法沿着项目进度网络路线进行正向与反向分析，从而计算出所有计划活动理论上的最早开始与完成日期、最迟开始与完成日期，不考虑任何资源限制。

如下图：单代号网络图



ES: 最早开始时间 EF: 最早完成时间

LS: 最迟开始时间 LF: 最迟完成时间

(1) 总时差(松弛时间):

在不延误总工期的前提下，该活动的机动时间。活动的总时差等于该活动最迟完成时间与最早完成时间之差，或该活动最迟开始时间与最早开始时间之差。

(2) 自由时差：

在不影响紧后活动的最早开始时间前提下，该活动的机动时间。

对于有紧后活动的活动，其自由时差等于所有紧后活动最早开始时间减去本活动最早完成时间所得之差的最小值。

对于没有紧后活动的活动，也就是以网络计划终点节点为完成节点的活动，其自由时差等于计划工期与本活动最早完成时间之差。

对于网络计划中以终点节点为完成节点的活动，其自由时差与总时差相等。此外，由于活动的自由时差是其总时差的构成部分，所以，当活动的总时差为零时，其自由时差必然为零，可不必进行专门计算。

Gantt图

工作编号	工作名称	工作时间 (M)	项目进度									
			1	2	3	4	5	6	7	8	9	10
1	需求分析	3										
2	设计建模	3										
3	编码	3.5										
4	测试	3										
5	实施部署	2										

检查日期

优点：甘特图直观、简单、容易制作，便于理解，能很清晰地标识出直到每一项任务的起始与结束时间，一般适用比较简单的小型项目，可用于WBS 的任何层次、进度控制、资源优化、编制资源和费用计划。

缺点：不能系统地表达一个项目所包含的各项工作之间的复杂关系，难以进行定量的计算和分析，以及计划的优化等。

考点1：进度管理

【2018年下】项目时间管理中的过程包括（）。

A. 活动定义、活动排序、活动的资源估算和工作进度分解

-
- B. 活动定义、活动排序、活动的资源估算、活动历时估算、制订计划和进度控制
 - C. 项目章程、项目范围管理计划、组织过程资产和批准的变更申请
 - D. 生产项目计划、项目可交付物说明、信息系统要求说明和项目度量标准

【参考答案】B

考点2: 工作分解结构WBS

【2010年下】项目时间管理包括使项目按时完成所必需的管理过程，活动定义是其中的一个重要过程。通常可以使用（）来进行活动定义。

- A. 鱼骨图
- B. 工作分解结构 (WBS)
- C. 层次分解结构
- D. 功能分解图

【参考答案】B

【点拨】项目时间管理包括使项目按时完成所必需的管理过程。项目时间管理中的过程包括：活动定义、活动排序、活动的资源估算、活动历时估算、制定进度计划以及进度控制。

为了得到工作分解结构 (Work Breakdown Structure, WBS) 中最底层的交付物，必须执行一系列的活动。对这些活动的识别以及归档的过程就是活动定义。

鱼骨图 (Ishikawa Diagram) 又称因果图，是一种发现问题“根本原因”的方法，通常用来进行因果分析。

考点3: 关键路径法

【2019年下】某工程项目包括六个作业A~F，各个作业的衔接关系以及所需时间见下表。作业D最多能拖延（）天，而不会影响该项目的总工期。

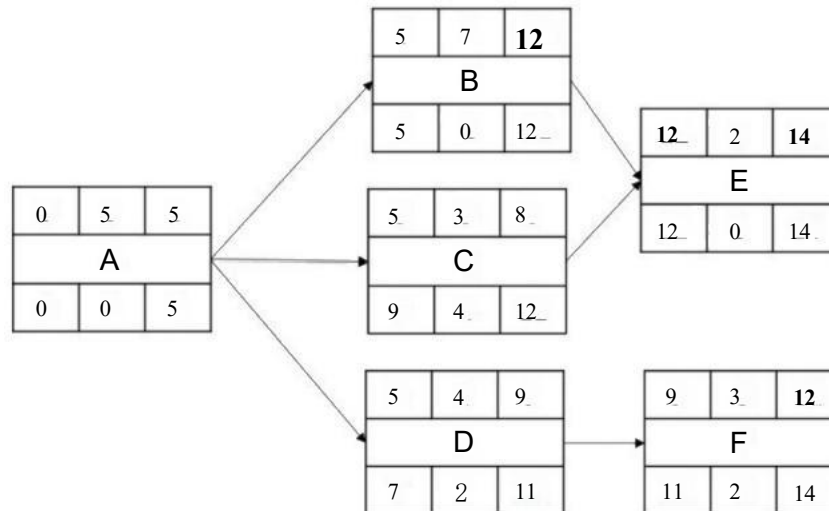
作业	A	B	C	D	E	F
紧前作业	-	A	A	A	B, C	D
时间/天	5	7	3	4	2	3

- A. 0
- B. 1
- C. 2
- D. 3

【参考答案】C

【. 点拨】 本题考查的是时间管理进度网络图的分析。

根据题干给出的依赖关系，可以画出单代号进度网络图，并分析其各活动的最早开始和完成时间、最晚开始和完成时间，以及总时差。结果如下：



本题考查的D 活动能够延迟的时间，就是其总时差，即可以延迟2天不会影响项目总工期。

2.3 软件质量管理

知识点1: 软件质量管理

影响软件质量的3组因素：

可理解性（我能理解它吗？）
可维修性（我能修复它吗？）
灵活性（我能改变它吗？）
可测试性（我能测试它吗？）



可移植性（我能在另一台机器上使用它吗？）
可再用性（我能再用它的某些部分吗？）
互运行性（我能把它和另一个系统结合吗？）

正确性（它按我的需要工作吗？）
健壮性（对意外环境它能适当地响应吗？）
效率（完成预定功能时它需要的计算机资源多吗？）
完整性（它是安全的吗？）
可用性（我能使用它吗？）
风险（能按预定计划完成它吗？）

软件质量控制与质量保证：

(1) 质量保证一般是每隔一定时间(例如，每个阶段末)进行的，主要通过系统的质量审计和过程分析来保证项目的质量。独特工具包括：质量审计和过程分析。

(2) **质量控制**是实时监控项目的具体结果，以判断它们是否符合相关质量标准，制定有效方案，以消除产生质量问题的原因。

(3) **质量保证的主要目标**

【事前预防】工作。

尽量在刚刚引入缺陷时即将其捕获，而不是让缺陷扩散到下一个阶段。

作用于**【过程】**而**【不是最终产品】**。

贯穿于**【所有的活动之中】**，而不是只集中于一点。

(4) 软件能力成熟度模型集成（CMMI）

<div>阶段式</div> <div>↓</div> <div>组织能力成熟度</div>	成熟度等级	特点
	初始级【L1】	随意且混乱、组织成功依赖于个人能力
	已管理级【L2】	项目级可重复【建立了项目级的控制过程】
	已定义级【L3】	组织级，文档化标准化
	定量管理级【L4】	量化式管理【过程性能可预测】
	优化级【L5】	持续优化

注：CMMI 另有连续式，其内容本质上与阶段式一致。

CMMI 文件体系：顶层方针文件--过程文件--规程文件--模板类文件

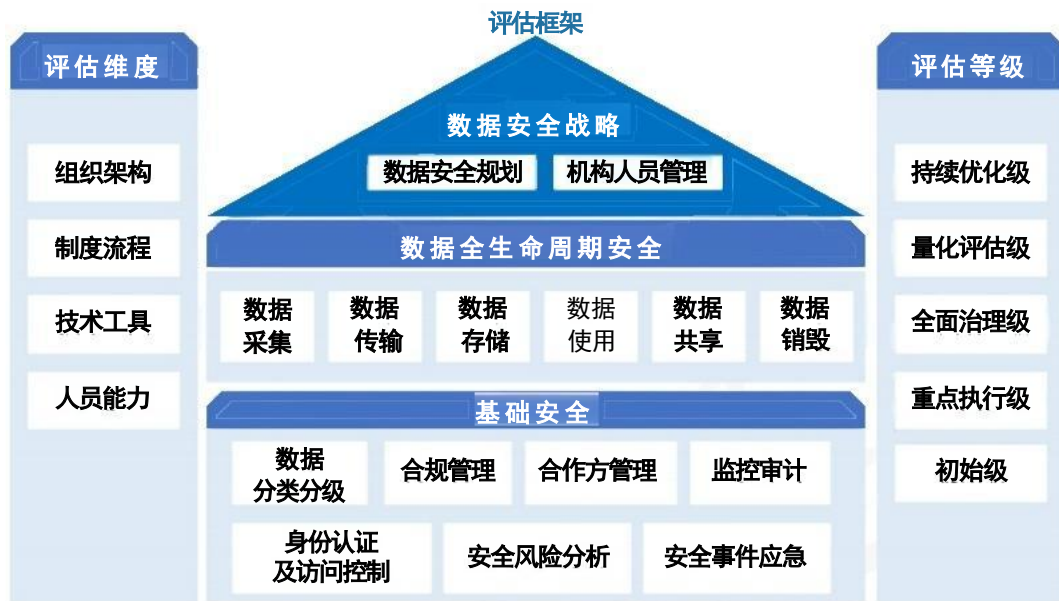
(5) 数据管理能力成熟度模型（DCMM）

【DCMM的8个核心能力域】定义了数据战略、数据治理、数据架构、数据应用、数据安全、数据质量、数据标准和数据生存周期等。

成熟度等级	描述
优化级【L5】	数据被认为是组织生存和发展的基础， 相关管理流程能实时优化 ，能在行业内进行最佳实践分享
量化管理级【L4】	数据被认为是获取竞争优势的重要资源， 数据管理的效率能量化分析和监控
稳健级【L3】	数据被当作绩效目标的重要资产，在组织层面制定了 标准化管理流程 ，促进数据管理规范化
受管理级【L2】	组织意识到数据是资产， 根据管理策略的要求制定管理流程 ，指定了相关人员进行初步管理
初始级【L1】	数据需求管理是项目级体现，没有统一管理流程， 被动式管理

(6) 数据安全治理的目标主要有三个：满足合规要求、管理数据安全风险、促进数据开发利用。这些目标旨在确保数据安全与业务发展的双向促进，同时保障组织在数据安全方面的合规性。

数据安全治理体系是组织达成数据安全治理目标需要具备的能力框架：



考点1：软件质量保证

【2011年下】软件质量保证是软件项目控制的重要手段，（）是软件质量保证的主要活动之一。

- A. 风险评估
- B. 软件评审
- C. 需求分析
- D. 架构设计

【参考答案】 B

【点拨】软件质量保证是软件质量管理的重要组成部分。软件质量保证主要是从软件产品的过程规范性角度来保证软件的品质。其主要活动包括：质量审计(包括软件评审)和过程分析。

考点2：CMMI

【2021年下】某软件企业在项目开发过程中目标明确，实施过程遵守既定的计划与流程，资源准备充分，权责到人，对整个流程进行严格的监测、控制与审查，符合企业管理体系与流程制度。因此，该企业达到了CMMI 评估的（）。

- A.可重复级
- B. 已定义级
- C. 量化级
- D. 优化级

【参考答案】B

【点拨】本题考查 CMMI 各级需要达到的规范程度，题目中虽未明示管理过程域，但体现的思想是符合企业的体系与流程，而可重复级仅到项目层次，只有到已定义级，才是针对企业，而此时又未强调量化，所以应选已定义级。

【2022年下】CMMI 是软件企业进行多方面能力评价的、集成的成熟度模型，软件企业在实施过程中，为了达到本地化，应组织体系编写组，建立基于CMMI 的软件质量管理体系文件，体系文件的层次结构一般分为四层，包括：

①顶层方针②模板类文件③过程文件④规程文件

按照自顶向下的塔型排列，以下顺序正确的是（）。

A.①④③②

B.①④②③

C.①②③④

D.①③④②

【参考答案】D

【点拨】各个过程定义的负责人根据CMMI 体系文件编写规范，完成相关文件的编写。

相关文件自顶向下排列为：

1) 方针文件：过程文件及其他相关文件都应遵循方针文件

2) 过程文件：根据过程文件模板编制各 PA 过程文件，并对流程进行描述。

3) 指南/规范文件：结合相应模板清晰说明如何完成这项工作，并说明对应的标准和需求。

4) 模板文件：模板分为两类，一类是word 文档模板，一类是excel 表格模板。word 文档：尽量利用公司现有的模板来制作，对需要填写替换的部分，给出具体解释或举例说明； excel 表格：主要用于需要自动计算数据的模板，对需要填写替换的部分，给出具体解释或举例说明。所以答案选择D 选项。

考点3: 数据管理能力成熟度模型 (DCMM)

【2022年下】数据管理能力成熟度评估模型 (DCMM) 是我国首个数据管理领域的国家标准，DCMM 提出了符合我国企业的数据管理框架，该框架将组织数据管理能力划分为8个能力域，分别为：数据战略、数据治理、数据架构、数据标准、数据质量、数据安全、（）。

A. 数据应用和数据生存周期

B. 数据应用和数据测试

C.数据维护和数据生存周期

D.数据维护和数据测试

【参考答案】A

考点4: 数据安全治理

【2024年下】数据安全治理的目标主要包括()三个方面。数据安全治理体系是组织达成数据安全治理目标需要具备的能力框架,其中数据分类分级属于该体系中的()。

A.满足用户需求、满足技术安全规范、促进数据开发利用

B.满足合规需求、管理用户安全风险、满足数据安全规范

C.满足用户需求、管理用户安全风险、促进数据开发利用

D.满足合规要求、管理数据安全风险、促进数据开发利用

A.数据全生命周期安全层

B.访问权限控制层

C.数据安全战略层

D.基础安全层

【参考答案】DD

2.4软件配置管理

知识点1: 软件配置管理

(1) **产品配置**是指一个产品在其生命周期各个阶段所产生的各种形式(机器可读或人工可读)和各种版本的文档、计算机程序、部件及数据的集合。

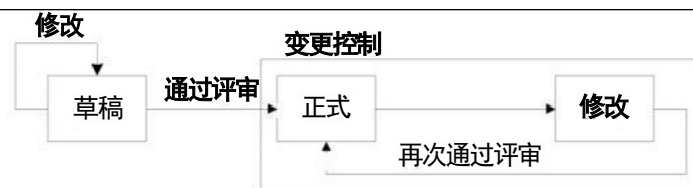
(2) 关于配置项

基线配置项(可交付成果):需求文档、设计文档、源代码、可执行代码测试用例、运行软件所需数据等。

非基线配置项: 各类计划 (如项目管理计划, 进度管理计划)、各类报告。

软件配置管理核心内容包括 **【版本控制】**和**【变更控制】**。

版本控制



处于草稿状态的配置项的版本号格式为：0.YZ，其中YZ 数字范围为01~99。随着草稿的不断完善，YZ的取值应递增。YZ的初值和增幅由开发者自己把握。

处于正式发布状态的配置项的版本号格式为：X.Y。其中X 为主版本号，取值范围为1~9；Y 为次版本号，取值范围为1~9。配置项第一次正式发布时，版本号为1.0。

如果配置项的版本升级幅度比较小，一般只增大Y 值，X 值保持不变。只有当配置项版本升级幅度比较大时，才允许增大X 值。

处于正在修改状态的配置项的版本号格式为：X.YZ。在修改配置项时，一般只增大Z 值，X.Y 值保持不变。

软件工具

按软件过程活动将软件工具分为：

软件开发工具：需求分析工具、设计工具、编码与排错工具。

软件维护工具：版本控制工具 (VSS、CVS、ScCS、SVN)、文档分析工具、开发信息库工具、逆向工程工具、再工程工具。

软件管理和软件支持工具：项目管理工具、配置管理工具、软件评价工具等。

配置管理工具的常见功能包括版本控制、变更管理、配置状态管理、访问控制和安全控制等。配置管理工具是包含了版本控制工具的。

版本控制工具用来存储、更新、恢复和管理一个软件的多个版本。

考点1：配置项

【2009年下】配置项是构成产品配置的主要元素，其中()不属于配置项。

- A. 设备清单
- B. 项目质量报告
- C. 源代码
- D. 测试用例

【参考答案】A

【点拨】配置项是构成产品配置的主要元素，配置项主要有以下两大类：

(1)属于产品组成部分的工作成果：如需求文档、设计文档、源代码和测试用例等；

(2) 属于项目管理和机构支撑过程域产生的文档：如工作计划、项目质量报告和项目跟踪报告等。这些文档虽然不是产品的组成部分，但是值得保存。所以设备清单不属于配置项。

【2011年下】软件产品配置是指一个软件产品在生存周期各个阶段所产生的各种形式和各种版本的文档、计算机程序、部件及数据的集合。该集合的每一个元素称为该产品配置中的一个配置项。下列不应该属于配置项的是（ ）。

- A. 源代码清单
- B. 设计规格说明书
- C. 软件项目实施计划
- D. CASE 工具操作手册

【参考答案】 D

【点拨】本题考查软件产品配置项的相关知识。源代码清单、设计规格说明书、软件项目实施计划均可以成为配置项。而工具操作手册是指导开发人员使用CASE 工具来做开发的一个说明文档，它与软件产品并无直接关联，不宜作为配置项。

考点2: 版本控制

【2015年下】项目配置管理中，配置项的状态通常包括（ ）。

- A. 草稿、正式发布和正在修改
- B. 草稿、技术评审和正式发布
- C. 草稿、评审或审批、正式发布
- D. 草稿、正式发布和版本变更

【参考答案】 A

【点拨】配置项的状态有3种：“草稿” (Draft)、“正式发布” (Released) 和“正在修改” (Changing)。

配置项状态变迁如图所示：



第三章 软件工程

3.1 信息系统开发方法

知识点1: 信息系统开发方法

(1) 结构化开发方法

用户至上，**自顶向下**，逐步分解(求解)，严格区分工作阶段，每阶段有任务与成果，强调系统开发过程的整体性和全局性，系统开发过程工程化，文档资料标准化。

优点：

理论基础严密，它的指导思想是用户需求在系统建立之前就能被充分了解和理解。由此可见，结构化方法注重开发过程的整体性和全局性。

缺点：

开发周期长；文档、设计说明繁琐，工作效率低。

要求在开发之初全面认识系统的信息需求，充分预料各种可能发生的变化，但这并不十分现实；若用户参与系统开发的积极性没有充分调动，就会造成系统交接过程不平稳，使系统运行与维护管理难度加大。

阶段固化，不善变化，适用于需求明确的开发场景。

(2) 原型法开发方法

适用于**需求不明确**的开发，按功能分为水平原型(界面)、垂直原型(复杂算法)；按最终结果分为抛弃式原型、演化式原型。原型法的特点在于原型法对用户的需求是动态响应、逐步纳入的，系统分析、设计与实现都是随着对一个工作模型的不断修改而同时完成的，相互之间并无明显界限，也没有明确分工。系统开发计划就是一个反复修改的过程。适于用户需求开始时定义不清、管理决策方法结构化程度不高的系统开发，开发方法更易被用户接受；但如果用户配合不好，盲目修改，就会拖延开发过程。

抛弃型原型 (Throw-It-Away Prototype)，此类原型在系统真正实现以后就放弃不用了。

演化型原型 (Evolutionary Prototype)，此类原型的构造从目标系统的一个或几个基本需求出发，通过修改和追加功能的过程逐渐丰富，演化成最终系统。

(3) 面向对象方法

最早来源于仿真领域，其特点是系统的描述及信息模型的表示与客观实体相对应，符合人们的思维习惯，有利于系统开发过程中用户与开发人员的交流和沟通，缩短开发周期，提供系统开发的准确性和效率。具有更好的复用性，关键在于建立一个全面、合理、统一的模型，分析、设计、实现三个阶段界限不明确。

(4) 面向服务的方法

以粗粒度、松散耦合的系统功能为核心，强调系统功能的标准化和构件化，加强了系统的灵活性、可复用性和可演化性。

从概念上讲，SOA 方法有三个主要的抽象级别：操作、服务、业务流程。

操作：代表单个逻辑工作单元 (LUW) 的事务。执行操作通常会导致读、写或修改一个或多个持久性数据。SOA 操作可以直接与面向对象 (OO) 的方法相比。它们都有特定的结构化接口，并且返回结构化的响应。完全同方法一样，特定操作的执行可能涉及调用附加的操作。操作位于最底层。

服务：代表操作的逻辑分组。例如，如果我们将Customer Profiling视为服务，则按照电话号码查找客户、按照名称和邮政编码列出顾客和保存新客户的数据就代表相关的操作。

业务流程：为实现特定业务目标而执行的一组长期运行的动作或活动。业务流程通常包括多个业务调用。业务流程的例子有：接纳新员工、出售产品或服务 and 完成订单。

(5) 其他开发方法

形式化方法(净室软件工程【受控污染级别的环境】;数学模型化;所有东西均可证明/验证,而不是测试)、统一过程方法、敏捷方法、基于架构的开发方法 (ABSD)

考点1: 原型开发方法

快速迭代式的原型开发能够有效控制成本，()是指在开发过程中逐步改进和细化原型直至产生出目标系统。

- A. 可视化原型开发
- B. 抛弃式原型开发
- C. 演化式原型开发
- D. 增量式原型开发

【参考答案】 C

【点拨】原型开发分两大类：快速原型法(又称抛弃式原型法)和演化式原型法。其中快速原型法是快速开发出一个原型，利用该原型获取用户需求，然后将该原型抛弃。而演化式原型法是将原型逐步进化为最终的目标系统。

考点2: 软件开发方法特点

【2011年下】下列关于各种软件开发方法的叙述中，错误的是()。

- A. 结构化开发方法的缺点是开发周期较长，难以适应需求变化

B. 可以把结构化方法和面向对象方法结合起来进行系统开发，使用面向对象方法进行自顶向下的划分，自底向上地使用结构化方法开发系统

C. 与传统方法相比，敏捷开发方法比较适合需求变化较大或者开发前期需求不是很清晰的项目，以它的灵活性来适应需求的变化

D. 面向服务的方法以粗粒度、松散耦合和基于标准的服务为基础，增强了系统的灵活性、可复用性和可演化性

【参考答案】 B

【点拨】 B 选项中“自底向上地使用结构化方法开发系统”显然是错误的，因为结构化方法的一个核心特色为：“自顶向下，逐步求精”，而非自底向上。

【模拟题练习】 以下关于软件开发方法的说法中，正确的是（）。

A. 面向服务的方法中服务的抽象层级属于低级，多个服务可以构成业务流程

B. 结构化开发方法是一种自顶向下的开发方法，该方法是建立在严格数学基础上的

C. 原型化开发方法可分为水平型原型和垂直型原型，其中垂直型原型适合解决复杂算法问题

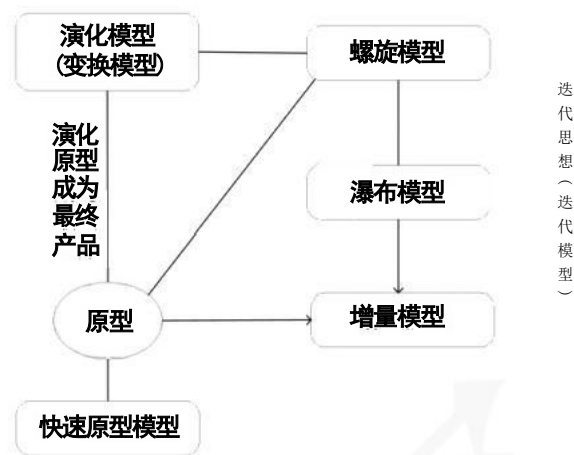
D. 面向对象方法更符合人们的思维习惯，但复用方面弱于结构化开发方法

【参考答案】 C

【点拨】 面向服务的方法中服务的抽象层级属于中级，而非低级。结构化开发方法是一种自顶向下的开发方法，但它不是建立在严格数学基础上的。建立在严格数学基础上的，是形式化方法。面向对象方法更符合人们的思维习惯，其复用方面强于结构化开发方法。

3.2 软件过程模型

知识点1: 软件过程模型



(1) 原型模型

典型的原型开发方法模型。适用于需求不明确的场景，可以帮助用户明确需求。可以分为【抛弃型原型】与【演化型原型】

原型模型两个阶段：

- 1、原型开发阶段；
- 2、目标软件开发阶段。

(2) 瀑布模型



瀑布模型是将软件生存周期中的各个活动规定为线性顺序连接的若干阶段的模型，包括需求分析、设计、编码、运行与维护。

瀑布模型的特点是严格区分阶段，每个阶段因果关系紧密相连，只适合需求明确的项目。

缺点：软件需求完整性、正确性难确定；

严格串行化，很长时间才能看到结果；

瀑布模型要求每个阶段一次性完全解决该阶段工作，这不现实。

(3) 增量模型

融合了瀑布模型的基本成分和原型实现的迭代特征，可以有多个可用版本的发布，核心功能往往最先完成，在此基础上，每轮迭代会有新的增量发布，核心功能可以得到充分测试。强调每一个增量均发布一个可操作的产品。

(4) 螺旋模型

以快速原型为基础+瀑布模型，典型特点是引入了风险分析。它是由制定计划、风险分析、实施工程、客户评估这一循环组成的，它最初从概念项目开始第一个螺旋。

(5) V 模型和W 模型

V 模型强调测试贯穿项目始终，而不是集中在测试阶段。是一种测试的开发模型。

W 模型强调测试和开发【并行进行】。

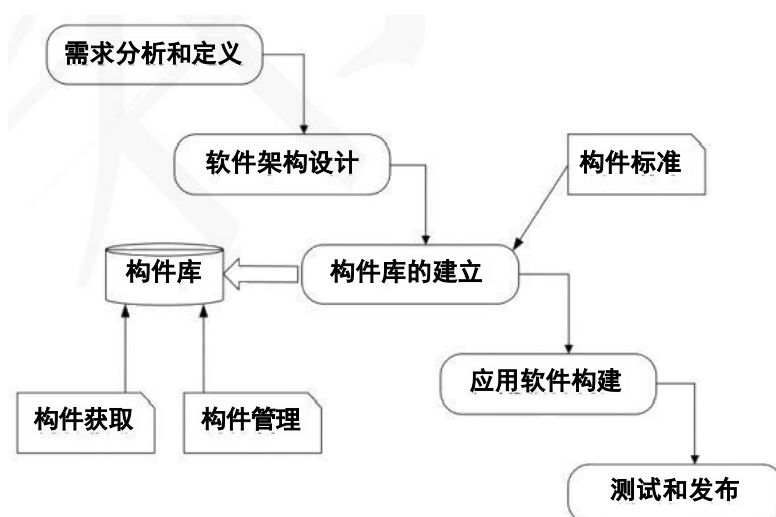
(6) 快速应用开发 RAD

概念：RAD 是瀑布模型的一个高速变种，适用比传统生命周期快得多的开发方法，它强调极短的开发周期，通常适用基于构件的开发方法获得快速开发。

过程：业务建模 → 数据建模 → 过程建模 → 应用生成 → 测试与交付

适用性：RAD 对模块化要求比较高，如果某项功能不能被模块化，则其构件就会出问题；如果高性能是一个指标，且必须通过调整结构使其适应系统构件才能获得，则RAD 也有可能不能奏效；RAD 要求开发者和客户必须在很短的时间完成一系列的需求分析，任何一方配合不当都会导致失败；RAD 只能用于管理信息系统的开发，不适合技术风险很高的情况。

(7) 构件组装模型



【优点】易扩展、易重用、降低成本、安排任务更灵活。

【缺点】构件设计要求经验丰富的架构师、设计不好的构件难重用、强调重用可能牺牲其它指标(如性能)、第三方构件质量难控制。

(8) 统一过程(在软考中UP、RUP 都指统一过程)

典型特点是用例驱动、以架构为中心、迭代和增量。

统一过程把一个项目分为四个不同的阶段：

构思阶段(初始/初启阶段)：定义最终产品视图和业务模型、确定系统范围。

细化阶段(精化阶段)：**设计及确定系统架构**、制定工作计划及资源要求。

构造阶段：开发剩余构件和应用程序功能，把这些构件集成为产品，并进行详细测试。

移交阶段：确保软件对最终用户是可用的，进行β测试，制作产品发布版本。

9个核心工作流：**业务建模、需求、分析与设计、实现、测试、部署、配置与变更管理、项目管理、环境。**

(10) 敏捷开发

敏捷开发是一种以人为核心、迭代、循序渐进的开发方法，适用于小团队和小项目，具有小步快跑的思想。常见的敏捷开发方法有极限编程法、水晶法、并列争球法和自适应软件开发方法。

极限编程 (XP)：在一些对费用控制严格的公司中的使用，非常有效，近螺旋式的开发方法。四大价值观(沟通【加强面对面沟通】、简单【不过度设计】、反馈【及时反馈】、勇气【接受变更的勇气】)，十二大最佳实践(简单设计、测试驱动、代码重构、结对编程、持续集成、现场客户、发行版本小型化、系统隐喻、代码集体所有制、规划策略、规范代码、40小时工作机制)。

水晶方法：提倡“机动性”的方法，拥有对不同类型项目非常有效的敏捷过程。

开放式源码：程序开发人员在地域上分布很广【其他方法强调集中办公】。

SCRUM：明确定义了可重复的方法过程。

特征驱动开发方法 (FDD)：认为有效的软件开发需要3要素【人、过程、技术】。定义了6种关键的项目角色：**项目经理、首席架构设计师、开发经理、主程序员、程序员和领域专家。**

ASD 方法：其核心是三个非线性的、重叠的开发阶段：猜测、合作与学习。

动态系统开发方法 (DSDM)：倡导以业务为核心。

敏捷宣言：

个体和交互胜过过程和工具；

可工作的软件胜过大量的文档；

客户合作胜过合同谈判；

响应变化胜过遵循计划。

考点1: 模型的概念及特点

【2010年下】() 把整个软件开发流程分成多个阶段, 每一个阶段都由目标设定、风险分析、开发和有效性验证以及评审构成。

- A. 原型模型
- B. 瀑布模型
- C. 螺旋模型
- D. V 模型

【参考答案】C

【2018年下】软件开发过程模型中, () 主要由原型开发阶段和目标软件开发阶段构成。

- A. 原型模型
- B. 瀑布模型
- C. 螺旋模型
- D. 基于构件的模型

【参考答案】A

考点2: 模型的适用场景

【2009年下】() 方法以原型开发思想为基础, 采用迭代增量式开发, 发行版本小型化, 比较适合需求变化较大或者开发前期对需求不是很清晰的项目。

- A. 信息工程
- B. 结构化
- C. 面向对象
- D. 敏捷

【参考答案】D

【模拟题练习】快速应用开发 (Rapid Application Development, RAD) 是一种比传统生存周期法快得多的开发方法, 它强调极短的开发周期。RAD 模型是() 的一个高速变种, 通过使用() 方法获得快速开发。以下哪种情形适合使用快速应用开发方法()。

- | | | | |
|--------------|---------------------|---------|-----------|
| A. 瀑布模型 | B. 原型模型 | C. 螺旋模型 | D. 增量模型 |
| A. 基于构件的软件 | B. 基于服务的开发 | C. 统一过程 | D. 面向对象开发 |
| A. 系统模块化程度较高 | B. 新系统与现有系统有较高的互操作性 | | |

C. 用户不愿参与到需求分析中

D. 技术层面难度不大、但需要采用很多新技术

【参考答案】AAA

【. 点拨】

快速应用开发 (Rapid Application Development, RAD) 是一种比传统生存周期法快得多的开发方法，它强调极短的开发周期。RAD 模型是瀑布模型的一个高速变种，通过使用基于构件的开发方法获得快速开发。如果需求理解得很好，且约束了项目范围，利用这种模型可以很快地开发出功能完善的信息系统。但是 RAD 也具有以下局限性：

①并非所有应用都适合 RAD。RAD 对模块化要求比较高，如果有哪一项功能不能被模块化，那么 RAD 所需要的构建就会有问题；如果高性能是一个指标，且该指标必须通过调整接口使其适应系统构件才能获得，则 RAD 也有可能不能奏效。

②开发者和客户必须在很短的时间完成一系列的需求分析，任何一方配合不当，都会导致 RAD 项目失败。

③RAD 只能用于管理信息系统的开发，不适合技术风险很高的情况。例如，当一个新系统要采用很多新技术，或当新系统与现有系统有较高的互操作性时，就不适合使用 RAD。

3.3 基于构件的软件工程 (CBSE)

知识点1: 基于构件的软件工程

CBSE 体现了“购买而不是重新构造”的哲学。

CBSE 的构件应该具备的特征：

- 1、可组装性：所有外部交互必须通过公开定义的接口进行。
- 2、可部署性：构件总是二进制形式的，能作为一个独立实体在平台上运行。
- 3、文档化：用户根据文档来判断构件是否满足需求。
- 4、独立性：可以在无其他特殊构件的情况下进行组装和部署。
- 5、标准化：符合某种标准化的构件模型。

【构件的组装】：

- 1、**顺序组装**：按顺序调用已经存在的构件，可以用两个已经存在的构件来创建一个新的构件。
- 2、**层次组装**：被调用构件的“提供”接口必须和调用构件的“请求”接口兼容。
- 3、**叠加组装**：多个构件合并形成新构件，新构件整合原构件的功能，对外提供新的接口。

【构件模型要素】

【接口】 构件通过构件接口来定义，构件模型规定应如何定义构件接口以及在接口定义中应该包含的要素，如操作名、参数以及异常等。

【使用信息】 为使构件远程分布和访问，必须给构件一个特定的、全局唯一的名字或句柄。 **构件元数据是构件本身相关的数据**，比如构件的接口和属性信息。用户可以通过元数据找到构件提供的服务。构件模型的实现通常包括访问构件的元数据的特定方法。 **构件是通用实体，在部署的时候，必须对构件进行配置来适应应用系统。**

【部署】 构件模型包括一个规格说明，指出应该如何打包构件使其部署成为一个独立的可执行实体。部署信息中包含有关包中内容的信息和它的二进制构成的信息。

考点1: 构件的组装

【模拟题练习】 构件的组装有多种组装方式，其中，（）被调用构件的“提供”接口必须和调用构件的“请求”接口兼容。

- A. 顺序组装
- B. 层次组装
- C. 自适应组装
- D. 叠加组装

【参考答案】 B

考点2: 构件模型要素

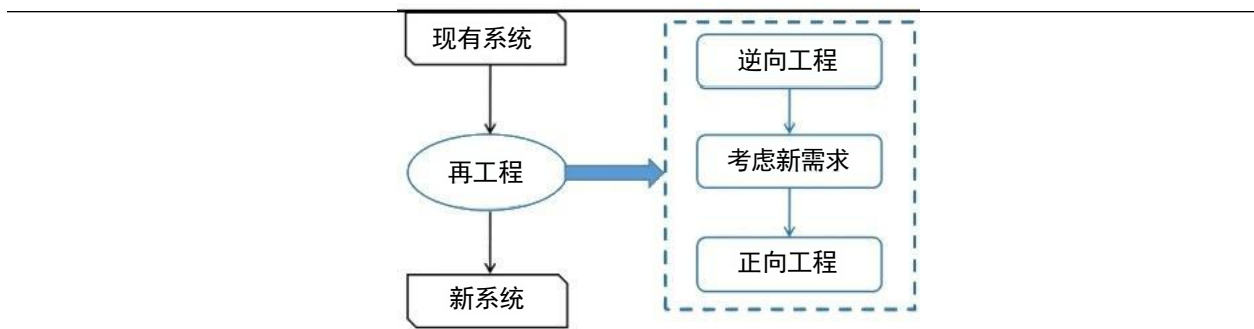
【2024年上】 以下关于构件的描述中，（）是不正确的。

- A. 构件是二进制形式，无需在部署前编译
- B. 构件元数据是构件本身相关的数据
- C. 构件是通用实体，不能对构件进行配置来适应应用系统
- D. 构件是一个独立的软件单元

【参考答案】 C

3. 4逆向工程

知识点1: 逆向工程



【逆向工程是设计的恢复过程】

实现级：包括程序的抽象语法树、符号表、过程的设计表示。

结构级：包括反映程序分量之间相互依赖关系的信息，例如调用图、结构图、程序和数据结构。

功能级：包括反映程序段功能及程序段之间关系的信息，例如数据和控制流模型。

领域级：包括反映程序分量或程序诸实体与应用领域概念之间对应关系的信息，例如实体关系模型。

与逆向工程相关的概念有重构、设计恢复、再工程和正向工程。

(1) 重构/重组 (restructuring)。 重构是指在【同一抽象级别】上转换系统描述形式。

(2) 设计恢复 (design recovery)。设计恢复是指借助工具从已有程序中抽象出有关数据设计、总体结构设计和过程设计等方面的信息。

(3) 逆向工程 (reverse engineering)：逆向工程是分析程序，力图在比源代码更高抽象层次上建立程序的表示过程，逆向工程是设计的恢复过程。

(4) 正向工程 (forward engineering)。正向工程是指不仅从现有系统中恢复设计信息，而且使用该信息去改变或重构现有系统，以改善其整体质量。

(5) 再工程/重构工程 (Re-engineering)。再工程是对现有系统的重新开发过程，包括逆向工程、新需求的考虑过程和正向工程三个步骤。

考点1：逆向工程的抽象层次

【2009年下】逆向工程导出的信息可以分为4个抽象层次，其中()可以抽象出程序的抽象语法树、符号表等信息；()可以抽象出反映程序段功能及程序段之间关系的信息。

A. 实现级 B. 结构级 C. 功能级 D. 领域级

A. 实现级 B. 结构级 C. 功能级 D. 领域级

【参考答案】 AC

考点2：逆向工程相关概念

【2017年下】应用系统构建中可以采用多种不同的技术，（）可以将软件某种形式的描述转换为更高级的抽象表现形式，而利用这些获取的信息，（）能够对现有系统进行修改或重构，从而产生系统的一个新版本。

- | | |
|-------------------------------|------------------------------|
| A. 逆向工程 (Reverse Engineering) | B. 系统改进 (System Improvement) |
| C. 设计恢复 (Design Recovery) | D. 再工程 (Re-engineering) |
| A. 逆向工程 (Reverse Engineering) | B. 系统改进 (System Improvement) |
| C. 设计恢复 (Design Recovery) | D. 再工程 (Re-engineering) |

【参考答案】 AD

3.5 净室软件工程

知识点1: 净室软件工程

净室即无尘室、洁净室。也就是一个**受控污染级别的环境**。

使用盒结构规约(或形式化方法)进行分析和设计建模，并且强调将正确性验证，而不是测试，作为发现和消除错误的主要机制。

使用统计的测试来获取认证被交付的软件的可靠性所必需的出错率信息。

【技术手段】：

1. 统计过程控制下的增量式开发：**控制迭代**

2. 基于函数的规范和设计：盒子结构

定义3种抽象层次：行为视图（**黑盒**）→ 有限状态机视图（**状态盒**）→ 过程视图（**明盒**）

3. **正确性验证：净室工程的核心**

4. 统计测试和软件认证：使用统计学原理，总体太大时必须采用抽样方法。

【缺点】：

太理论化，正确性验证的步骤比较困难且耗时。

开发小组不进行传统的模块测试，这是不现实的。

脱胎于传统软件工程，不可避免带有传统软件工程的一些弊端。

考点1: 净室软件工程的技术手段

以下关于软件开发方法的叙述，错误的是（）。

- A. 对于较为复杂的应用问题，适合采用形式化方法进行需求分析
- B. 形式化方法的优势在于能够精确地表述和研究应用问题及其软件实现

C. 净室软件工程将正确性验证作为发现和排除错误的主要机制

D. 净室软件工程强调统计质量控制技术，包括客户对软件使用预期的测试

【参考答案】 A

【点拨】形式化方法是一种具有坚实数学基础的方法，从而允许对系统和开发过程做严格处理和论证，适用于那些系统安全级别要求极高的软件的开发。形式化方法的主要优越性在于它能够数学地表述和研究应用问题及软件实现。但是它要求开发人员具备良好的数学基础。用形式化语言书写的大型应用问题的软件规格说明往往过于细节化，并且难为用户和软件设计人员所理解。由于这些缺陷，形式化方法在目前的软件开发实践中并未得到普遍应用。

净室软件工程 (Cleanroom Software Engineering, CSE) 是软件开发的一种形式化方法，可以开发较高质量的软件。它使用盒结构规约进行分析和建模，并且将正确性验证作为发现和排除错误的主要机制，使用统计测试来获取认证软件可靠性所需要的信息。CSE 强调在规约和设计上的严格性，还强调统计质量控制技术，包括基于客户对软件的预期使用测试。

【模拟题练习】净室软件工程是一种应用数学与统计学理论以经济的方式生产高质量软件工程的技术，力图通过严格的工程化的软件过程达到开发中的零缺陷或接近零缺陷。以下技术手段中，() 是净室软件工程的核心。

A. 统计控制下的增量式开发

B. 基于函数的规范和设计

C. 正确性验证

D. 统计测试和软件认证

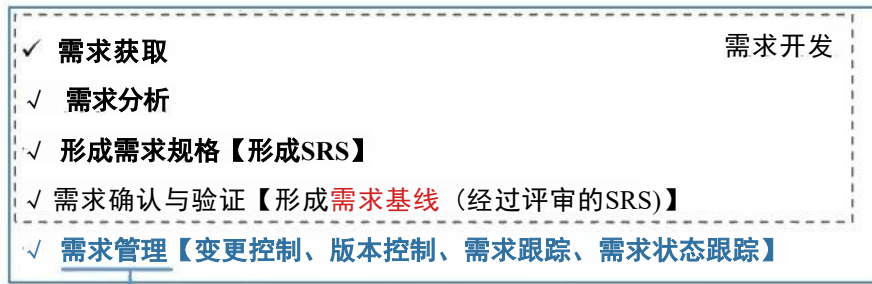
【参考答案】 C

3.6 需求工程

知识点1: 需求工程阶段划分

软件需求是指用户对系统在功能、行为、性能、设计约束等方面的期望。

【需求工程主要活动的阶段划分】



考点1: 需求工程的主要活动

【2011年下】下列关于软件需求管理或需求开发的叙述中，正确的是（）。

- A. 所谓需求管理是指对需求开发的管理
- B. 需求管理包括：需求获取、需求分析、需求定义和需求验证
- C. 需求开发是将用户需求转化为应用系统成果的过程
- D. 在需求管理中，要求维持对用户原始需求和所有产品构件需求的双向跟踪

【参考答案】D

【点拨】需求管理是一种用于查找、记录、组织和跟踪系统需求变更的系统化方法。而非对需求开发的管理。需求开发包括：需求获取、需求分析、需求定义和需求验证，而非需求管理。需求的跟踪属于需求管理的范畴。

知识点2: 需求开发

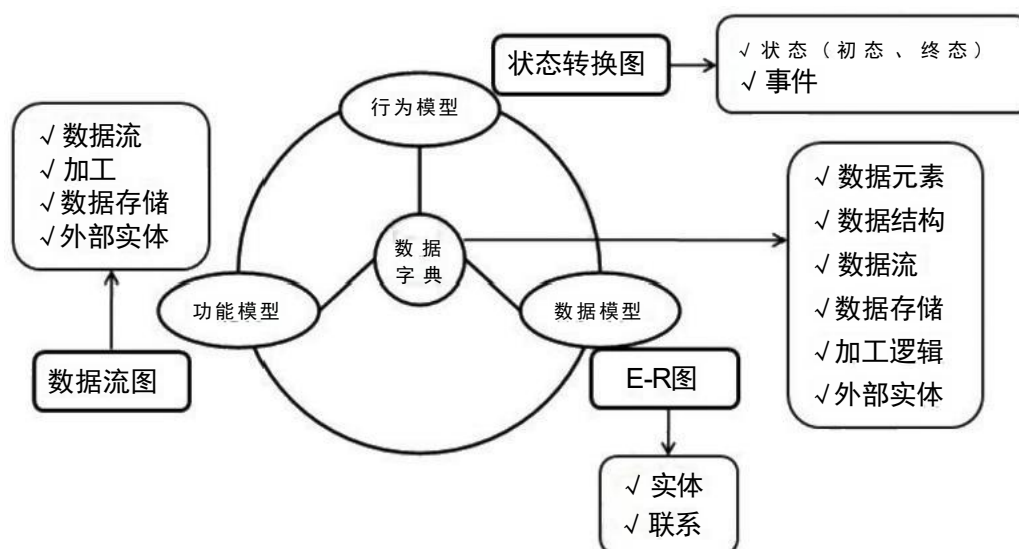
需求获取：



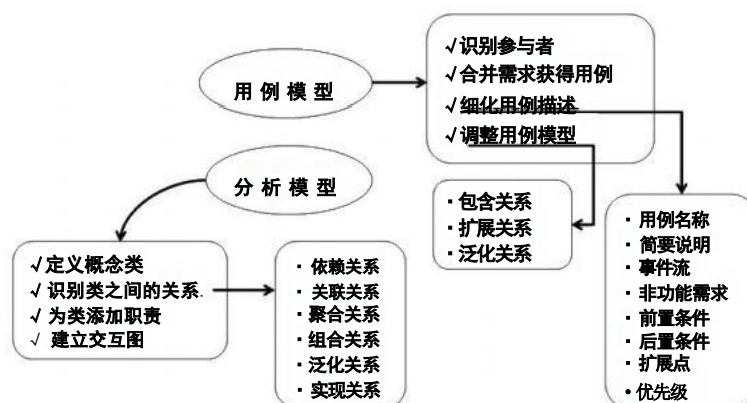
方法	特点
用户面谈	1对1-3, 有代表性的用户, 了解主观想法, 交互好。成本高, 要有领域知识支撑。
联合需求计划 (JRP)	高度组织的群体会议, 各方参与, 了解想法, 消除分歧, 交互好, 成本高。
问卷调查	用户多, 无法一一访谈, 成本低。
现场观察	针对较为复杂的流程和操作。
原型化方法	通过简易系统方式解决早期需求不确定问题。
头脑风暴法	一群人围绕新业务, 发散思维, 不断产生新的观点。

需求分析：

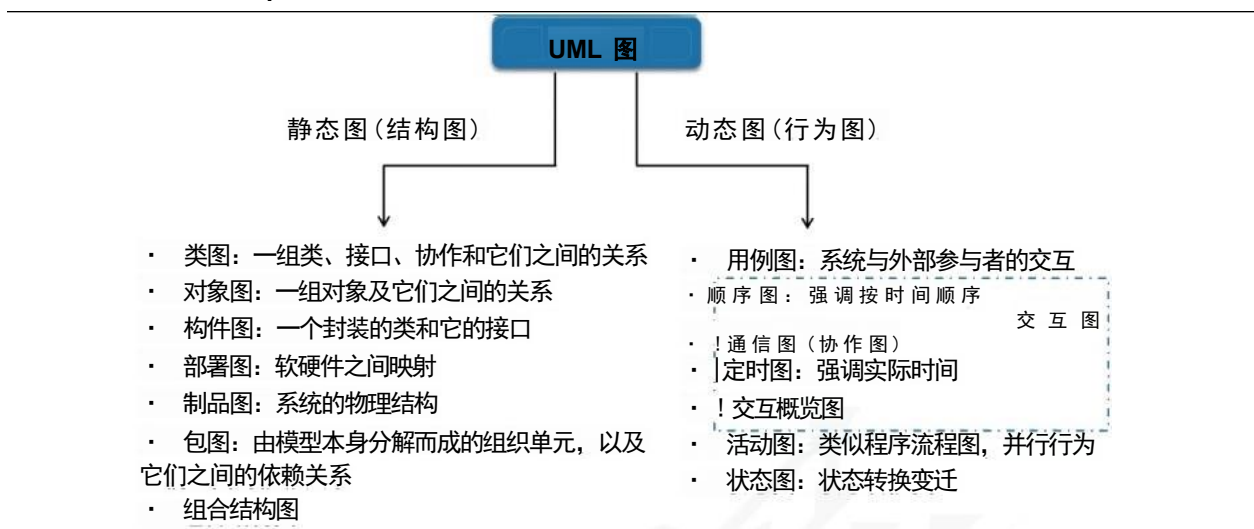
结构化需求分析 (SA)：



面向对象需求分析：



(1)UML 图分类



类图(Class diagram): 类图描述一组类、接口、协作和它们之间的关系。

对象图(Object diagram): 对象图描述一组对象及它们之间的关系。对象图描述了在类图中所建立的事物实例的静态快照。

构件图(Component diagram): 构件图描述一个封装的类和它的接口、端口, 以及由内嵌的构件和连接件构成的内部结构。构件图用于表示系统的静态设计实现视图。对于由小的部件构建大的系统来说, 构件图是很重要的。构件图是类图的变体。 一个封装的类和它的接口

部署图(Deployment diagram): 部署图描述对运行时的处理节点及在其中生存的构件的配置。部署图给出了架构的静态部署视图, 通常一个节点包含一个或多个部署图。软硬件之间映射。

制品图: 系统的物理结构

包图: 由模型本身分解而成的组织单元, 以及它们之间的依赖关系, 包的图标像是一个带标签的文件夹, 包的基本思想是把共同工作的元素放到一个文件夹中。例: 多个类或构件组成了一个子系统, 就可以将它们放到一个包中。

组合结构图

用例图: 系统与外部参与者的交互

顺序图(Sequence diagram): 又称序列图。是一种交互图 (Interaction diagram), 它强调对象之间消息发送的顺序, 同时显示对象之间的交互。强调按时间顺序。顺序图的组合片段: Loop【循环】: 如果满足“循环条件”, 则重复执行本框中的内容。 Alt【条件分支】: 满足条件1, 则执行条件1对应的内容, 满足条件2则执行条件2对应的内容。 Opt【可选分支】: 如果条件满足, 则执行框中内容, 否则跳过不执行。

通信图 (Communication diagram): 又称协作图。也是一种交互图，它强调收发消息的对象或参与者的结构组织。顺序图和通信图表达了类似的基本概念，但它们所强调的概念不同，顺序图强调的是时序，通信图强调的是对象之间的组织结构(关系)。

定时图：强调实际时间

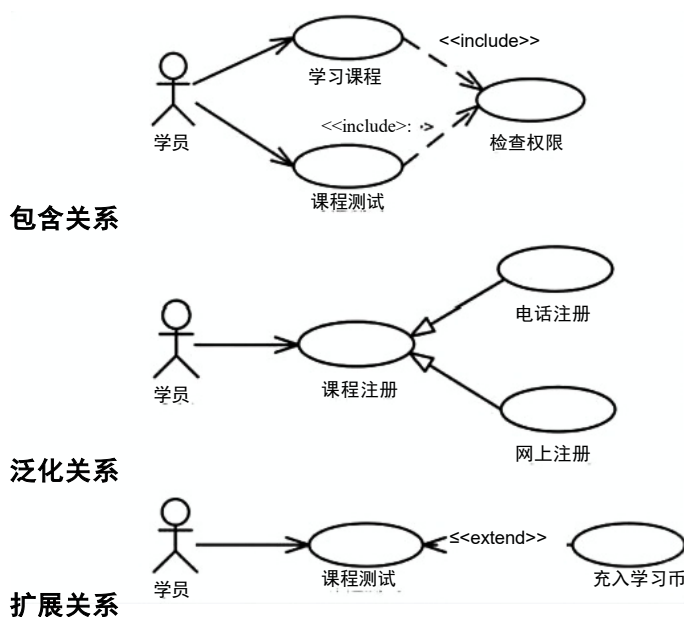
交互概览图

状态图 (State diagram): 状态图描述一个状态机，它由状态、转移、事件和活动组成。状态图给出了对象的动态视图。它对于接口、类或协作的行为建模尤为重要，而且它强调事件导致的对象行为，这非常有助于对反应式系统建模。状态转换变迁。

活动图 (Activity diagram): 活动图将进程或其他计算结构展示为计算内部一步步的控制流和数据流。活动图专注于系统的动态视图。它对系统的功能建模和业务流程建模特别重要，并强调对象间的控制流程。类似程序流程图，并行行为。

(2) UML 图关系

用例关系包括：包含关系、扩展关系、泛化关系。

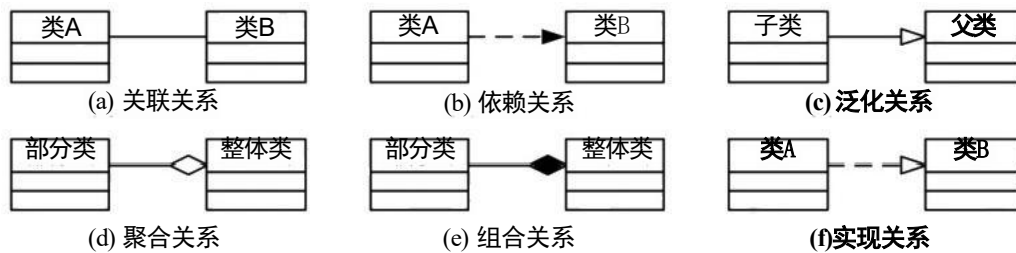


包含关系：其中这个提取出来的公共用例称为**抽象用例**，而把原始用例称为基本用例或基础用例，当可以从两个或两个以上的用例中提取公共行为时，应该使用包含关系来表示它们。

扩展关系：如果一个用例明显地混合了两种或两种以上的不同场景，即根据情况可能发生多种分支，则可以将这个用例分为一个**基本用例**和一个或多个**扩展用例**，这样使描述可能更加清晰。

泛化关系：当多个用例共同拥有一种类似的结构和行为的时候，可以将它们的共性抽象成为父用例，其他的用例作为泛化关系中的子用例。在用例的泛化关系中，子用例是父用例的一种特殊形式，子用例继承了父用例所有的结构、行为和关系。

类图/对象图关系：



依赖关系： 一个事物发生变化影响另一个事物。

泛化关系： 特殊/一般关系

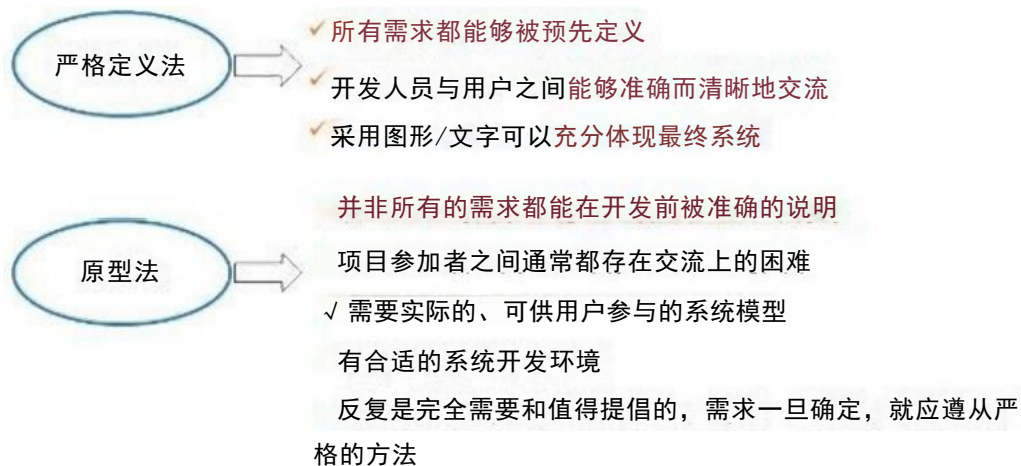
关联关系： 描述了一组链，链是对象之间的连接。

聚合关系： 整体与部分生命周期不同。

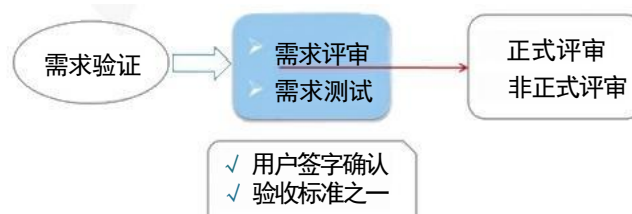
组合关系： 整体与部分生命周期相同。

实现关系： 接口与类之间的关系

需求定义（形成需求规格）：



需求确认与验证：



考点1：需求获取方法

【2014年下】下列关于联合需求计划（Joint Requirement Planning, JRP）的叙述中，不正确的是（ ）。

A. 在 JRP 实施之前，应制定详细的议程，并严格遵照议程进行

B. 在讨论期间尽量避免使用专业术语

C.JRP 是一种相对来说成本较高但十分有效的需求获取方法

D. JRP 的主要目的是对需求进行分析和验证

【参考答案】D

【点拨】JRP 的主要意图是收集需求，而不是对需求进行分析和验证。

考点2: 需求分析方法

【2014年下】在结构化分析方法中，用()表示功能模型，用()表示行为模型。

A.ER 图

B.用例图

C.DFD

D. 对象图

A.通信图

B.顺序图

C.活 动 图

D. 状态转换图

【参考答案】CD

【点拨】在结构化分析中，主要进行三个方面的建模：功能建模、行为建模和数据建模。功能建模一般采用 DFD，行为建模一般采用状态转换图，数据建模一般采用ER 图。

【2021年下】结构化设计是一种面向数据流的设计方法，以下不属于结构化设计工具的是()。

A. 盒图

B.HIPO图

C.顺序图

D.程序流程图

【参考答案】C

【点拨】顺序图属于面向对象分析与设计工具，而非结构化设计工具。答案选择C 选项。

考点3: 需求定义方法

【2011年下】通常有两种常用的需求定义方法：严格定义方法和原型方法。下述的各种假设条件中，“()” 不适合使用严格定义方法进行需求定义。

A. 所有需求都能够被预先定义

B. 开发人员与用户之间能够准确而清晰地交流

C. 需求不能在系统开发前被完全准确地说明

D. 采用图形(或文字)充分体现最终系统

【参考答案】C

【点拨】需求定义方法包括严格定义方法和原型方法两种。严格定义方法适用于需求已全面获取，需求较为明确的情况。如果达不到这个要求，则适宜用原型方法。

考点4: 需求基线

【2020年下】软件需求开发的最终文档经过评审批准后，就定义了开发工作的()，它在客户和开发者之间构筑了产品功能需求和非功能需求的一个()，是需求开发和需求管理之间的桥梁。

- A.需求基线 B. 需求标准 C. 需求用例 D.需求分析
A. 需求用例 B.需求管理标准 C.需求约定 D.需求变更

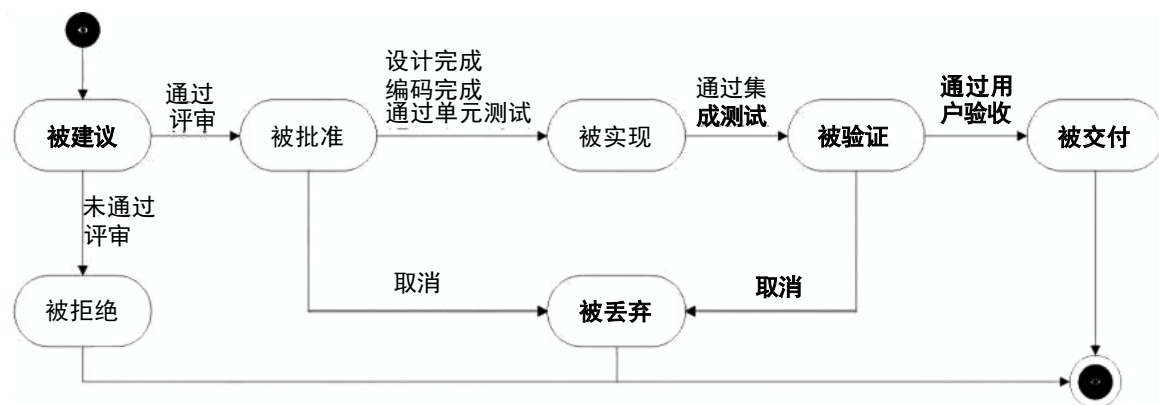
【参考答案】 AC

【点拨】需求开发的结果应该有项目视图和范围文档、用例文档和 SRS, 以及相关的分析模型。经评审批准，这些文档就定义了开发工作的需求基线。本题第一空描述的是需求基线，选择A 选项。

这个基线在用户和开发人员之间就构成了软件需求的一个约定，它是需求开发和需求管理之间的桥梁。第二空选择C选项。

知识点3: 需求管理

(1) 定义需求基线



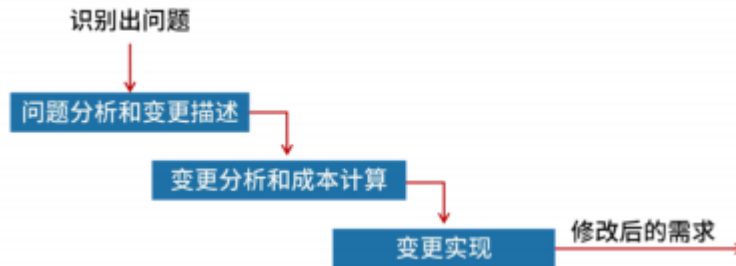
需求的状态

(2) 需求跟踪



(3) 变更控制

带有风险的做法：无足够用户参与，忽略了用户分类，用户需求的不断增加，模棱两可的需求，不必要的特性，过于精简的 SRS， 不准确的估算。



考点1：需求管理

【2009年下】以下关于需求管理的叙述中，正确的是（）。

- A. 需求管理是一个对系统需求及其变更进行了解和控制的过程
- B. 为了获得项目，开发人员可以先向客户做出某些承诺
- C. 需求管理的重点在于收集和分析项目需求
- D. 软件开发过程是独立于需求管理的活动

【参考答案】 A

【2018年下】需求管理是一个对系统需求变更、了解和控制的过程。以下活动中，（）不属于需求管理的主要活动。

- A. 文档管理
- B. 需求跟踪
- C. 版本控制
- D. 变更控制

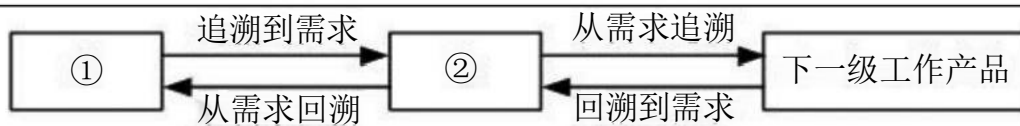
【参考答案】 A

【点拨】需求管理的活动包括：

- 1、变更控制；2、版本控制；3、需求跟踪；4、需求状态跟踪。

考点2：需求跟踪

【2011年下】利用需求跟踪能力链 (traceability link) 可以跟踪一个需求使用的全过程，也就是从初始需求到实现的前后生存期。需求跟踪能力链有4类，如下图所示：



其中的①和②分别是（）。

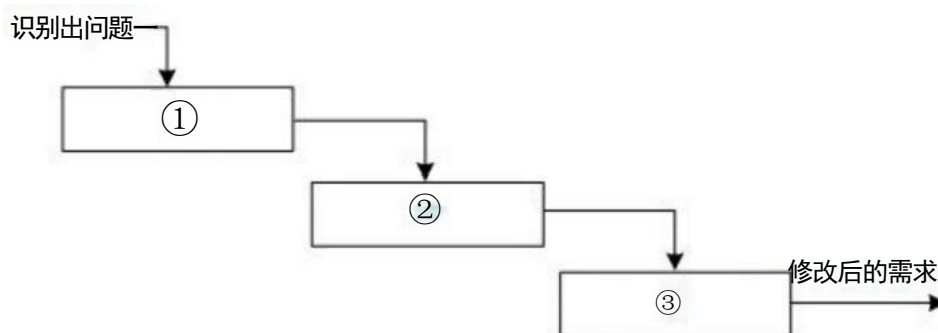
- A. 客户需求、软件需求
- B. 软件需求、客户需求
- C. 客户需求、当前工作产品
- D. 软件需求、当前工作产品

【参考答案】A

【点拨】本题考查需求跟踪相关内容。需求跟踪时，是分层次进行的，首先需要确认从用户方获取的需求，是否与软件需求能一一对应，然后再看软件需求到下一级工作产品之间是否存在一一对应的关系。这样层层传递的方式，可以尽量避免开发不需要的功能，以及遗漏该开发的内容。

考点3：变更控制

【2015年下】 一个大型软件系统的需求总是有变化的。为了降低项目开发的风险，需要一个好的变更控制过程。如下图所示的需求变更管理过程中，①②③处对应的内容应是（）；自动化工具能够帮助变更控制过程更有效地运作，（）是这类工具应具有的特性之一。

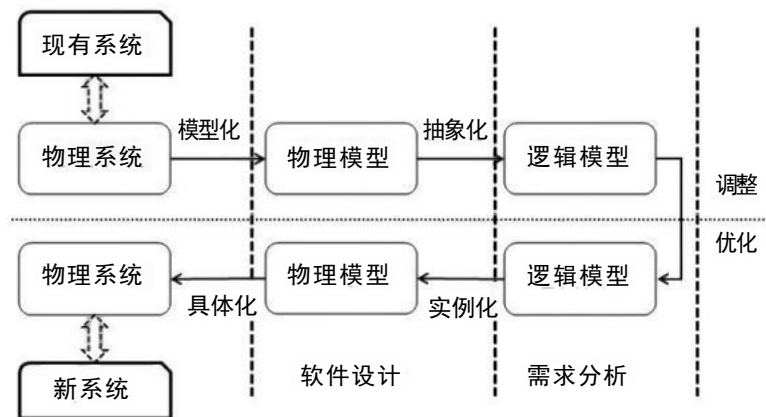


- A. 问题分析与变更描述，变更分析与成本计算，变更实现
- B. 变更描述与变更分析，成本计算，变更实现
- C. 问题分析与变更描述，变更分析，变更实现
- D. 变更描述，变更分析，变更实现

- A. 自动维护系统的不同版本
- B. 支持系统文档的自动更新
- C. 自动判定变更是否能够实施
- D. 记录每一个状态变更的日期及变更者

3.7 系统建模过程

知识点1： 系统建模



(1) 结构化建模方法：结构化建模方法是以过程为中心的技术，可用于分析一个现有的系统以及定义新系统的业务需求。结构化建模方法所绘制的模型称为数据流图（DFD）。对于流程较为稳定的系统可考虑结构化建模方法。

(2) 信息工程建模方法(或数据库建模方法)：信息工程建模方法是一种以数据为中心，但过程敏感的技术，它强调在分析和研究过程需求之前，首先研究和分析数据需求。信息工程建模方法所创建的模型被称为实体-联系图（ERD）。主要用于数据建模。

(3) 面向对象建模方法：面向对象建模方法将“数据”和“过程”集成到被称为“对象”的结构中，消除了数据和过程的人为分离现象。面向对象建模方法所创建的模型被称为对象模型。随着面向对象技术的不断发展和应用，形成了面向对象的建模标准，即 UML（统一建模语言）。UML 定义了几种不同类型的模型图，这些模型图以对象的形式共建一个信息系统或应用系统。目前比较常用的建模方法。

考点1: 系统建模方法

【模拟题练习】结构化建模方法和信息建模方法都是常见的信息系统建模方法，请问这两种建模方法的基本工具分别是（）。

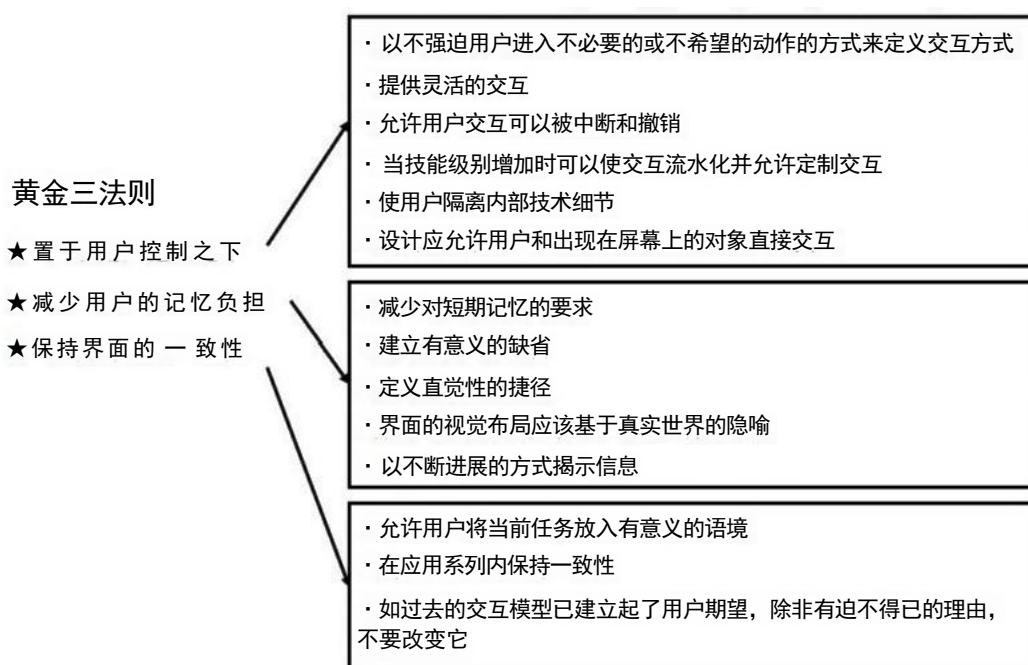
- A. 数据流图和实体联系图
- B. 数据流图和数据字典
- C. 流程图和 UML
- D. 数据流图和流程图

【参考答案】 A

3.8 系统设计

知识点1: 界面设计

用户界面设计是指用户与系统之间架起一座桥梁，主要内容包括：定义界面形式、定义基本的交互控制形成、定义图形和符号、定义通用的功能键和组合键的含义及其操作内容、定义帮助策略等。



考点1: 界面设计

【2014年下】下列关于用户界面设计的叙述中，错误的是()。

- A. 界面交互模型应经常进行修改
- B. 界面的视觉布局应该尽量与真实世界保持一致
- C. 所有可视信息的组织需要按照统一的设计标准
- D. 确保用户界面操作和使用的一致性

【参考答案】 A

知识点2: 结构化设计

(1) **概要设计【外部设计】**：功能需求分配给软件模块，确定每个模块的功能和调用关系，形成**模块结构图**。

(2) **详细设计【内部设计】**：为每个具体任务选择适当的技术手段和处理方法。

(3) 结构化设计原则：

模块独立性原则（高内聚、低耦合）；

保持模块的大小适中；

多扇入，少扇出；

深度和宽度均不宜过高。

(4) **模块四要素**：

输入和输出：模块的输入来源和输出去向都是同一个调用者，即一个模块从调用者那儿取得输入，进行加工后再把输出返回调用者。

处理功能：指模块把输入转换成输出所做的工作。

内部数据：指仅供该模块本身引用的数据。

程序代码：指用来实现模块功能的程序。

(5) 模块独立性的度量

1. 聚合：衡量模块内部各元素结合的紧密程度

内聚类型	描述
功能内聚	完成一个单一功能，各个部分协同工作，缺一不可。
顺序内聚	处理元素相关，而且必须顺序执行。
通信内聚	所有处理元素集中在一个数据结构的区域上。
过程内聚	处理元素相关，而且必须按特定的次序执行。
时间内聚(瞬时内聚)	所包含的任务必须在同一时间间隔内执行。
逻辑内聚	完成逻辑上相关的一组任务。
偶然内聚(巧合内聚)	完成一组没有关系或松散关系的任务。

2. 耦合：度量不同模块间互相依赖的程度

耦合类型	描述
非直接耦合	两个模块之间没有直接关系，它们之间的联系完全是通过主模块的控制和调用来实现的。
数据耦合	一组模块借助参数表传递简单数据。
标记耦合	一组模块通过参数表传递记录信息(数据结构)。
控制耦合	模块之间传递的信息中包含用于控制模块内部逻辑的信息。
外部耦合	一组模块都访问同一全局简单变量，而且不是通过参数表传递该全局变量的信息。
公共耦合	多个模块都访问同一个公共数据环境。
内容耦合	一个模块直接访问另一个模块的内部数据；一个模块不通过正常入口转到另一个模块的内部；两个模块有一部分程序代码重叠；一个模块有多个入口。

考点1: 概要设计和详细设计

【2011年下】系统设计是软件开发的重要阶段，（）主要是按系统需求说明来确定此系统的软件结构，并设计出各个部分的功能和接口。

- A. 外部设计
- B. 内部设计
- C. 程序设计
- D. 输入/输出设计

【参考答案】 A

【点拨】在软件开发中，外部设计又称为概要设计，其主要职能是设计各个部分的功能、接口、相互如何关联。内部设计又称为详细设计，其主要职能是设计具体一个模块的实现。所以本题应选A。

考点2: 模块独立性

【2021年下】软件设计过程中，可以用耦合和内聚两个定性标准来衡量模块的独立程度，耦合衡量不同模块彼此间互相依赖的紧密程度，应采用以下设计原则（），内聚衡量一个模块内部各个元素彼此结合的紧密程度，以下属于高内聚的是（）。

- A. 尽量使用内容耦合、少用控制耦合和特征耦合、限制公共环境耦合的范围、完全不用数据耦合
- B. 尽量使用数据耦合、少用控制耦合和特征耦合、限制公共环境耦合的范围、完全不用内容耦合
- C. 尽量使用控制耦合、少用数据耦合和特征耦合、限制公共环境耦合的范围、完全不用内容耦合
- D. 尽量使用特征耦合、少用数据耦合和控制耦合、限制公共环境耦合的范围、完全不用内容耦合

- A. 偶然内聚
- B. 时间内聚
- C. 功能内聚
- D. 逻辑内聚

【参考答案】BC

【模拟题练习】软件设计过程中，可以用耦合和内聚两个定性标准来衡量模块的独立程度，耦合衡量不同模块彼此间互相依赖的紧密程度，以下关于耦合的说法中正确的是()。

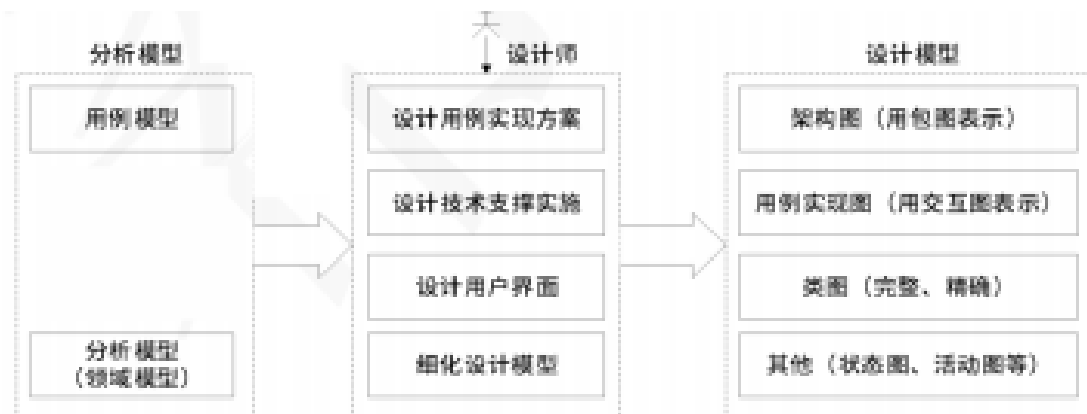
- A. 公共耦合的耦合性最高，所以要控制使用
- B. 标记耦合是指一组模块借助参数表传递简单数据
- C. 高耦合低内聚，耦合性越高越好
- D. 尽量使用数据耦合、少用控制耦合和特征耦合、限制公共环境耦合的范围、完全不用内容耦合

【参考答案】 D

【点拨】对题目选项出现的几种耦合的使用优先级进行排序。(耦合性越低越好) 非直接耦合>数据耦合>特征耦合>控制耦合>外部耦合>公共耦合>内容耦合

知识点3: 面向对象设计

(1) 过程



(2) 设计原则

单一职责原则：设计目的单一的类。

开放 - 封闭原则：对扩展开放，对修改封闭。

李氏 (Liskov) 替换原则：子类可以替换父类。

依赖倒置原则：要依赖于抽象，而不是具体实现；针对接口编程，不要针对实现编程。

接口隔离原则：使用多个专门的接口比使用单一的总接口要好。

组合重用原则：要尽量使用组合，而不是继承关系达到重用目的。

迪米特 (Demeter) 原则 (最少知识法则)：一个对象应当对其他对象有尽可能少的了解。

(3) 类的分类

类的分类



考点1: 面向对象设计过程

【2009年下】面向对象的设计模型包含以()表示的软件体系结构图，以()表示的用例实现图，完整精确的类图，针对复杂对象的状态图和用以描述流程化处理的活动图等。

- A. 部署图 B. 包图 C. 协同图 D. 交互图
- A. 部署图 B. 包图 C. 协同图 D. 交互图

【参考答案】BD

【点拨】面向对象的设计模型包含以包图表示的软件体系结构图，以交互图表示的用例实现图，完整精确的类图，针对复杂对象的状态图和用以描述流程化处理的活动图等。

考点2: 面向对象设计原则

【2011年下】最少知识原则 (也称为迪米特法则) 是面向对象设计原则之一，指一个软件实体应当尽可能少地与其他实体发生相互作用。这样，当一个实体被修改时，就会尽可能少地影响其他的实体。下列叙述中，“()” 不符合最少知识原则。

- A. 在类的划分上，应当尽量创建松耦合的类
- B. 在类的设计上，只要有可能，一个类型应当设计成不变类
- C. 在类的结构设计上，每个类都应当尽可能提高对其属性和方法的访问权限

D. 在对其他类的引用上，一个对象对其他对象的引用应当降到最低

【参考答案】C

【. 点拨】迪米特 (Demeter) 了 法则 (最少知识原则): 一个对象应当对其他对象有尽可能少的解。

迪米特法则的应用准则:

- 1) 在类的划分上，应当创建有弱耦合的类。类之间的耦合越弱，就越有利于复用。
- 2) 在类的结构设计上，每一个类都应当尽量降低成员的访问权限。一个类不应当public自己的属性，而应当通过提供取值和赋值的方法让外界间接访问自己的属性。
- 3) 在类的设计上，只要有可能，一个类应当设计成不变类。
- 4) 在对其他对象的引用上，一个类对其他对象的引用应该降到最低。

其中迪米特法则的主要理念是让一个对象尽可能少地了解其他对象，这样，就能尽可能少地产生违规操作，让设计出来的系统更稳定。在本题中，C 选项提到“尽可能提高对其属性和方法的访问权限”违背了迪米特法则。

【2015年下】在面向对象设计的原则中，()原则是指抽象不应该依赖于细节，细节应该依赖于抽象，即应针对接口编程，而不是针对实现编程。

- A. 开闭
- B. 里氏替换
- C. 最少知识
- D. 依赖倒置

【参考答案】D

考点3: 类的分类

【2013年下】在面向对象设计中，()可以实现界面控制、外部接口和环境隔离。()作为完成用例业务的责任承担者，协调、控制其他类共同完成用例规定的功能或行为。

- | | | | |
|--------|--------|--------|--------|
| A. 实体类 | B. 控制类 | C. 边界类 | D. 交互类 |
| A. 实体类 | B. 控制类 | C. 边界类 | D. 交互类 |

【参考答案】CB

【. 点拨】

1. 实体类

实体类映射需求中的每个实体，实体类保存需要存储在永久存储体中的信息，例如，在线教育平台系统可以提取出学员类和课程类，它们都属于实体类。实体类通常都是永久性的，它们所具有的属性和关系是长期需要的，有时甚至在系统的整个生存期都需要。

2. 控制类

控制类是用于控制用例工作的类，一般是由动宾结构的短语（“动词+名词”或“名词+动词”）转化来的名词，例如，用例“身份验证”可以对应于一个控制类“身份验证器”，它提供了与身份验证相关的所有操作。控制类用于对一个或几个用例所特有的控制行为进行建模，控制对象（控制类的实例）通常控制其他对象，因此，它们的行为具有协调性。

3. 边界类

边界类用于封装在用例内、外流动的信息或数据流。边界类位于系统与外界的交接处，包括所有窗体、报表、打印机和扫描仪等硬件的接口，以及与其他系统的接口。要寻找和定义边界类，可以检查用例模型，每个参与者和用例交互至少要有一个边界类，边界类使参与者能与系统交互。边界类是一种用于对系统外部环境与其内部运作之间的交互进行建模的类。常见的边界类有窗口、通信协议、打印机接口、传感器和终端等。实际上，在系统设计时，产生的报表都可以作为边界类来处理。

【模拟题练习】在面向对象设计中，（）保存需要存储在永久存储体中的信息。（）位于系统与外界的交接处，用于封装在用例内、外流动的信息或数据流。

- | | | | |
|--------|--------|--------|--------|
| A. 实体类 | B. 边界类 | C. 交互类 | D. 控制类 |
| A. 实体类 | B. 边界类 | C. 交互类 | D. 控制类 |

【参考答案】 AB

3. 9 软件测试

知识点1: 软件测试

(1) 动态测试与静态测试

【以测试过程中程序执行状态为依据可分类】

动态测试【计算机运行程序】

分为3个步骤：构造测试实例、执行程序以及分析结果。

静态测试【人工监测和计算机辅助分析】

桌前检查

代码审查

代码走查

控制流分析：是否存在没有使用的语句/无法达到的语句/调用并不存在的子程序。

数据流分析：引用未定义的变量、对以前未使用的变量再次赋值。

接口分析：模块之间接口的一致性、子程序和函数之间的接口一致性、函数形参与实参的数量、顺序、类型的一致性。

表达式分析：括号不配对、数组引用越界、除数为零。

(2) 黑盒测试与白盒测试

白盒测试【结构测试】：关注内部结构与逻辑。

控制流分析、数据流分析、路径分析、程序变异【错误驱动测试】	
【路径覆盖】(最强)	
【逻辑覆盖】(由强到弱排列) 修正条件/判定条件组合条件/判定覆盖	
条件覆盖 判定覆盖 语句覆盖	

黑盒测试【功能测试】：关注输入输出及功能。

等价类划分：不同等价类，揭示不同问题；有效等价类/无效等价类。
边界值分析：1<=x<=10, 可取x的值为0、1、10和11作为测试数据。
错误推测：依靠测试人员的经验和直觉。
判定表：最适合描述在多个逻辑条件取值的组合所构成的复杂情况下，分别要执行哪些不同的动作。
因果图：根据输入条件与输出结果之间的因果关系来设计测试用例。

(3) 自动化测试。自动化测试就是软件测试的自动化，即在预先设定的条件下自动运行被测程序，并分析运行结果。总的来说，这种测试方法就是将以前人驱动的测试行为转化为机器执行的一种过程。不适合场景：项目周期短，需求变动频繁。常见自动化测试：单元自动化测试、接口自动化测试、UI 自动化测试。

(4) 软件测试阶段

单元测试：依据【详细设计】，模块测试，模块功能、性能、接口等。

集成测试：依据【概要设计】，模块间的接口。

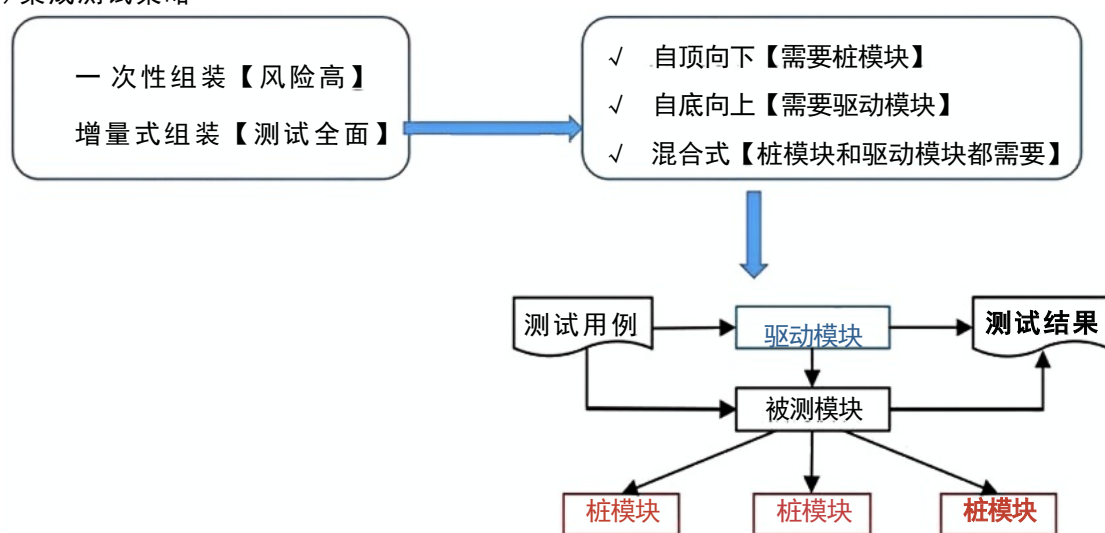
系统测试：依据【需求文档】，在真实环境下，验证完整的软件配置项能否和系统正确连接。

确认测试：依据【需求文档】，验证软件与需求的一致性。内部确认测试、Alpha 测试、Beta 测试、验收测试。

其他测试：

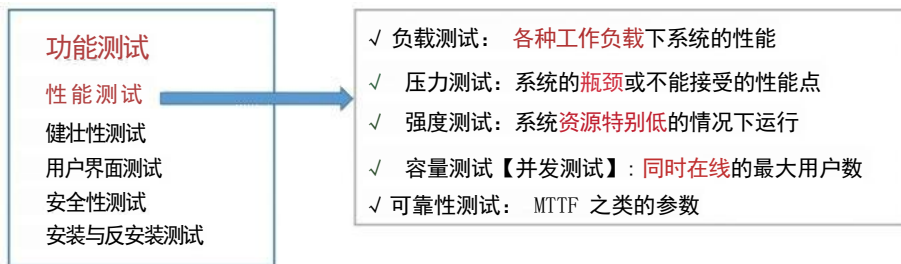
其他测试	描述
AB 测试	多版本同时使用，利于收集各版本的用户反馈，评估出最好版本。故也算是一种【网页优化方法】。
Web 测试	Web系统测试与其他系统测试测试内容基本相同，只是测试重点不同。 Web代码测试包括：源代码规则分析、链接测试、框架测试、表格测试、图形测试等方面。
链接测试	链接测试可分为3个方面： 1、测试所有链接是否按指示的那样确实链接到了该链接的页面。 2、测试所链接的页面是否存在。 3、保证Web应用系统上没有孤立的页面。
表单测试	验证服务器是否能正确保存这些数据，后台运行的程序能否正确解释和使用这些信息。 测试提交操作的完整性。
回归测试	测试软件变更之后，变更部分的正确性和对变更需求的符合性。

(5) 集成测试策略



(6) 系统测试

系统测试



系统测试活动与步骤：制订系统测试计划（进行人员以及任务的确定，明确测试范围、测试方法、测试环境与辅助工具）、设计系统测试用例（如：等价类划分、边界值分析等测试方法的应用）、执行系统测试（执行设计好的测试用例，并记录结果）、缺陷管理与改错（消除已发现的错误）。

(7) 软件测试与软件调试

软件调试方法：

蛮力法：主要思想是“通过计算机找错”，低效，耗时。

回溯法：从出错处人工沿控制流程往回追踪，直至发现出错的根源。复杂程序由于回溯路径多，难以实施。

原因排除法：主要思想是演绎和归纳，用二分法实现。

软件测试与软件调试的区别：调试是测试之后的活动，测试和调试在目标、方法和思路都有所不同。

软件测试	软件调试
目的是找出存在的错误	目的是定位错误并修改程序以修正错误
从一个已知的条件开始，使用预先定义的过程，有预知的结果	从一个未知的条件开始，结束的过程不可预计
测试过程可以事先设计，进度可以事先确定	调试不能描述过程或持续时间

考点1: 动态测试与静态测试

【2018年下】软件测试一般分为两个大类：动态测试和静态测试。前者通过运行程序发现错误，包括()等方法；后者采用人工和计算机辅助静态分析的手段对程序进行检测，包括()等方法。

A. 边界值分析、逻辑覆盖、基本路径

B. 桌面检查、逻辑覆盖、错误推测

C. 桌面检查、代码审查、代码走查

D. 错误推测、代码审查、基本路径

A. 边界值分析、逻辑覆盖、基本路径

B. 桌面检查、逻辑覆盖、错误推测

C. 桌面检查、代码审查、代码走查

D. 错误推测、代码审查、基本路径

【参考答案】 AC

【. 点拨】本题考查测试的分类，测试可以分为动态测试与静态测试。

动态测试是通过运行程序发现错误，包括黑盒测试(等价类划分、边界值分析法、错误推测法)与白盒测试(各种类型的覆盖测试)。

静态测试是人工测试方式，包括桌前检查(桌面检查)、代码走查、代码审查。

考点2: 黑盒测试与白盒测试

【2012年下】以下关于黑盒测试用例设计方法的叙述，错误的是()。

A. 边界值分析通过选择等价类边界作为测试用例，不仅重视输入条件边界，而且也必须考虑输出域边界

B. 因果图方法是从用自然语言书写的程序规格说明的描述中找出因(输入条件)和果(输出或程序状态的改变)，可以通过因果图转换为判定表

C. 正交试验设计法，就是使用已经造好了的正交表格来安排试验并进行数据分析的一种方法，目的是用最少的测试用例达到最高的测试覆盖率

D. 等价类划分法根据软件的功能说明，对每一个输入条件确定若干个有效等价类和无效等价类，但只能为有效等价类设计测试用例

【参考答案】 D

【. 点拨】黑盒测试也称为功能测试，主要用于集成测试，确认测试和系统测试阶段。黑盒测试根据软件需求规格说明所规定的功能来设计测试用例，一般包括功能分解、等价类划分、边界值分析、判定表、因果图、状态图、随机测试、错误推测和正交试验法等。

在设计测试用例时，等价类划分是用得最多的一种黑盒测试方法。所谓等价类就是某个输入域的集合，对每一个输入条件确定若干个有效等价类和若干个无效等价类，分别设计覆盖有效等价类和无效等价类的测试用例。无效等价类是用来测试非正常的输入数据的，所以要为每个无效等价类设计一个测试用例。

边界值分析通过选择等价类边界作为测试用例，不仅重视输入条件边界，而且也必须考虑输出域边界。在实际测试工作中，将等价类划分法和边界值分析结合使用，能更有效地发现软件中的错误。

因果图方法是从用自然语言书写的程序规格说明的描述中找出因(输入条件)和果(输出或程序状态的改变)，可以通过因果图转换为判定表。

正交试验设计法，就是使用已经造好了的正交表格来安排试验并进行数据分析的一种方法，目的是用最少的测试用例达到最高的测试覆盖率。

【2022年下】在白盒测试中，测试强度最高的是()。

-
- A. 语句覆盖
 - B. 分支覆盖
 - C. 判定覆盖
 - D. 路径覆盖

【参考答案】 D

考点3: V 模型

【2015年下】()的目的是检查模块之间,以及模块和已集成的软件之间的接口关系,并验证已集成的软件是否符合设计要求。其测试的技术依据是()。

- A. 单元测试
 - B. 集成测试
 - C. 系统测试
 - D. 回归测试
- A. 软件详细设计说明书
 - B. 技术开发合同
 - C. 软件概要设计文档
 - D. 软件配置文档

【参考答案】 BC

考点4: 其他测试

【模拟题练习】以下关于测试的说法,正确的是()。

- A. Web 系统测试的测试内容与其它系统测试内容完全不同,其测试内容包括:链接测试、表格测试、图形测试等
- B. 链接测试需要保证 Web 应用系统上没有孤立的页面
- C. 回归测试一般用于集成测试阶段,它强调组装模块时应增量式组装,因为这样测试更全面
- D. AB 测试也叫金丝雀测试,主要看新版本的软件能否正常运行,如果能正常运行就逐步平移部署

【参考答案】 B

考点5: 集成测试策略

【2016年下】软件集成测试将已通过单元测试的模块集成在一起,主要测试模块之间的协作性。从组装策略而言,可以分为()。集成测试计划通常是在()阶段完成,集成测试一般采用黑盒测试方法。

- A. 批量式组装和增量式组装
 - B. 自顶向下和自底向上组装
 - C. 一次性组装和增量式组装
 - D. 整体性组装和混合式组装
- A. 软件方案建议
 - B. 软件概要设计
 - C. 软件详细设计
 - D. 软件模块集成

【参考答案】 CB

【点拨】集成测试可以分为一次性组装和增量式组装，增量式组装测试效果更好。集成测试计划一般在概要设计阶段完成。

考点6：系统测试

【2009年下】系统测试将软件、硬件、网络等其它因素结合，对整个软件进行测试。（）不是系统测试的内容。

- A. 路径测试
- B. 可靠性测试
- C. 安装测试
- D. 安全测试

【参考答案】 A

【点拨】系统测试是根据系统方案说明书来设计测试用例，常见的系统测试主要有恢复测试、安全性测试、压力测试、性能测试、可靠性测试、可用性测试、可维护性测试和安装测试。

【2019年下】软件性能测试有多种不同类型的测试方法，其中，（）用于测试在限定的系统下考查软件系统极限运行的情况，（）可用于测试系统同时处理的在线最大用户数量。

- | | | | |
|---------|---------|---------|---------|
| A. 强度测试 | B. 负载测试 | C. 压力测试 | D. 容量测试 |
| A. 强度测试 | B. 负载测试 | C. 压力测试 | D. 容量测试 |

【参考答案】 AD

【点拨】根据测试目的不同，性能测试主要包括压力测试、负载测试、并发测试和可靠性测试等。

强度测试：是在系统资源特别低的情况下考查软件系统极限运行情况。本题第一空选择A 选项。

负载测试：用于测试超负荷环境中程序是否能够承担，确定在各种工作负载下系统的性能，测试当负载逐渐增加时，系统各项性能指标的变化情况。

压力测试：通过确定系统的瓶颈或不能接收的性能点，来获得系统能够提供的最大服务级别的测试。负载测试和压力测试可以结合进行，统称为负载压力测试。

容量测试：并发测试也称为容量测试，主要用于测试系统可同时处理的在线最大用户数量。本题第二空选择D 选项。

考点7：软件测试与软件调试

【2025年上】以下关于软件测试与调试说法错误的是（）。

A. 测试是调试之后的活动，测试和调试在目标、方法和思路上都不同

B. 测试从一个已知的条件开始，使用预先定义的过程，有预知的结果；调试从一个未知的条件开始，结束的过程不可预计

C. 测试过程可以事先设计，进度可以事先确定；而调试不能描述过程或持续时间

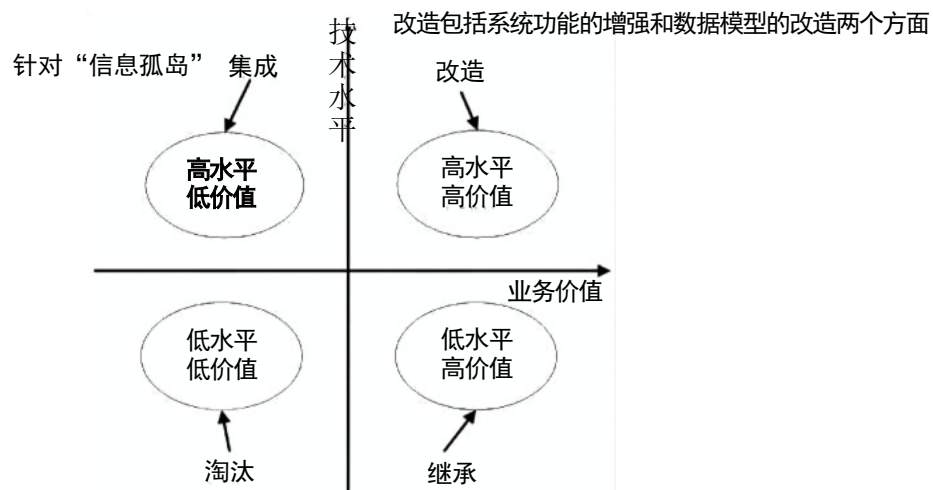
D. 测试的目的是找出程序中存在的错误，而调试的目的是定位错误并且修改程序以修正错误

【参考答案】 A

3. 10系统运行与软件维护

知识点1:系统转换计划

(1) 遗留系统演化策略



淘汰策略：遗留系统的技术含量较低，且具有较低的业务价值。对遗留系统的完全淘汰是企业资源的根本浪费，系统分析师应该善于“变废为宝”，通过对遗留系统功能的理解和借鉴，可以帮助新系统的设计，降低新系统开发的风险。

继承策略：遗留系统的技术含量较低，已经满足企业运作的功能或性能要求，且具有较高的商业价值，目前企业的业务尚紧密依赖该系统。对这种遗留系统的演化策略为继承。在开发新系统时，需要完全兼容遗留系统的功能模型和数据模型。为了保证业务的连续性，新老系统必须并行运行一段时间，再逐渐切换到新系统上运行。

改造策略：遗留系统具有较高的业务价值，基本上能够满足企业业务运作和决策支持的需要。这种系统可能建成的时间还很短，对这种遗留系统的演化策略为改造。改造包括系统功能的增强和数据模型的改造两个方面。系统功能的增强是指在原有系统的基础上增加新的应用要求，对遗留系统本身不做改变；数据模型的改造是指将遗留系统的旧的数据模型向新的数据模型的转化。

集成策略：遗留系统的技术含量较高，但其业务价值较低，可能只能完成某个部门(或子公司)的业务管理。这种系统在各自的局部领域里工作良好，但对于整个企业来说，存在多个这样的系统，不同的系统基于不同的平台、不同的数据模型，形成了一个信息孤岛，对这种遗留系统的演化策略为集成。

(2) 新旧系统转换策略



直接转换：直接转换是在原有系统停止运行的某一时刻，新系统立即投入运行，中间没有过渡阶段。采用这种方式时，人力和费用最省，适用于系统不太复杂或现有系统完全不能使用的场合。但是这种方式风险高。

并行转换：并行转换就是新系统和旧系统并行工作一段时间，经过这段时间的试运行后，再用新系统正式替换下现有系统。那么这种方式，它的好处就是风险很小。在转换期间还可以同时比较新旧两套系统的性能，而且能够让操作人员得到全面的培训，所以对于一些比较大的信息系统，或者处理过程比较复杂，数据比较重要的系统。并行转换是一种最常用的转换方式。那么这种转换方式也有缺点，缺点就在于两套系统并行期间。要有两组人员，或者两套处理方式同时并存，在人力和费用上消耗比较大，转换的周期比较长，而且难以控制新旧系统当中数据的变化。所以这就要求要做好转换计划，并且要加强管理。

分段转换：这是直接转换和并行转换的结合，也就是分期分批、逐步转换。一般比较大的系统采用这种方式比较合适，它能够保证软件平稳运行，费用也不太高，就是将大的系统分成多个子系统，成熟一个子系统就切换一个子系统，主要是分期分批。这种新旧转换，震动比较小，用户比较容易接受。但是由于采取的是渐进的方式，会导致新旧系统的转换周期比较长。

考点1: 遗留系统演化策略

【2022年下】在对遗留系统进行评估时，对于技术含量较高、业务价值较低且仅能完成某个部门的业务管理的遗留系统，一般采用的遗留系统演化策略是()策略。

A.淘汰

B. 继承

C. 集成

D. 改造

【参考答案】 C

【模拟练习题】遗留系统 (Legacy System) 的演化策略分为淘汰策略、继承策略、改造策略和集成策略。我们在对遗留系统进行评估时，对于技术含量较低，但目前企业的业务尚紧密依赖该系统，一般采用的遗留系统演化策略是 () 策略。

A. 改造

B. 继承

C. 集成

D. 淘汰

【参考答案】 B

考点2: 旧系统转换策略

【模拟练习题】某公司正在实施新旧信息系统的转换，若想把转换的风险降到最低，采用 () 方式最合适。

A. 并行转换

B. 分段转换

C. 分块转换

D. 直接转换

【参考答案】 A

【点拨】直接转换就是在确定系统试运行准确无误时，立刻启用新系统并终止旧系统的运行，该方式简单，但风险大；并行转换是新旧系统并行工作一段时间，并经过一段时间的考验以后，新系统正式替代旧系统，该方式安全，风险小，但实施费用高；分段转换实际上是以上两种转换方式的结合，是一种混合方式，风险适中；分块转换为迷惑选项，不存在此种转换方式。由以上分析知，并行转换的风险最小，故选A。

知识点2: 系统维护

(1) 影响软件可维护性的因素

【可理解性】 是指通过阅读源代码和相关文档，了解软件的功能和如何运行的容易程度。

【可修改性】 是指修改软件的难易程度。

【可测试性】 是指验证软件程序正确的难易程度。

可测试性好的软件，通常意味着软件设计简单，复杂性低。因为软件的复杂性越大，测试的难度也就越大。

【可靠性】 一个软件的可靠性越高，需要维护的概率就会越低。

【可移植性】 是指将软件从一个环境移植到新的环境下正确运行的难易程度。

软件运行环境的变化是软件维护的一种常见情形，可移植性好的软件会降低维护的概率。

(2) 软件维护类型

正确性维护【修BUG】： 识别和纠正软件错误/缺陷，测试不可能发现所有错误。

适应性维护【应变】： 指使应用软件适应环境变化【外部环境、数据环境】而进行的修改。

完善性维护【新需求】： 为扩充功能和改善性能而进行的修改。

预防性维护【针对未来】： 为了适应未来的软硬件环境的变化，应主动增加预防性的新的功能，以使用系统适应各类变化而不被淘汰。经典实例：**【专用】改【通用】**。

考点1: 影响软件可维护性的因素

【模拟题练习】 可维护性通常包括()，对()的修改不归结为软件的维护工作。

- | | |
|-------------|---------------------|
| A. 可用性和可理解性 | B. 可修改性、数据独立性和数据一致性 |
| C. 可测试性和稳定性 | D. 可理解性、可修改性和可测试性 |
| A. 文档 | B. 数据 |
| C. 需求分析 | D. 代码 |

【参考答案】 DC

【点拨】 软件的可维护性通常包括可理解性、可修改性和可测试性。

软件维护是软件生命周期的最后一个阶段，而与软件维护有关的绝大多数问题的根源在于计划阶段和开发阶段的工作缺陷，这就经常涉及软件中的代码、数据和文档的维护工作，而需求分析并不是软件的维护工作。

考点2: 软软件维护类型

【2011年下】 软件()是指改正产生于系统开发阶段而在系统测试阶段尚未发现的错误。

- A. 完善性维护
- B. 适应性维护

C. 正确性维护

D. 预防性维护

【参考答案】C

【. 点拨】软件维护的类型包括：改正性维护（正确性维护）、适应性维护、完善性维护、预防性维护。

改正性维护：在软件交付使用后，必然会有一部分隐藏的错误被带到运行阶段来。这些隐藏下来的错误在某些特定的使用环境下就会暴露出来。为了识别和纠正软件错误、改正软件性能上的缺陷、排除实施中的错误使用，应当进行的诊断和改正错误的过程，就叫做改正性维护。

适应性维护：随着计算机的飞速发展，外部环境（新的硬、软件配置）或数据环境（数据库、数据格式、数据输入/输出方式、数据存储介质）可能发生变化，为了使软件适应这种变化，而去修改软件的过程就叫做适应性维护。

完善性维护：在软件的使用过程中，用户往往会对软件提出新的功能与性能要求。为了满足这些要求，需要修改或再开发软件，以扩充软件功能、增强软件性能、改进加工效率、提高软件的可维护性。这种情况下进行的维护活动叫做完善性维护。

预防性维护：为了提高软件的可维护性、可靠性等而提出的一种维护类型，它为以后进一步改进软件打下良好基础。通常，预防性维护定义为：“把今天的方法学用于昨天的系统以满足明天的需要”。也就是说，采用先进的软件工程方法对需要维护的软件或软件中的某一部分（重新）进行设计、编制和测试。

3.11 系统工程建模

知识点1: 基于模型的系统工程(Model Based Systems Engineering, MBSE)

MBSE(Model Based Systems Engineering, 基于模型的**系统工程**)

【MBSE 的三大支柱】

【软件工程】

【建模语言】沟通标准化, MBSE的核心。典型代表: SysML, OPMo----->UML

【建模思路】类似于路线图, 如: INCONSE 面向对象系统工程法。-----> 面向对象开发

【建模工具】如: Agilian, Modelio。-----> Visio、Rose



MBSE 是建模方法的形式化应用, 以使建模方法支持系统需求、分析、设计、验证和确认等活动, 这些活动从概念性设计阶段开始, 持续贯穿到设计开发以及后来的所有生命周期阶段。

考点1: MBSE的三大支柱

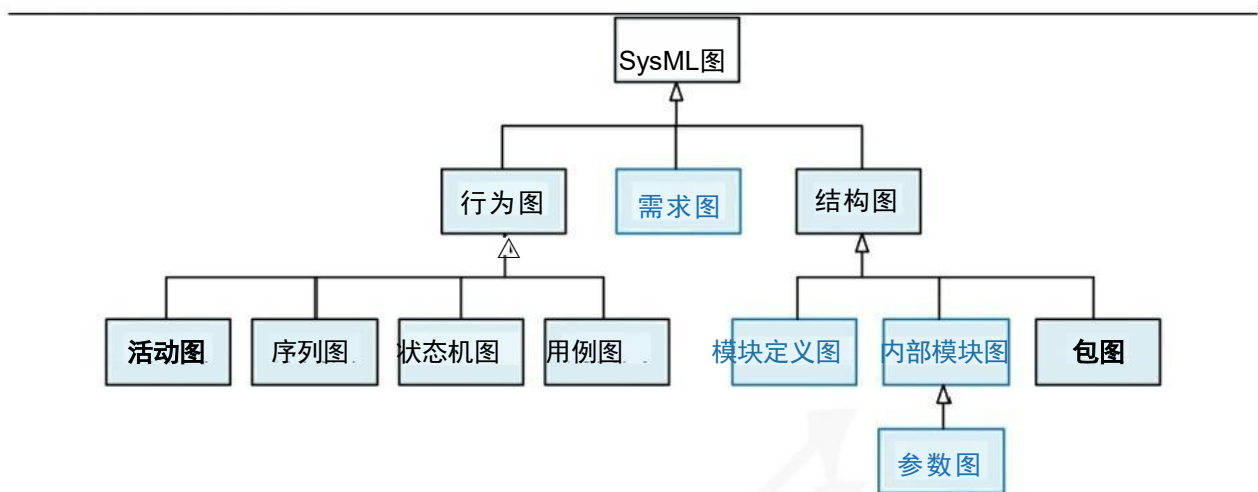
【模拟题练习】 MBSE 是建模方法的形式化应用, 下面()不属于 MBSE 的三大支柱。

- A. 建模语言
- B. 建模工具
- C. 建模分析
- D. 建模思路

【参考答案】C

知识点2: SysML【系统建模语言】

对象管理组织OMG 在对UML2.0 的子集进行重用和扩展的基础上, 提出了一种新的系统建模语言 (Systems Modeling Language, SysML), 作为系统工程的标准建模语言。 SysML 的目的是统一系统工程中使用的建模语言。



SysML 中的需求关系：**包含、跟踪、继承需求、改善、满足、验证和复制。**

包含 (Include): 需求可以且只能包含其他需求。

跟踪 (Trace): 对提供方元素(位于箭头端)的修改可能会导致对客户端元素(位于尾端)修改的需要。

继承 (deriveReq): 一个需求可以继承另一个需求的属性。

改善 (refine): 表示一个需求改进了另一个需求的满足程度。

满足 (satisfy): 一般是模块满足某种需求。

验证 (verify): 表示一个需求验证了另一个需求的正确性。

复制 (Copy): 表示一个需求复制了另一个需求的特性。

考点1: SysML 的需求关系

【模拟题练习】SysML(Systems Modeling Language) 是系统工程的标准建模语言，下面需求关系解释错误的是 ()

A. 跟踪 (Trace): 对提供方元素(位于箭头端)的修改可能会导致对客户端元素(位于尾端)修改的需要。

B. 验证 (verify): 表示一个需求验证了另一个需求的正确性。

C. 改善 (refine): 表示一个需求改进了另一个需求的满足程度。

D. 包含 (Include): 一般是模块满足某种需求。

【参考答案】D

第四章 软件架构设计

4.1 软件架构的概念

知识点1: 架构的概念

架构设计就是需求分配，即将满足需求的职责分配到组件上。

软件架构=软件体系结构

架构的本质：

软件架构为软件系统提供了一个结构、行为和属性的高级抽象。

软件架构风格是特定应用领域的惯用模式，架构定义一个词汇表和一组约束。

架构的作用：

软件架构是项目干系人进行交流的手段。

软件架构是可传递和可复用的模型，通过研究软件架构可能预测软件的质量。

软件架构使推理和控制的更改更加简单，有助于循序渐进的原型设计，可以作为培训的基础。

考点1：架构的概念

【2013年下】软件架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。架构风格定义了一类架构所共有的特征，主要包括架构定义、架构词汇表和架构（）。

A. 描述

B. 组织

C. 约束

D. 接口

【参考答案】 C

【点拨】软件架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。架构风格定义一个系统家族，即一个架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。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，并指导如何将各个模块和子系统有效地组织成一个完整的系统。对软件架构风格的研究和实践促进对设计的重用，一些经过实践证实的解决方案也可以可靠地用于解决新的问题。

【2014年下】以下关于软件架构设计重要性的描述，（）是错误的。

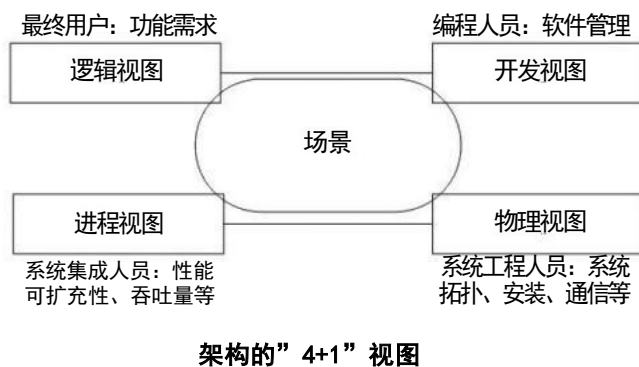
A. 软件架构设计能够满足系统的性能、安全性、可维护性等品质

- B. 软件架构设计能够帮助项目干系人 (Stakeholder) 更好地理解软件结构
- C. 软件架构设计能够帮助架构师更好地捕获和细化系统需求
- D. 软件架构设计能够有效地管理系统的复杂性，并降低系统维护费用

【参考答案】 C

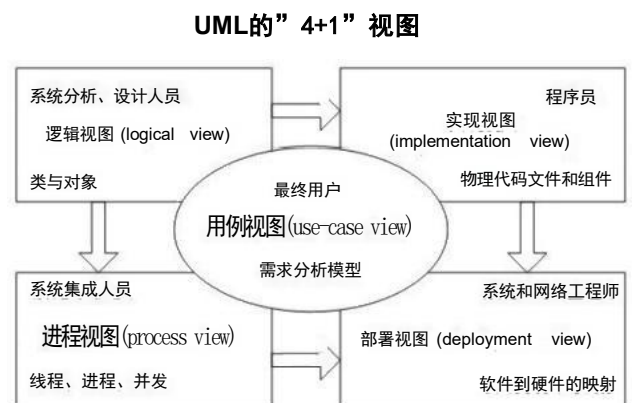
【. 点拨】软件架构设计不能捕获需求，软件架构设计是在需求捕获并进行分析之后开展的工作。

知识点2: 架构“4+1”视图



视角与视图

从不同的视角来检查，所以会有不同的视图。



考点1：“4+1”视图

【2011年下】1995年 Kruchten 提出了著名的“4+1”视图，用来描述软件系统的架构。在“4+1”视图中，（）用来描述设计的对象模型和对象之间的关系；（）描述了软件模块的组织与管理；（）描述设计的并发和同步特征。

- | | | | |
|---------|---------|---------|---------|
| A. 逻辑视图 | B. 用例视图 | C. 过程视图 | D. 开发视图 |
| A. 逻辑视图 | B. 用例视图 | C. 过程视图 | D. 开发视图 |
| A. 逻辑视图 | B. 用例视图 | C. 过程视图 | D. 开发视图 |

【参考答案】 ADC

【. 点拨】本题考查“4+1”视图。

“4+1”视图中的“4”，指的是：逻辑视图、开发视图、过程视图、物理视图，“1”指的是场景视图。

场景视图又称为用例视图，显示外部参与者观察到的系统功能。

逻辑视图从系统的静态结构和动态行为角度显示系统内部如何实现系统的功能。

开发视图又称为实现视图，显示的是源代码以及实际执行代码的组织结构。

处理视图又称为过程视图，显示程序执行时并发的状态。

物理视图展示软件到硬件的映射。

4. 2软件架构风格

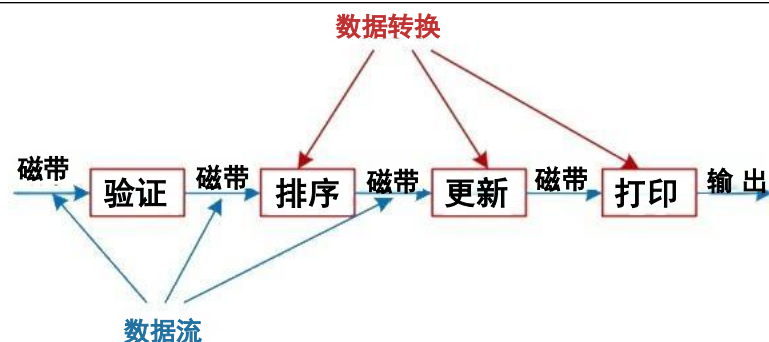
软件架构风格是描述某一特定应用领域中系统组织方式的惯用模式。架构风格定义一个系统家族，即一个架构定义一个词汇表和一组约束。词汇表中包含一些构件和连接件类型，而这组约束指出系统是如何将这些构件和连接件组合起来的。架构风格反映了领域中众多系统所共有的结构和语义特性，并指导如何将各个模块和子系统有效地组织成一个完整的系统。对软件架构风格的研究和实践促进对设计的重用，一些经过实践证实的解决方案也可以可靠地用于解决新的问题。

五大架构风格	子风格
数据流风格 【Data Flow】	批处理【Batch Sequential】、管道-过滤器【Pipes and Filters】
调用/返回风格 【Call/Return】	主程序/子程序【Main Program and Subroutine】 面向对象【Object-oriented】、分层架构【Layered System】
独立构件风格 【Independent Components】	进程通信【Communicating Processes】、 事件驱动系统(隐式调用)【Event system】
虚拟机风格 【Virtual Machine】	解释器【interpreter】、规则系统【Rule-based System】
以数据为中心 【Data-centered】	数据库系统【Database System】、黑板系统【Blackboard System】、 超文本系统【Hypertext System】

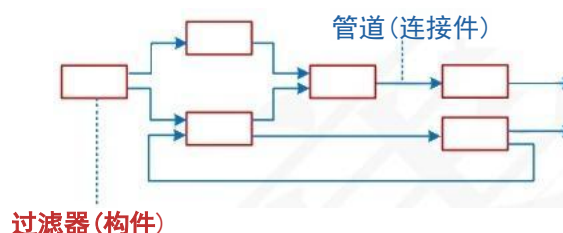
知识点1:数据流风格

(1) 批处理序列

构件为一系列固定顺序的计算单元，构件之间只通过数据传递交互。每个处理步骤是一个独立的程序，每一步必须在其前一步结束后才能开始，数据必须是完整的，以整体的方式传递。



(2) 管道-过滤器



每个构件都有一组输入和输出，构件读输入的数据流，经过内部处理，然后产生输出数据流。这个过程通常是通过对输入数据流的变换或计算来完成的，包括通过计算和增加信息以丰富数据、通过浓缩和删除以精简数据、通过改变记录方式以转化数据和递增地转化数据等。这里的构件称为过滤器，连接件就是数据流传输的管道，将一个过滤器的输出传到另一个过滤器的输入。

优点	缺点	典型实例
1、松耦合【高内聚-低耦合】； 2、良好的重用性/可维护性； 3、可扩展性【标准接口适配】； 4、良好的隐蔽性； 5、支持并行。	1、交互性较差； 2、复杂性较高； 3、性能较差(每个过滤器都需要解析与合成数据)。	传统编译器 网络报文处理

考点1: 数据流风格

【2016年下】某公司拟为某种新型可编程机器人开发相应的编译器。该编译过程包括词法分析、语法分析、语义分析和代码生成四个阶段，每个阶段产生的结果作为下一个阶段的输入，且需独立存储。针对上述描述，该集成开发环境应采用()架构风格最为合适。

- A. 管道-过滤器
- B. 数据仓储
- C. 主程序-子程序
- D. 解释器

【参考答案】 A

【点拨】“每个阶段产生的结果作为下一个阶段的输入”是典型的数据流架构风格的特点，选项中，仅有管道-过滤器属于这种风格。

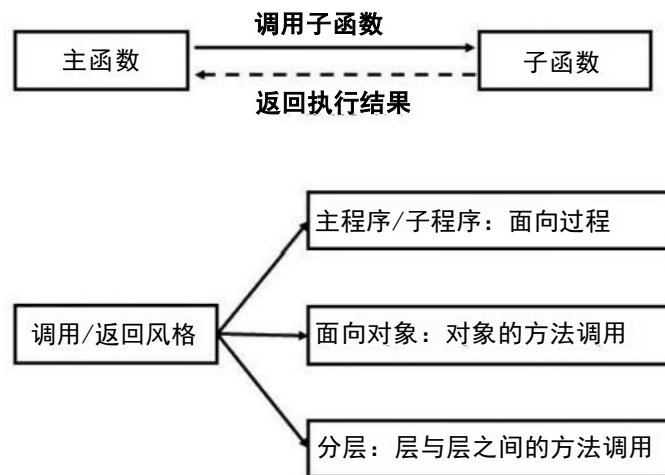
【模拟题练习】在架构风格中，（）和（）非常相似，两者共同点是：每一步处理都是独立的，并且每一步是顺序执行的，只有当前一步处理完后，后一步处理才能开始，数据传送在步与步之间作为一个整体。然而他们也有区别，前者是处理是全部的、高潜伏性的、输入时可随机存取、无合作性、无交互性的；而后者递增的、数据结果延迟小、输入时处理局部化、有反馈、可交互的。

A.管道一过滤器 B. 批处理 C. 微内核一扩展 D. 虚拟机

A.管道一过滤器 B. 批处理 C. 微内核一扩展 D. 虚拟机

【参考答案】 BA

知识点2: 调用/返回风格



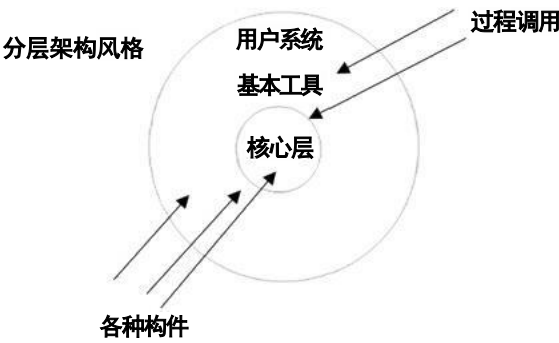
(1) 主程序/子程序

单线程控制，把问题划分为若干个处理步骤，构件即为主程序和子程序，子程序通常可合成为模块。过程调用作为交互机制，即充当连接件的角色。调用关系具有层次性，其语义逻辑表现为主程序的正确性取决于它调用的子程序的正确性。

(2) 面向对象

构件是对象，对象是抽象数据类型的实例。在抽象数据类型中，数据的表示和它们的相应操作被封装起来，对象的行为体现在其接受和请求的动作。连接件即是对象间交互的方式，对象是通过函数和过程的调用来交互的。

(3) 层次结构

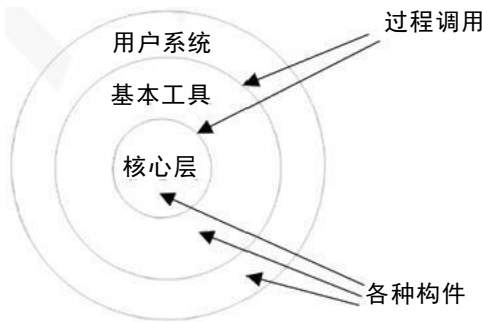


概念：构件组织成一个层次结构，连接件通过决定层间如何交互的协议来定义。每层为上一层提供服务，使用下一层的服务，只能见到与自己邻接的层。通过层次结构，可以将大的问题分解为若干个渐进的小问题逐步解决，可以隐藏问题的复杂度。修改某一层，最多影响其相邻的两层(通常只能影响上层)。

优点	缺点	特点
1、良好的重用性，只要接口不变可用在其它处； 2、可维护性好； 3、可扩展性好，支持递增设计。	1、并不是每个系统都方便分层； 2、很难找到一个合适的、正确的层次抽象方法； 3、不同层次之间耦合度高的系统很难实现。	各个层次的组件形成不同功能级别的虚拟机； 多层相互协同工作，而且实现透明。

考点1：调用/返回风格

【2019年下】对软件体系结构风格的研究和实践促进了对设计的复用。Garlan 和Shaw 对经典体系结构风格进行了分类。其中，()属于数据流体系结构风格；()属于虚拟机体系结构风格；而下图描述的属于()体系结构风格。



- A.面向对象

A. 面向对象

A. 层次型
- B.事件系统

B.事件系统

B. 事件系统
- C.规则系统

C.规则系统

C.规则系统
- D.批处理

D.批处理

D. 批处理

【参考答案】DCA

【. 点拨】关于架构的传统分类如下所示：

数据流风格：批处理序列、管道-过滤器，第一空选择 D 选项批处理。

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，第二空选择C 选项规则系统。

仓库风格：数据库系统、超文本系统、黑板系统。

题干给出的图示是典型的层次结构。第三空选择A 选项。

【2020年下】软件脆弱性是软件中存在的弱点(或缺陷),利用它可以危害系统安全策略,导致信息丢失、系统价值和可用性降低。嵌入式系统软件架构通常采用分层架构,它可以将问题分解为一系列相对独立的子问题,局部化在每一层中,从而有效地降低单个问题的规模和复杂性,实现复杂系统的分解。但是,分层架构仍然存在脆弱性。常见的分层架构的脆弱性包括()等两个方面。

- A. 底层发生错误会导致整个系统无法正常运行、层与层之间功能引用可能导致功能失效
- B. 底层发生错误会导致整个系统无法正常运行、层与层之间引入通信机制势必造成性能下降
- C. 上层发生错误会导致整个系统无法正常运行、层与层之间引入通信机制势必造成性能下降
- D. 上层发生错误会导致整个系统无法正常运行、层与层之间功能引用可能导致功能失效

【参考答案】B

【. 点拨】本题考查的是分层结构的特点。

首先根据分层的特点来看,分层架构是低耦合的,依赖关系非常简单,上层只能依赖于下层,没有循环依赖。所以底层错误将导致整个系统无法运行,而上层错误一般影响的是错误的这一部分,对整个系统的影响并不是完全的。所以C 选项和D 选项的描述是错误的。

其次,系统的风险可以看作是威胁利用了脆弱性而引起的。其中,威胁可以看成从系统外部对系统产生的作用而导致系统功能及目标受阻的现象。脆弱性可以看成是系统内部的薄弱点。脆弱性是客观存在的,但它本身没有实际伤害。B 选项“层与层之间引入通信机制势必造成性能下降”是客观存在的系统薄弱点,而A 选项的描述是一种可能性,并不是客观存在的,所以B 选项是系统脆弱性的体现。

【2015年下】以下关于软件架构风格与系统性能的关系叙述中,错误的是()。

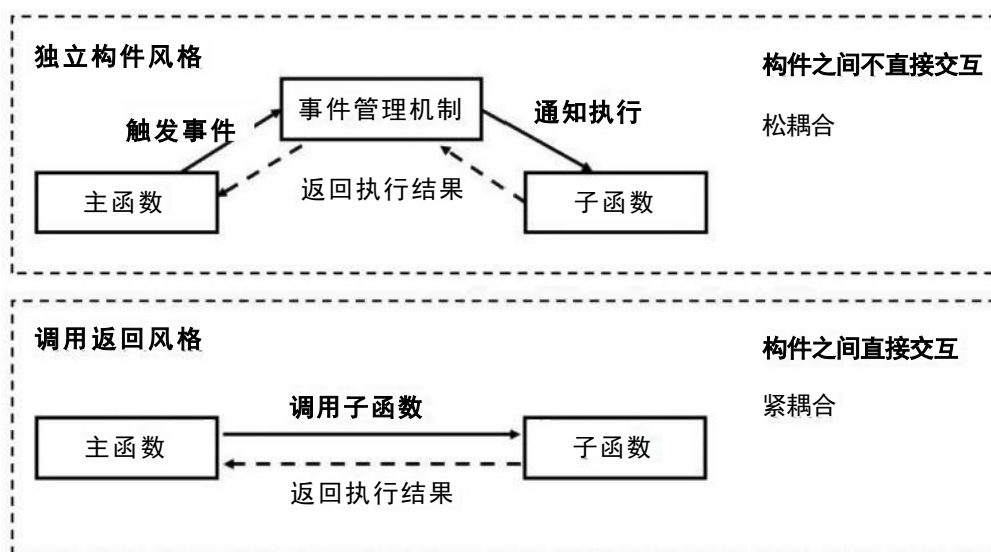
- A. 对于采用层次化架构风格的系统,划分的层次越多,系统的性能越差
- B. 对于采用隐式调用架构风格的系统,可以通过处理函数的并发调用提高系统处理性能
- C. 采用面向对象架构风格的系统,可以通过引入对象管理层提高系统性能

D. 对于采用解释器架构风格的系统，可以通过部分解释代码预先编译的方式提高系统性能

【参考答案】 C

【. 点拨】引入对象管理层不但不能提高性能，反而会降低系统性能。这个道理与分层模型中增加层次是一样的。

知识点3:独立构件风格



优点	缺点	特点
1、松耦合。 2、良好的重用性/可修改性/可扩展性。	1、构件放弃了对系统计算的控制。一个构件触发一个事件时，不能确定其它构件是否会响应它。而且即使它知道事件注册了哪些构件的过程，它也不能保证这些过程被调用的顺序。 2、数据交换的问题。 3、既然过程的语义必须依赖于被触发事件的上下文约束，关于正确性的推理就存在问题。	系统由若干子系统构成且成为一个整体；系统有统一的目标；子系统有主从之分；每一子系统有自己的事件收集和处理机制

(1) 进程通信

构件是独立的过程，连接件是消息传递。构件通常是命名过程，消息传递的方式可以是点对点、异步或同步方式，以及远程过程（方法）调用等。

(2) 事件驱动系统（隐式调用）

构件不直接调用一个过程，而是触发或广播一个或多个事件。构件中的过程在一个或多个事件中注册，当某个事件被触发时，系统自动调用在这个事件中注册的所有过程。一个事件的触发就导致了另一个模块中的过程调用。这种风格中的构件是匿名的过程，它们之间交互的连接件往往是以过程之间的隐式调用来实现的。主要优点是软件复用提供了强大的支持，为构件的维护和演化带来了方便；其缺点是构件放弃了对系统计算的控制。

考点1：独立构件风格

【2017年下】某公司拟开发一个新闻系统，该系统可根据用户的注册兴趣，向用户推送其感兴趣的新闻内容，该系统应该采用（）架构风格最为合适。

- A. 事件驱动系统
- B. 主程序-子程序
- C. 黑板
- D. 管道-过滤器

【参考答案】 A

【点拨】根据题目的意思，用户会注册自己的兴趣，然后系统也会把新闻按兴趣分类，如果某个新闻事件发生，可以通过事件来触发推送动作，将新闻推送给对其感兴趣的用户。这是典型的事件驱动系统应用场景。

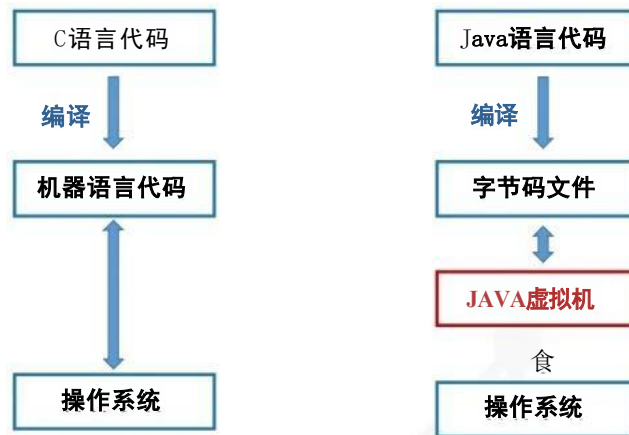
【2009年下】某公司欲开发一个基于图形用户界面的集成调试器。该调试器的编辑器和变量监视器可以设置调试断点。当调试器在断点处暂停运行时，编辑程序可以自动卷屏到断点，变量监视器刷新变量数值。针对这样的功能描述，采用（）的架构风格最为合适。

- A. 数据共享
- B. 虚拟机
- C. 隐式调用
- D. 显式调用

【参考答案】 C

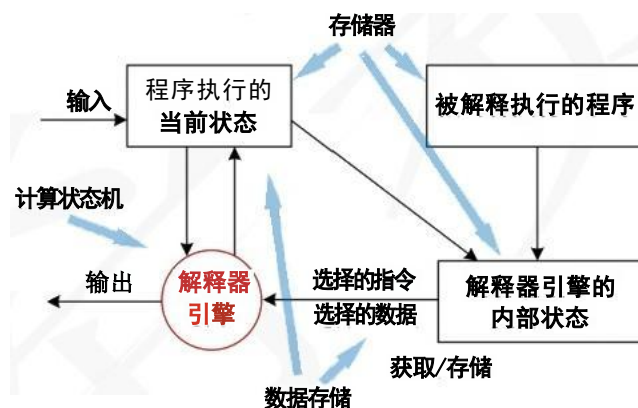
【点拨】根据题干描述，调试器在设置断点时，其本质是在断点处设置一个事件监听函数，当程序执行到断点位置时，会触发并调用该事件监听函数，监听函数负责进行自动卷屏、刷新变量数值等动作。这是一个典型的回调机制，属于隐式调用的架构风格。

知识点4: 虚拟机风格



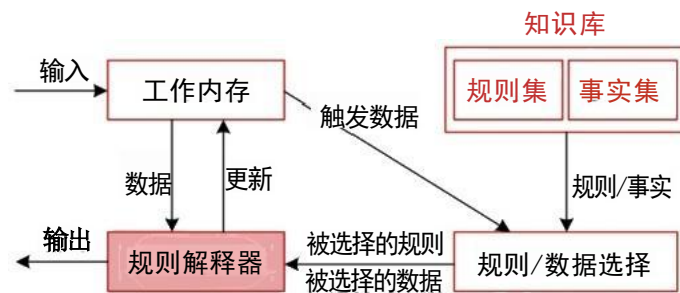
(1) 解释器

解释器通常包括一个完成解释工作的解释引擎、一个包含将被解释的代码的存储区、一个记录解释引擎当前工作状态的数据结构，以及一个记录源代码被解释执行的进度的数据结构。具有解释器风格的软件中含有一个虚拟机，可以仿真硬件的执行过程和一些关键应用，其缺点是执行效率比较低。



(2) 基于规则的系统

基于规则的系统包括规则集、规则解释器、规则/数据选择器和工作内存，一般用在人工智能领域和 DSS 中。



子分类	优点	缺点	特点	适合领域
解释器	可以灵活应对	复杂度		适用于需要“自定义规则”的场合
规则为中心	自定义场景	较高	在解释器的基础上增加经验规则	适用于专家系统

考点1: 解释器风格

【2010年下】某游戏公司欲开发一个大型多人即时战略游戏，游戏设计的目标之一是能够支持玩家自行创建战役地图，定义游戏对象的行为和对象之间的关系。针对该目标，公司应该采用()架构风格最为合适。

- A. 管道-过滤器
- B. 隐式调用
- C. 主程序-子程序
- D. 解释器

【参考答案】D

【点拨】本题主要考查软件架构设计策略与架构风格问题。根据题干描述，该软件系统特别强调用户定义系统中对象的关系和行为这一特性，这需要在软件架构层面提供一种运行时的系统行为定义与改变的能力，根据常见架构风格的特点和适用环境，可以知道最合适的架构设计风格应该是解释器风格。

【2011年下】某企业内部现有的主要业务功能已经封装为Web 服务。为了拓展业务范围，需要将现有的业务功能进行多种组合，形成新的业务功能。针对业务灵活组合这一要求，采用()架构风格最为合适。

- A. 管道-过滤器
- B. 解释器
- C. 显式调用
- D. 黑板

【参考答案】 B

【. 点拨】根据题干描述，需要将现有的业务功能进行多种组合，形成新的业务功能。而解释器在程序语言定义的计算和有效硬件操作确定的计算之间建立对应的联系。通过一个解析引擎进行解释和执行。所以可以用到解释。

考点2: 规则系统

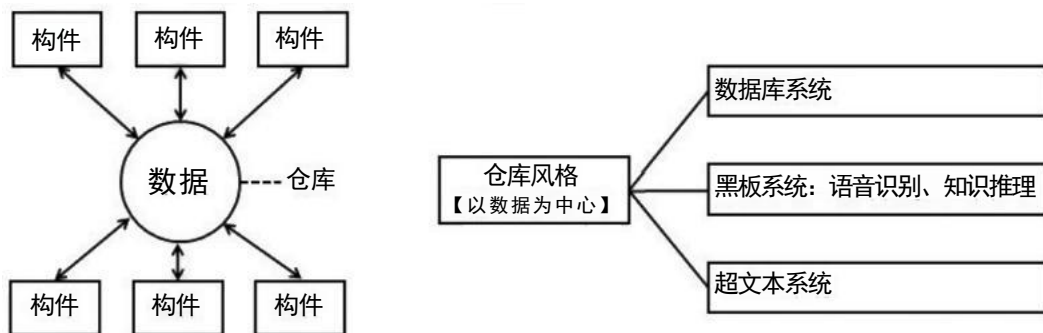
【2015年下】某公司拟开发一个地面清洁机器人。机器人的控制者首先定义清洁任务和任务之间的关系，机器人接受任务后，需要响应外界环境中触发的一些突发事件，根据自身状态进行动态调整，最终自动完成任务。针对上述需求，该机器人应该采用()架构风格最为合适。

- A. 面向对象
- B. 主程序-子程序
- C. 规则系统
- D. 管道-过滤器

【参考答案】 C

【. 点拨】规则系统属于虚拟机风格的一种，在本题中要求机器人的控制者首先定义清洁任务和任务之间的关系，然后由机器人执行，这说明机器人能对自定义的一些逻辑进行解析，这是虚拟机风格的一大特色。

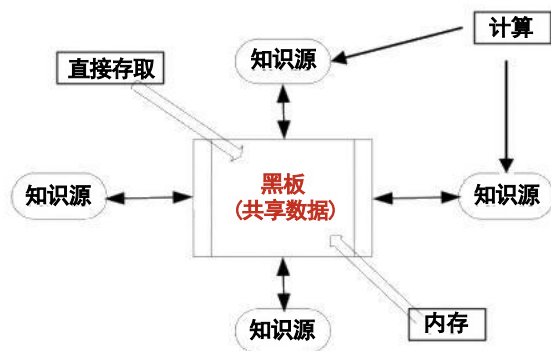
知识点5: 仓库风格(以数据为中心的风格)



(1) 数据库系统

构件主要有两大类，一类是中央共享数据源，保存当前系统的数据状态；另一类是多个独立处理单元，处理单元对数据元素进行操作。

(2) 黑板系统



包括知识源、黑板和控制三部分。知识源包括若干独立计算的不同单元，提供解决问题的知识。知识源响应黑板的变化，也只修改黑板；黑板是一个全局数据库，包含问题域解空间的全部状态，是知识源相互作用的唯一媒介；知识源响应是通过黑板状态的变化来控制的。黑板系统通常应用在对于解决问题没有确定性算法的软件中（信号处理、问题规划和编译器优化等）。

(3) 超文本系统

构件以网状链接方式相互连接，用户可以在构件之间进行按照人类的联想思维方式任意跳转到相关构件。超文本是一种非线性的网状信息组织方法，它以结点为基本单位，链作为结点之间的联想式关联。超文本系统通常应用在互联网领域。

现代集成编译环境一般采用这种架构风格。

架构风格	子分类	优点	缺点	特点	典型实例
仓库风格	数据库系统			以数据为中心	
	黑板系统	可更改性和可维护性； 可重用的知识源；容错性和健壮性	测试困难；不能保证有好的解决方案；难以建立好的控制策略；低效；开发困难； 缺少并行机制	在以数据为中心的基础上，使用中心数据触发业务逻辑部件	语音识别 模式识别 图像处理 知识推理

考点1: 数据库系统

【2018年下】在仓库风格中，有两种不同的构件，其中，（）说明当前状态，（）在中央数据存储上执行。

- | | | | |
|--------|----------|-------|--------|
| A.注册表 | B.中央数据结构 | C.事件 | D.数据库 |
| A.独立构件 | B.数据结构 | C.知识源 | D.共享数据 |

【参考答案】 BA

考点2: 黑板系统

【2015年下】某公司拟开发一个语音识别系统，其语音识别的主要过程包括分割原始语音信号、识别音素、产生候选词、判定语法片段、提供语义解释等，每个过程都需要进行基于先验知识的条件判断并进行相应的识别动作。针对该系统的特点，采用()架构风格最为合适。

- A. 解释器
- B. 面向对象
- C. 黑板
- D. 隐式调用

【参考答案】 C

【点拨】语音识别的处理是黑板风格的经典应用实例。

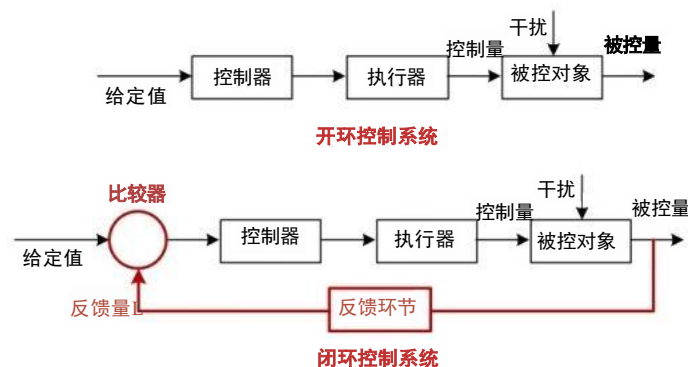
考点3: 超文本系统

【模拟题练习】超文本系统中出现的构件以网状链接方式相互连接，用户可以在构件之间进行按照人类的联想思维方式任意跳转到相关构件。超文本是一种非线性的网状信息组织方法，它以结点为基本单位，链作为结点之间的联想式关联。超文本系统通常应用在互联网领域。该风格属于()架构风格。

- A. 数据流风格
- B. 调用/返回风格
- C. 仓库风格
- D. 独立构件风格

【参考答案】 C

知识点6: 闭环控制架构(过程控制)



当软件被用来操作一个物理系统时，软件与硬件之间可以粗略地表示为一个反馈循环，这个反馈循环通过接受一定的输入，确定一系列的输出，最终使环境达到一个新的状态。适合于嵌入式系统，涉及连续的动作与状态。

适合于嵌入式系统，用于解决简单闭环控制问题。

经典应用：空调温控，定速巡航。

考点1: 过程控制

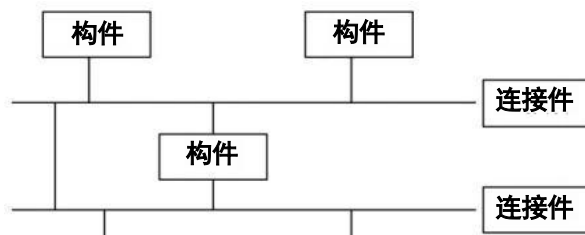
【2015年下】某公司拟开发一个轿车巡航定速系统，系统需要持续测量车辆当前的实时速度，并根据设定的期望速度自动控制轿车的油门和刹车。针对上述需求，采用()架构风格最为合适。

- A.解释器
- B.过程控制
- C.分层
- D.管道-过滤器

【参考答案】B

【点拨】过程控制又称闭环风格，该风格的最大的特点是设定参数，并不断测量现有的实际数据，将实际值与设定值进行比较，以确定接下来的操作。在本题中，定速巡航的场景正好符合这个模式。

知识点7: C2 风格



C2 架构的基本规则：

构件和连接件都有一个顶部和一个底部。

构件的顶部要连接到连接件的底部，构件的底部要连接到连接件的顶部，构件之间不允许直连。

一个连接件可以和任意数目的其它构件和连接件连接。

当两个连接件进行直接连接时，必须由其中一个的底部到另一个的顶部。

考点1：C2 风格

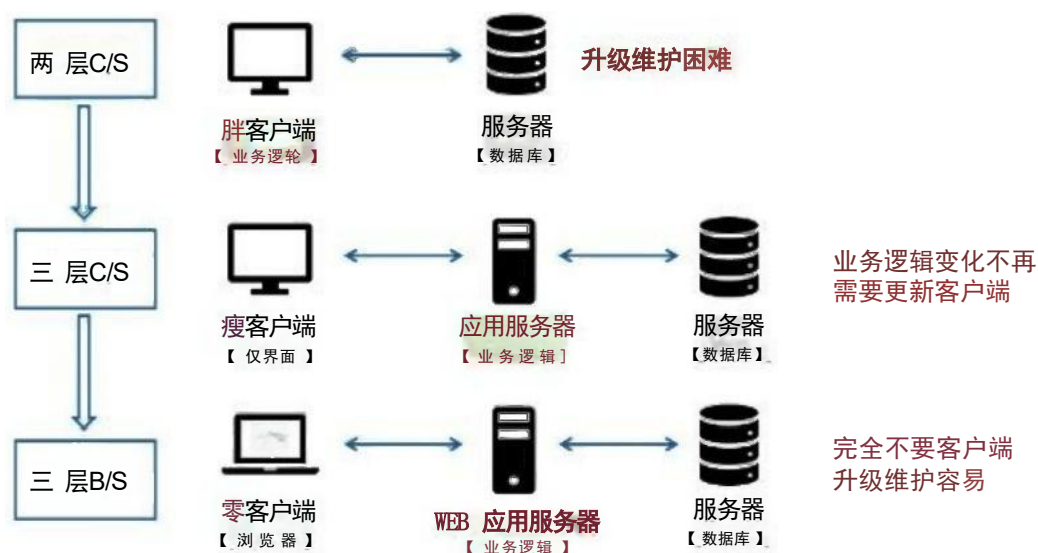
【2017年下】系统中的构件和连接件都有一个顶部和一个底部，构件的顶部应连接到某连接件的底部，构件的底部则应连接到某连接件的顶部，构件和构件之间不允许直接连接，连接件直接连接时，必须由其中一个的底部连接到另一个的顶部。上述构件和连接件的组织规则描述的是（）架构风格。

- A.管道-过滤器
- B. 分层系统
- C.C2
- D. 面向对象

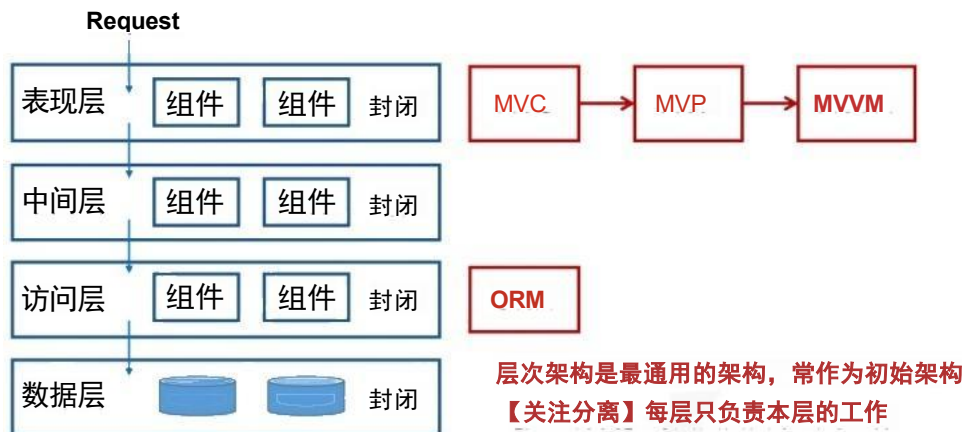
【参考答案】 C

4.3典型架构应用

知识点1: 层次架构



层次架构是最通用的架构，常作为初始架构，【关注分离】每层只负责本层的工作。



(1)MVC

Model (模型) 是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。

View (视图) 是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的。

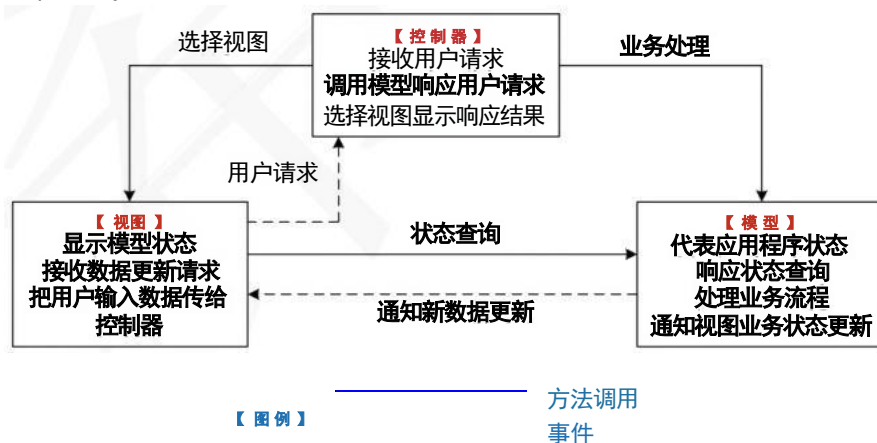
Controller (控制器) 是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

J2EE 体系结构中：

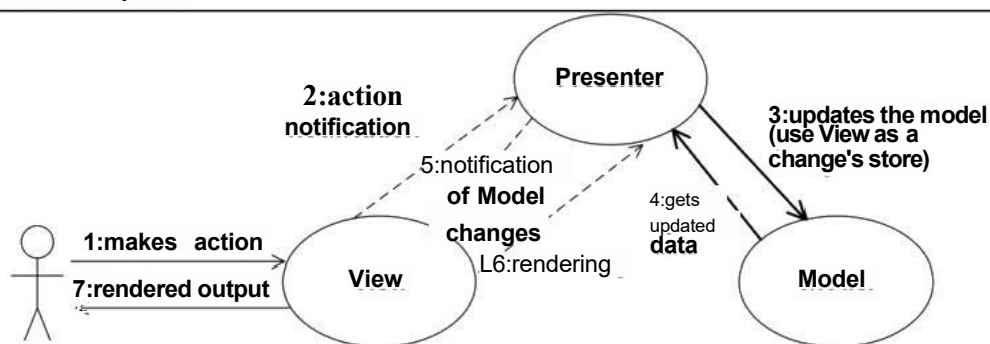
视图 (View):JSP

控制 (Controller):Servlet

模型 (Model):Entity Bean、Session Bean



(2)MVP



MVP与MVC关系：**MVP是MVC的变种。**

MVP 的优点：

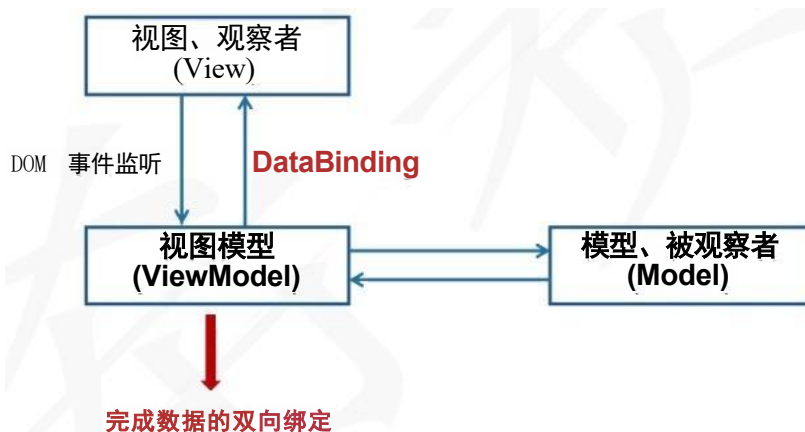
模型与视图完全分离，我们可以修改视图而不影响模型。

可以更高效地使用模型，因为所有的交互都发生在一个地方——Presenter内部。

可以将一个Presenter用于多个视图，而不需要改变Presenter的逻辑。

如果我们把逻辑放在Presenter中，那么我们就可以脱离用户接口来测试这些逻辑(单元测试)。

(3)MVVM



考点1: C/S 架构

【2020年下】针对二层C/S软件架构的缺点，三层C/S架构应运而生。在三层C/S架构中，增加了一个（）。三层C/S架构是将应用功能分成表示层、功能层和（）三个部分。其中（）是应用的用户接口部分，担负与应用逻辑间的对话功能。

- | | | | |
|----------|-----------|---------|--------|
| A. 应用服务器 | B. 分布式数据库 | C. 内容分发 | D. 镜像 |
| A. 硬件层 | B. 数据层 | C. 设备层 | D. 通信层 |
| A. 表示层 | B. 数据层 | C. 应用层 | D. 功能层 |

【参考答案】 ABA

【. 点拨】C/S 架构是基于资源不对等，且为实现共享而提出来的，是20世纪90年代成熟起来的技术，C/S 结构将应用一分为二，服务器(后台)负责数据管理，客户机(前台)完成与用户的交互任务。

但是二层C/S 存在很多缺点，因此，三层 C/S 结构应运而生。三层 C/S 结构是将应用功能分成表示层、功能层和数据层三个部分。

表示层是应用的用户接口部分，它担负着用户与应用间的对话功能。它用于检查用户从键盘等输入的数据，并显示应用输出的数据。在变更用户接口时，只需改写显示控制和数据检查程序，而不影响其他两层。检查的内容也只限于数据的形式和取值的范围，不包括有关业务本身的处理逻辑。

功能层相当于应用的本体，它是将具体的业务处理逻辑编入程序中。而处理所需的数据则要从表示层或数据层取得。表示层和功能层之间的数据交往要尽可能简洁。

数据层就是数据库管理系统，负责管理对数据库数据的读写。数据库管理系统必须能迅速执行大量数据的更新和检索。因此，一般从功能层传送到数据层的要求大都使用 SQL 语言。

考点2: MVC 架构

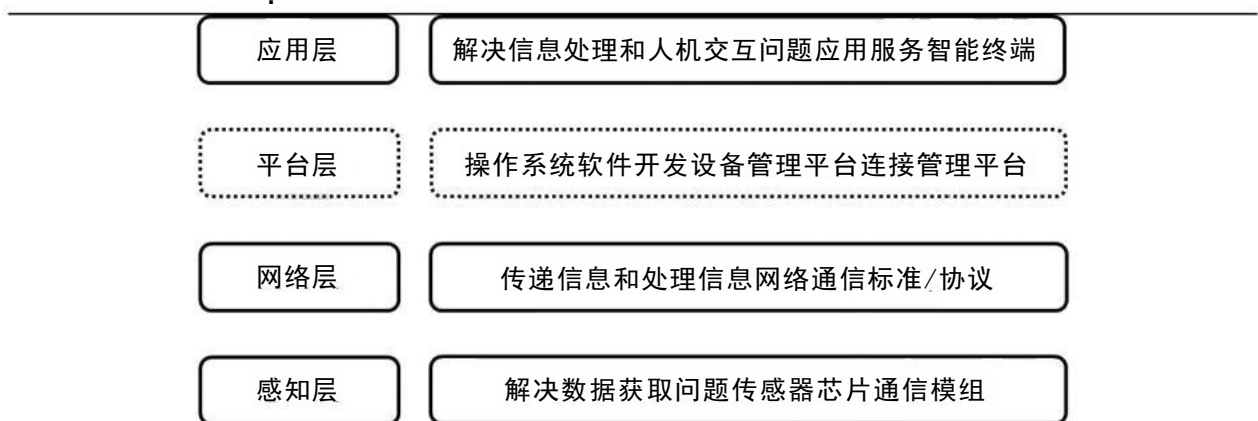
【2009年下】在一个典型的基于 MVC(Model-View-Controller) 的 J2EE 应用中，分发客户请求、有效组织其它构件为客户端提供服务的控制器由()实现。

- A.Entity Bean
- B.Session Bean
- C.Servlet
- D.JSP

【参考答案】C

【. 点拨】在一个典型的基于 MVC(Model View Controller) 的 J2EE 应用中，系统的界面由 JSP 构件实现，分发客户请求、有效组织其它构件为客户端提供服务的控件器由Servlet 构件实现，数据库相关操作由 Entity Bean 构件实现，系统核心业务逻辑由Session Bean 构件实现。

知识点2: 物联网分层结构



考点1: 物联网分层结构

【2021年下】软件架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，按照软件架构风格，物联网系统属于（）软件架构风格。

- A. 层次型
- B. 事件系统
- C. 数据流
- D. C2

【参考答案】A

【点拨】由于物联网从架构角度来看，是分三层的：

感知层：识别物体、采集信息。如：二维码、RFID、摄像头、传感器（温度、湿度）。

网络层：传递信息和处理信息。通信网与互联网的融合网络、网络管理中心、信息中心和智能处理中心等。

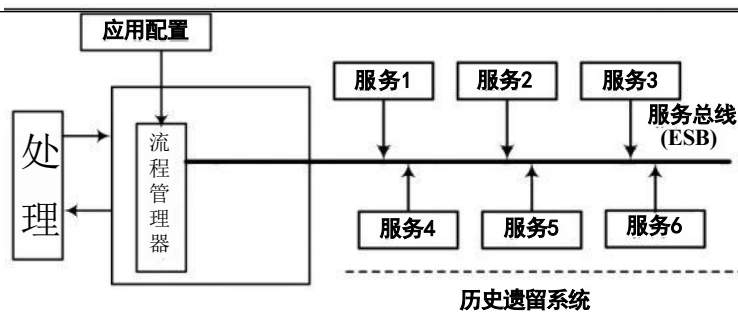
应用层：解决信息处理和人机交互的问题。

所以应属于层次型架构风格。

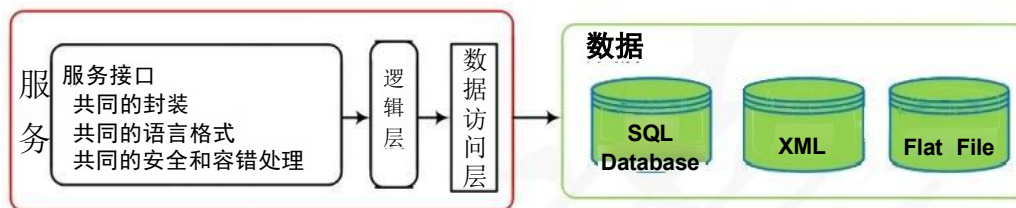
知识点3: 基于服务的架构 (SOA)

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。

(1) 典型的SOA 架构



(2) 单个服务的内部结构



(3) SOA 层次和特点



服务是标准化程度更高的构件。

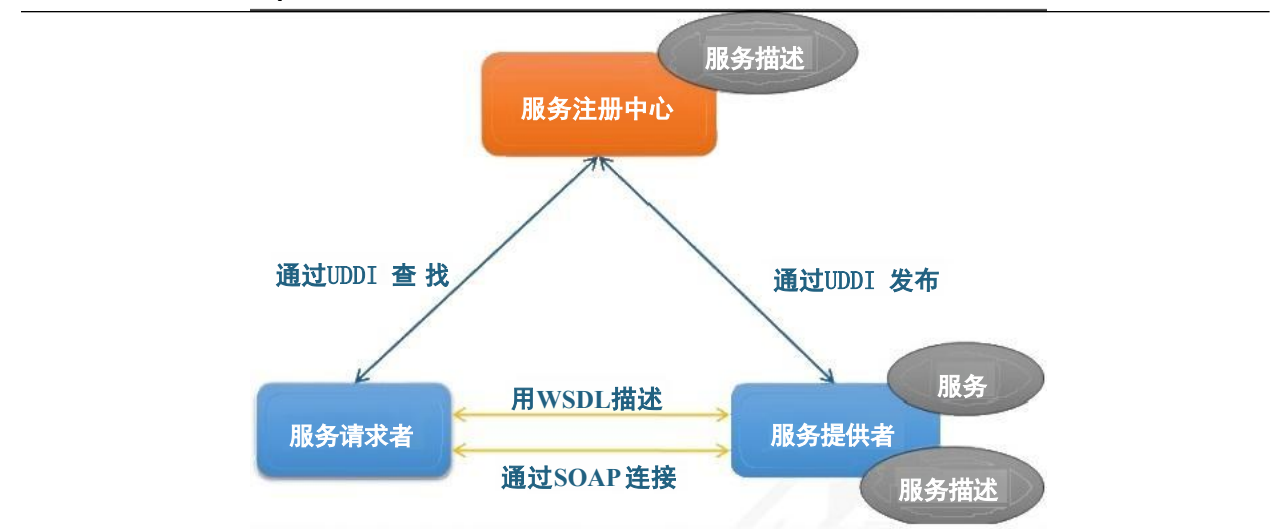
服务构件粗粒度，传统构件细粒度居多(粗粒度)。

服务构件的接口是标准的，主要是WSDL 接口，传统构件常以具体API形式出现(标准化结构)。

服务构件的实现与语言无关，传统构件绑定某种特定语言。

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

(4) 关键技术



功能	协议
发现服务	UDDI、DISCO
描述服务	WSDL、XML Schema
消息格式层	SOAP、REST
编码格式层	XML (DOM, SAX)
传输协议层	HTTP、TCP/IP、SMTP等

其中：

【UDDI】 是 Web 服务集成的一个体系框架，包含了服务描述与发现的标准规范。

【WSDL】 服务描述语言，三个基本属性（服务做什么/如何访问服务/服务位于何处。）

【SOAP】 基于XML 的协议，在分布式环境中交换信息。4个部分： SOAP 封装（消息内容，谁发的，谁接收）、SOAP 编码规则（数据类型实例）、 SOAP RPC（远程过程调用和应答协定）、 SOAP 绑定（使用底层协议交换信息）。

REST: 表述性状态转移 (Representational State Transfer, REST) 是一种只使用HTTP 和 XML 进行基于Web 通信的技术，可以降低开发的复杂性，提高系统的可伸缩性。

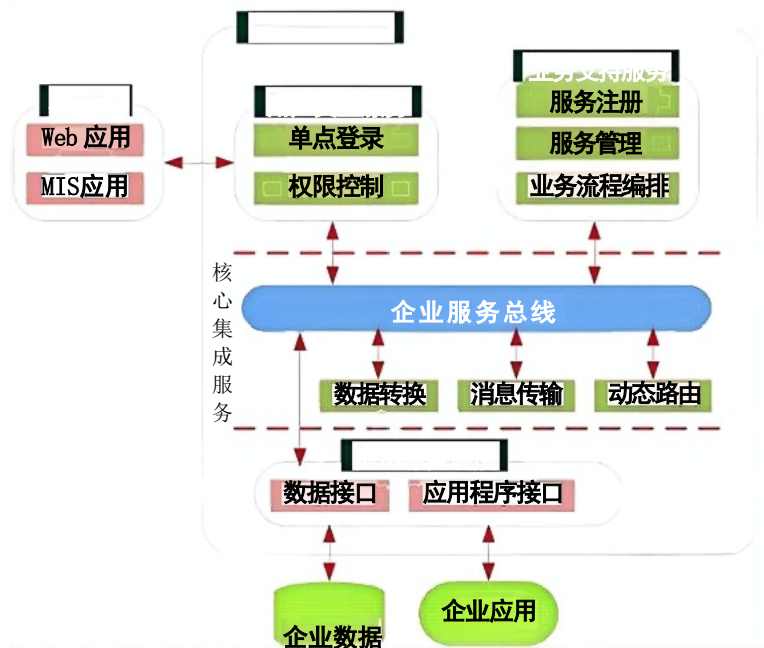
REST 的5个原则：

- 1) 网络上的所有事物都被抽象为资源。
- 2) 每个资源对应一个唯一的资源标识。
- 3) 通过通用的连接件接口对资源进行操作。
- 4) 对资源的各种操作不会改变资源标识。
- 5) 所有的操作都是无状态的。

REST API

命令	用途
GET	从指定的资源请求数据，只进行数据检索，不进行其他操作
POST	将数据发送到服务器进行创建，通常用于上传文件或提交表单
PUT	更新目标资源的所有当前表示，使用上传的内容进行替换
DELETE	删除指定的资源
HEAD	与GET方法类似，但只传输状态行和头部信息
PATCH	对资源进行部分修改

ESB：消息中间件、服务总线，它是面向服务架构的一种实现方式。



【服务请求者与服务提供者之间解耦】

- 提供位置透明性的消息路由和寻址服务；
- 提供服务注册和命名的管理功能；
- 支持多种的消息传递范型；
- 支持多种可以广泛使用的传输协议；
- 支持多种数据格式及其相互转换；
- 提供日志和监控功能。

考点1：基于服务的架构（SOA） 关键技术

（ ）是由中间件技术实现并支持 SOA 的一组基础架构，它提供了一种基础设施，其优势在于（ ）。

A.ESB

B.微服务

C.云计算

D.Multi-Agent System

A.支持了服务请求者与服务提供者之间的直接链接

B.支持了服务请求者与服务提供者之间的紧密耦合

C.消除了服务请求者与服务提供者之间的直接链接

D.消除了服务请求者与服务提供者之间的关系

【参考答案】AC

【. 点拨】企业服务总线 (Enterprise Service Bus,ESB) 技术，其思想是提供一种标准的软件底层架构，各种程序组件能够以服务单元的方式“插入”到该平台上运行，并且组件之间能够以标准的信息通信方式来进行交互。

ESB 的定义通常如下：企业服务总线是由中间件技术实现的支持面向服务架构的基础软件平台，支持异构环境中的服务以及基于消息和事件驱动模式的交互，并且具有适当的服务质量和可管理性。

这种交互过程不再是点对点的直接交互模式，而是由事件驱动的消息交互模式。通过这种方式，ESB 最大限度上解耦了组件之间的依赖关系，降低了软件系统互连的复杂性。 ESB 不支持服务请求者与服务提供者之间的直接链接，二者之间仍然存在关系，只是这种关系是松耦合的。本题第二空选择C选项，其他选项的说法都是错误的。

【2024年下】以下关于 SOAP 协议说法错误的是（）。

A. SOAP RPC表示远程过程调用和应答的协定

B.封装和编码规则被定义在相同的XML 命名空间中，这样使得定义更加简单

C. SOAP 封装定义了一个框架，描述消息中的内容是什么, 是谁发送的，谁应当接受并处理它们，以及如何处理它们

D. SOAP 绑定定义了SOAP 使用哪种协议交换信息

【参考答案】B

【. 点拨】 SOAP 是在分散或分布式的环境中交换信息的简单的协议，是一个基于XML 的协议。它包括4个部分： SOAP 封装 (Envelop), 定义了一个描述消息中的内容是什么, 是谁发送的, 谁应当接收并处理它以及如何处理它们的框架； SOAP 编码规则 (Encoding Rules), 用于表示应用程序需要使用的数据类型的实例； SOAP RPC 表示 (RPC Representation) 是远程过程调用和应答的协定； SOAP 绑定 (Binding) 是使用底层协议交换信息。虽然这4个部分都作为SOAP 的一部分, 作为一个整体定义的, 但它们在功能上是相交的、彼此独立的。特别地, 信封和编码规则是被定义在不同的XML 命名空间 (Namespace) 中, 这样使得定义更加简单。 SOAP 的两个主要设计目标是简单性和可扩展性, 这就意味着有一些传统消息系统或分布式对象系统中的某些性质将不是 SOAP 规范的一部分。例如, 分布式垃圾收集 (Distributed Garbage Collection)、成批传送消息、对象引用和对象激活等。

【2024年下】 Web 服务描述语言WSDL 描述了Web 服务的三个基本属性, 包括 ()。

- A. 谁要访问服务、如何访问服务和服务位于何处
- B. 服务做什么、谁要访问服务和服务位于何处
- C. 服务做什么、如何访问服务和谁要访问服务
- D. 服务做什么、如何访问服务和服务位于何处

【参考答案】 D

【2025上】在 REST API 中, () 用于对一个资源进行部分修改, 而不需要发送整个资源的完整表示。

- A. PART
- B. POST
- C. PUT
- D. PATCH

【参考答案】 D

知识点4: 微服务

(1) 什么是微服务

微服务顾名思义, 就是很小的服务, 所以它属于面向服务架构的一种。

(2) 微服务的优势

优点	解读
【复杂应用解耦】	小服务(且专注于做一件事情) 化整为零，易于小团队开发
【独立】	独立开发 独立测试及独立部署(简单部署) 独立运行(每个服务运行在其独立进程中)
【技术选型灵活】	支持异构(如：每个服务使用不同数据库)
【容错】	故障被隔离在单个服务中，通过重试、平稳退化等机制实现应用层容错
【松耦合，易扩展】	可根据需求独立扩展

(3) 微服务面临的挑战

分布式环境下的数据一致性【更复杂】

测试的复杂性【服务间依赖测试】

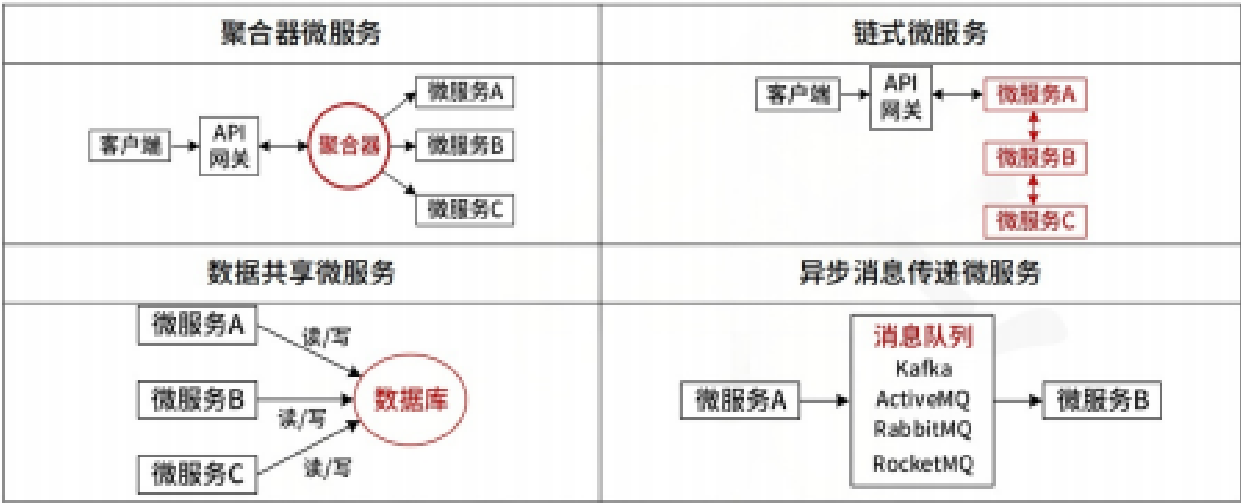
运维的复杂性

(4) 微服务与 SOA 的对比

微服务	SOA
能拆分的就拆分	是整体的，服务能放一起的都放一起
纵向业务划分	是水平分多层
由单一组织负责	按层级划分不同部门的组织负责
细粒度	粗粒度
两句话可以解释明白	几百字只相当于SOA的目录
独立的子公司	类似大公司里面划分了一些业务单元(BU)
组件小	存在较复杂的组件
业务逻辑存在于每一个服务中	业务逻辑横跨多个业务领域
使用轻量级的通信方式，如HTTP	企业服务总线(ESB)充当了服务之间通信的角色

微服务架构实现	SOA实现
团队级，自底向上开展实施	企业级，自顶向下开展实施
一个系统被拆分成多个服务，粒度细	服务由多个子系统组成，粒度大
无集中式总线，松散的服务架构	企业服务总线，集中式的服务架构
集成方式简单(HTTP/REST/JSON)	集成方式复杂(ESB/WS/SOAP)
服务能独立部署	单块架构系统，相互依赖，部署复杂

(5) 微服务架构模式方案



(6) 微服务设计约束

1、微服务个体约束

【每个微服务都是独立的，修改一个微服务不能影响另一个微服务】

2、微服务与微服务之间的横向关系

【通过第三方服务注册中心来满足服务的可发现性】

3、微服务与数据层之间的纵向约束

【数据是微服务的“私产”，访问时需要通过微服务】

4、全局视角下的微服务分布式约束

【高效运维整个系统】

考点1：微服务架构

【2025上】微服务架构中，断路器模式主要包含以下三种状态()。

- A.关闭状态、激活状态、挂起状态
- B.激活状态、打开状态、休眠状态
- C.激活状态、打开状态、熔断状态

D. 关闭状态、打开状态、半开状态

【参考答案】 D

【点拨】断路器模式包含以下三个主要状态：

关闭状态(Closed)： 初始状态，允许请求通过。如果请求失败次数超过阈值，则状态切换为“打开状态”。

打开状态(Open)： 不允许请求通过，直接返回错误。经过一段时间后，将状态切换为“半开状态”。

半开状态(Half-Open)： 允许部分请求通过。如果请求成功，则状态切换为关闭状态，否则，切换回打开状态。

考点2: 微服务架构与面向服务架构的对比

【2022年下】以下关于微服务架构与面向服务架构的描述中，正确的是()。

A. 两者均采用去中心化管理

B. 两者均采用集中式管理

C. 微服务架构采用去中心化管理，面向服务架构采用集中式管理

D. 微服务架构采用集中式管理，面向服务架构采用去中心化管理

【参考答案】 C

【点拨】微服务架构使用去中心化的扁平化管理方式，每个服务都是一个独立的应用程序，独立管理、使用独立的数据库、独立部署和独立运行。 SOA 是一种整体式架构，使用集中式的管理方式和统一的数据中心。答案选择C 选项。

【模拟题练习】下面关于微服务与 SOA 的说法正确的是()

A. 微服务是整体的，服务能放一起的都放一起，而SOA 则是能拆分的就拆分

B. 微服务是细粒度，而SOA 是粗粒度

C. 微服务是水平分多层的，而 SOA 是纵向业务划分

D. 在微服务中企业服务总线 (ESB) 充当了服务之间通信的角色，而 SOA 使用轻量级的通信方式，如 HTTP

【参考答案】 B

知识点5: 云原生架构风格

(1) 云计算基本概念：

云计算是集合了大量计算设备和资源，对用户屏蔽底层差异的分布式处理架构，其用户与提供实际服务的计算资源是相分离的。

云计算优点：超大规模、虚拟化、高可靠性、高可伸缩性、按需服务、成本低【前期投入低、综合使用成本也低】。

(2) 分类

按服务类型分类：

SaaS【软件即服务】	基于多租户技术实现，直接提供应用程序
PaaS【平台即服务】	虚拟中间件服务器、运行环境和操作系统
IaaS【基础设施即服务】	包括服务器、存储和网络等服务

按部署方式分类：

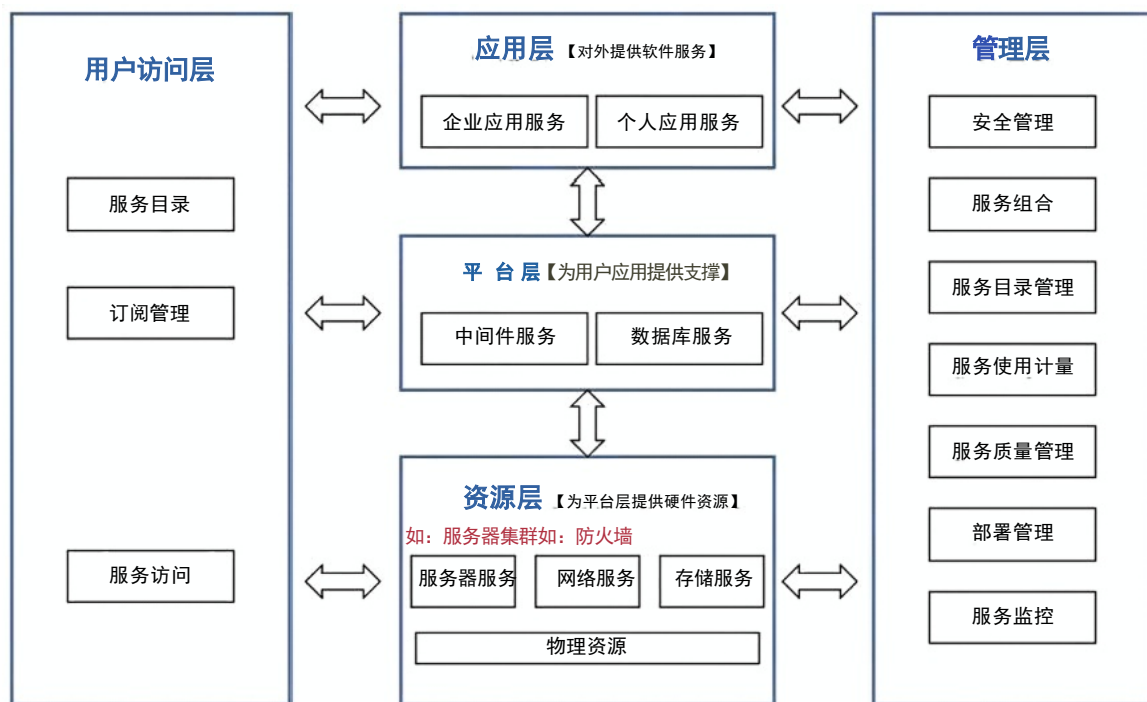
公有云：面向互联网用户需求，通过开放网络提供云计算服务

私有云：面向企业内部提供云计算服务

混合云：兼顾以上两种情况的云计算服务

(3) 云计算架构

【云原生】是基于分布部署和统一运管的分布式云，以容器、微服务、DevOps 等技术为基础建立的一套云技术产品体系。



【管理层】提供对所有层次云计算服务的管理功能。

【用户访问层】方便用户使用云计算服务所需的各种支撑服务，针对每个层次的云计算服务都需要提供相应的访问接口。

【应用层】提供软件服务，如：财务管理，客户关系管理，商业智能。

【平台层】为用户提供对资源层服务的封装，使用户可以构建自己的应用。

【资源层】提供虚拟化的资源，从而隐藏物理资源的复杂性。如：服务器，存储。

(4) 云原生架构设计原则：

服务化原则：使用微服务

弹性原则： 可根据业务变化自动伸缩

可观测原则：通过日志、链路跟踪和度量

韧性原则：面对异常的抵御能力

所有过程自动化原则：自动化交付工具

零信任原则： 默认不信任网络内部和外部的任何人/设备/系统

架构持续演进原则： 业务高速迭代情况下的架构与业务平衡

(5) 容器技术

Docker 容器基于操作系统虚拟化技术，共享操作系统内核、轻量、没有资源损耗、秒级启动，极大提升了系统的**应用部署密度和弹性**。

Kubernetes 提 供 了 分 布 式 应 用 管 理 的 核 心 能 力。

【资源调度】 根据请求资源量在集群中选择合适的节点来运行应用。

【应用部署与管理】 支持应用的自动发布与应用的回滚。

【自动修复】 当宿主机或者 OS 出现故障，节点健康检查会自动进行应用迁移。

【服务发现与负载均衡】 结 合 DNS 和 负 载 均 衡 机 制，支持容器化应用之间的相互通信。

【弹性伸缩】 可以对这个业务进行自动扩容。

【声明式API】 开发者可以关注于应用自身，而非系统执行细节。

【可扩展性架构】 所 有 K8s 组 件 都 是 基 于 一 致 的、开放的API 实现和交互。

【可移植性】 K8s 通过一系列抽象如 Load Balance Service（负载均衡服务）、CNI（容器网络接口）、CSI（容器存储接口），帮助业务应用可以屏蔽底层基础设施的实现差异，实现容器灵活迁移的设计目标。

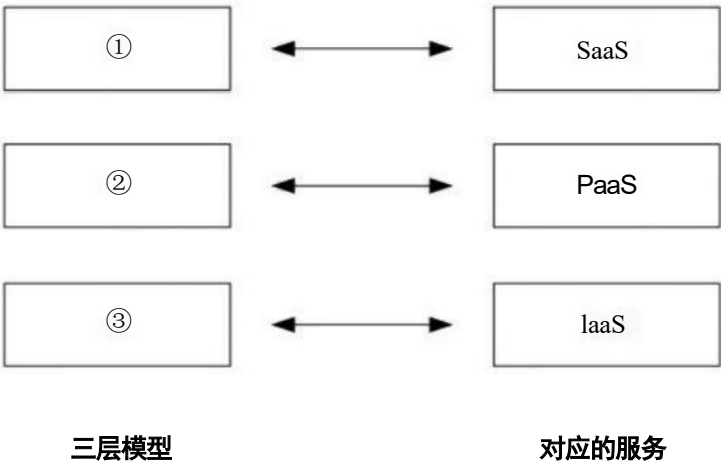
虚拟机技术与容器技术的对比：

对比项	虚拟机技术	容器技术
镜像大小	包含GuestOS, G量级以上	仅包含运行的Bin/Lib, M量级
资源要求	CPU与内存按核、按G分配	CPU与内存按单核、低于G量级分配
启动时间	分钟级	毫秒级
可持续性	跨物理机迁移	跨操作系统平台迁移
弹性伸缩	VM伸缩, CPU/内存手动伸缩	实例自动伸缩、CPU内存自动在线伸缩
隔离策略	操作系统、系统级别	Cgroups, 进程级别

考点1: 云计算

【2022年下】云计算服务体系结构如下图所示，图中①、②、③分别与 SaaS、PaaS、IaaS 相对应，

图中①、②、③应为（）。



- A. 应用层、基础设施层、平台层
- B. 应用层、平台层、基础设施层
- C. 平台层、应用层、基础设施层
- D. 平台层、基础设施层、应用层

【参考答案】B

考点2: 云原生架构

【模拟题练习】云原生架构本身作为一种架构，也有若干架构原则作为应用架构的核心架构控制面，通过遵从这些架构原则可以让技术主管和架构师在做技术选择时不会出现大的偏差。下面不属于云架构原则的是（）

- A. 服务化原则

- B. 可观测原则
- C. 可信任原则
- D. 韧性原则

【参考答案】C

知识点5: 边缘计算

边缘计算是指在靠近物或数据源头的一侧，采用网络、计算、存储、应用核心能力为一体的开放平台，就近提供最近端服务。

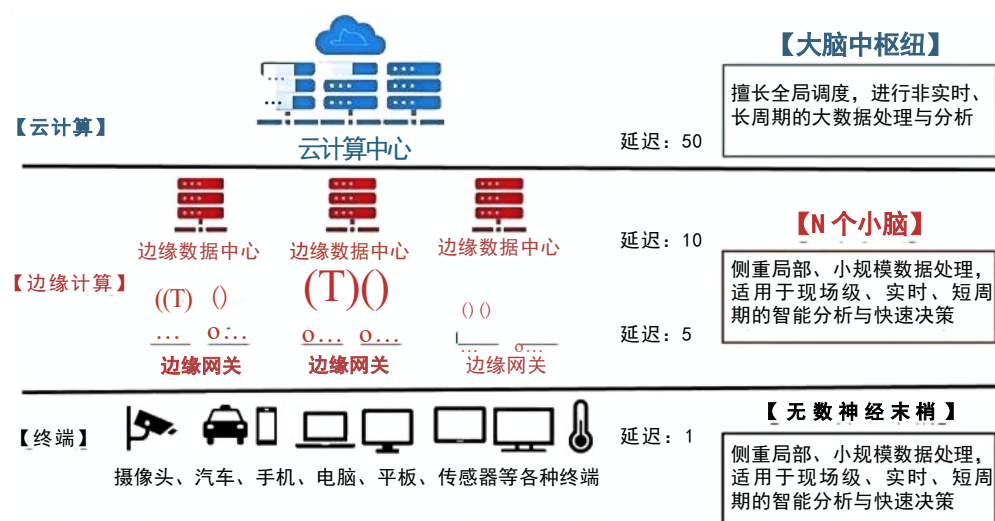
边缘计算的本质：计算处理职能的本地化。

边缘计算类型：

【云边缘】云服务在边缘侧的延伸

【边缘云】在边缘侧构建中小规模云服务能力

【边缘网关】以云化技术与能力重构原有嵌入式网关系统



边云协同的分类：

【资源协同】边缘节点有基础设施资源的调度管理能力，可与云端协同。

【数据协同】边缘节点采集数据并初步分析，再发给云端做进一步处理。

【智能协同】分布式智能，云端做集中式模型训练，再将模型下发到边缘节点。

【应用管理协同】边缘节点提供应用部署与运行环境，云端主要提供应用开发、测试环境。

【业务管理协同】边缘节点提供模块化、微服务化的应用/数字孪生/网络等应用实例；云端主要提供按照客户需求实现应用/数字孪生/网络等的业务编排能力。

【服务协同】边缘节点按照云端策略实现部分 ECSaaS 服务，通过 ECSaaS 与云端SaaS 的协同实现面向客户的按需 SaaS 服务；云端主要提供 SaaS 服务在云端和边缘节点的服务分布策略，以及云端承担的 SaaS 服务能力。

考点1: 边云协同

【模拟题练习】下面关于边云协同的分类，说法错误的是()。

- A. 智能协同：分布式智能，云端做集中式模型训练，再将模型下发到边缘节点
- B. 应用管理协同：边缘节点提供应用部署与运行环境，云端主要提供应用开发、测试环境
- C. 服务协同：边缘节点提供模块化、微服务化的应用/数字孪生/网络等应用实例；云端主要提供按照客户需求实现应用/数字孪生/网络等的业务编排能力
- D. 数据协同：边缘节点采集数据并初步分析，再发给云端做进一步处理

【参考答案】C

4.4 ADL

知识点1: 架构描述语言ADL

(1)什么是ADL

ADL 是这样一种形式化语言，它在底层语义模型的支持下，为软件系统的概念体系结构建模提供了具体语法和概念框架。

C2SADL **【基于组件和消息的软件架构描述语言】**

Wright **【分布、并发类型的架构描述语言】**

ACME **【架构互换语言】**

UniCon **【基于组件和连接的架构描述语言】**

Rapide **【基于事件的架构描述语言】**

其他**【Darwin、MetaH、Aesop、Weaves、SADL、xADL】**

(2)ADL 的三个基本元素

构件：计算或数据存储单元；

连接件：用于构件之间交互建模的体系结构构造块及其支配这些交互的规则；

架构配置：描述体系结构的构件与连接件的连接图。

(3)主要的架构描述语言

Aesop：支持体系结构风格的应用；

MetaH: 为设计者提供了关于实时电子控制软件系统的设计指导;

C2: 支持基于消息传递风格的用户界面系统的描述;

Rapide: 支持体系结构设计的模拟并提供了分析模拟结果的工具;

SADL: 提供了关于体系结构加细的形式化基础;

Unicon: 支持异构的构件和连接类型并提供了关于体系结构的高层编译器;

Wright: 支持体系结构构件之间交互的说明和分析。

考点1: ADL 概念

【2015年下】架构描述语言 (Architecture Description Language,ADL) 是一种为明确说明软件系统的概念架构和对这些概念架构建模提供功能的语言。ADL 主要包括以下组成部分: 组件、组件接口、()和架构配置。

- A. 架构风格
- B. 架构实现
- C. 连接件
- D. 组件约束

【参考答案】C

【模拟题练习】架构描述语言 (Architecture Description Language,ADL) 是一种为明确说明软件系统的概念架构和对这些概念架构建模提供功能的语言。ADL 主要包括以下组成部分: 组件、组件接口、连接件和架构配置。其中()是用来建立组件间的交互以及支配这些交互规则的架构构造模块。

- A. 组件
- B. 组件接口
- C. 连接件
- D. 架构配置

【参考答案】C

【点拨】组件(构件)是一个计算单元或数据存储。也就是说,组件是计算与状态存在的场所。在架构中,一个构件可能小到只有一个过程或大到整个应用程序。

连接件是用来建立组件间的交互以及支配这些交互规则的架构构造模块。

架构配置或拓扑是描述架构的组件与连接件的连接图。

【2024年上】ADL 架构描述语言是一种基于模型的描述语言，主要用于描述软件系统的构建组件、关系和行为，典型的ADL包 括（ ）。

①UNICON ②RAPIDE ③ACME ④AADL ⑤MDA

A.①②③⑤

B.②③④⑤

C.①②③④

D.①③④⑤

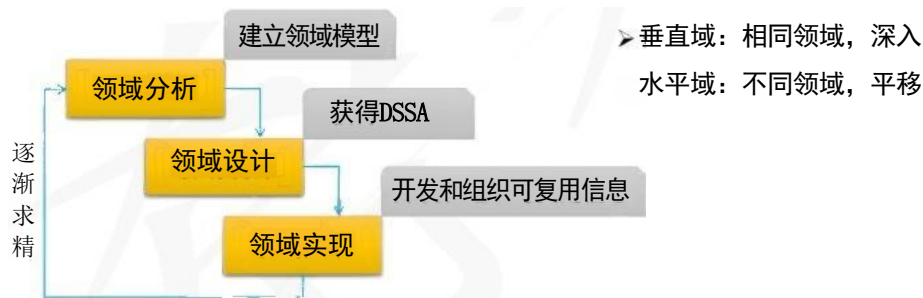
【参考答案】C

4.5 特定领域软件架构 (DSSA)

知识点1：DSSA

定义：特定领域软件架构以一个特定问题领域为对象，形成由领域参考模型、参考需求、参考架构等组成的开发基础架构，支持一个特定领域中多个应用的生成。

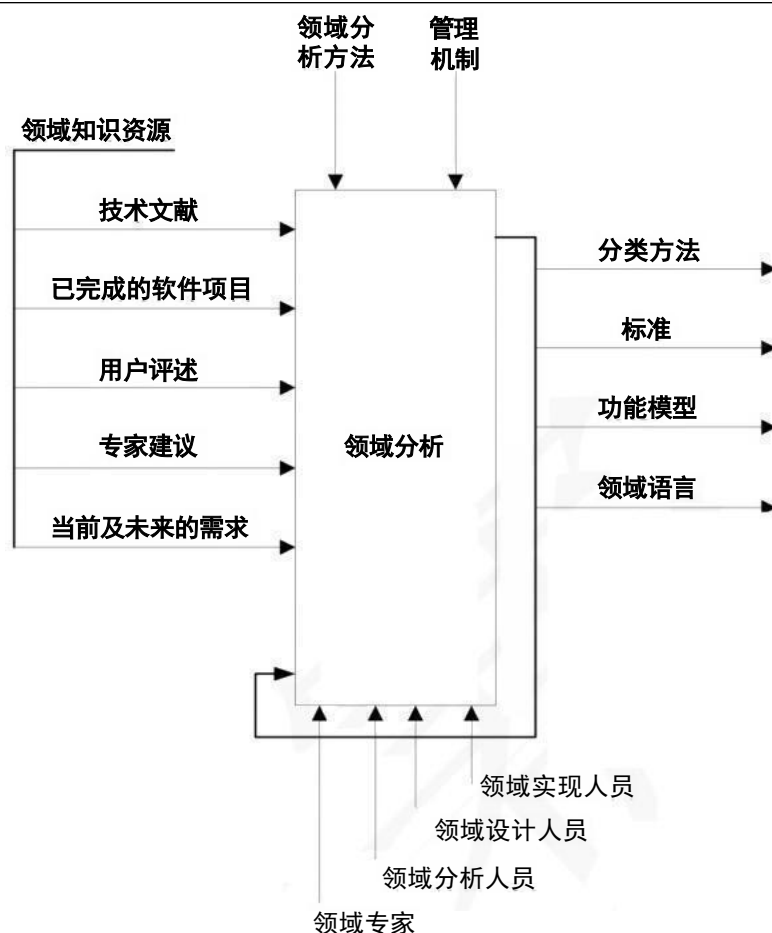
一、基本活动



垂直域：某一个狭小领域或者说某个行业的共性抽象。

水平域：多个行业可通用的一些共性的抽象。

二、领域分析机制



(1) 领域专家

领域专家可能包括该领域中系统的有经验的用户、从事该领域中系统的需求分析、设计、实现以及项目管理的有经验的软件工程师等。

领域专家的主要任务包括提供关于领域中系统的需求规约和实现的知识，帮助组织规范的、一致的领域字典，帮助选择样本系统作为领域工程的依据，复审领域模型、DSSA 等领域工程产品，等等。

(2) 领域分析人员

领域分析人员应由具有知识工程背景的有经验的系统分析员来担任。

领域分析人员的主要任务包括控制整个领域分析过程，进行知识获取，将获取的知识组织到领域模型中，根据现有系统、标准规范等验证领域模型的准确性和一致性，维护领域模型。

领域分析人员应熟悉软件重用和领域分析方法；熟悉进行知识获取和知识表示所需的技术、语言 and 工具；应具有一定的该领域的经验，以便于分析领域中的问题及与领域专家进行交互；应具有较高的进行抽象、关联和类比的能力；应具有较高的与他人交互和合作的能力。

(3) 领域设计人员

领域设计人员应由有经验的软件设计人员来担任。

领域设计人员应熟悉软件重用和领域设计方法；熟悉软件设计方法；应有一定的该领域的经验，以便于分析领域中的问题及与领域专家进行交互。

领域实现人员应由有经验的程序设计人员来担任。

领域实现人员应熟悉软件重用、领域实现及软件再工程技术；熟悉程序设计；具有一定的该领域的经验。

```
graph TD; A[定义领域范围] --> B[定义领域特定的元素]; B --> C[定义领域特定的设计和实现需求约束]; C --> D[定义领域模型和架构]; D --> E[产生、搜集可复用的产品单元]; E --> A;
```

并发的
递归的
反复的
螺旋型的

领域架构师

领域开发环境

参考结构

参考需求

架构

领域模型

开发工身

应用工程师

领域特定的应用开发环境

实例化的架构

操作员

应用执行环境

考点1: DSSA 基本活动

【2010年下】特定领域软件架构（DSSA）是在一个特定应用领域为一组应用提供组织结构参考的标准软件架构。实施DSSA的过程中包括一系列基本的活动，其中（）活动的主要目的是为了获得DSSA。该活动参加人员中，（）的主要任务是提供关于领域中系统的需求规约和实现的知识。

- | | | | |
|---------|----------|----------|----------|
| A. 领域需求 | B. 领域分析 | C. 领域设计 | D. 领域实现 |
| A. 领域专家 | B. 领域分析者 | C. 领域设计者 | D. 领域实现者 |

【参考答案】CA

【点拨】本题主要考查特定领域软件架构的基本定义和基本活动。特定领域软件架构（DSSA）是在一个特定应用领域为一组应用提供组织结构参考的标准软件架构。实施DSSA的过程中包括一系列基本的活动，其中领域设计活动的主要目的是为了获得DSSA。该活动参加人员中，领域专家的主要任务是提供关于领域中系统的需求规约和实现的知识。

【2021年下】特定领域软件架构（Domain Specific Software Architecture, DSSA）是指特定应用领域中为一组应用提供组织结构参考的标准软件架构。从功能覆盖的范围角度，（）定义了一个特定的系统族，包含整个系统族内的多个系统，可作为该领域系统的可行解决方案的一个通用软件架构；（）定义了多个系统和多个系统族中功能区域的共有部分，在子系统级上涵盖多个系统族的特定部分功能。

- | | | | |
|--------|--------|--------|--------|
| A. 垂直域 | B. 水平域 | C. 功能域 | D. 属性域 |
| A. 垂直域 | B. 水平域 | C. 功能域 | D. 属性域 |

【参考答案】AB

【点拨】从功能覆盖的范围角度理解DSSA中领域的含义有两种方法：

（1）垂直域。定义了一个特定的系统族，导出在该领域中可作为系统的可行解决方案的一个通用软件架构。

（2）水平域。定义了多个系统和多个系统族中功能区域的共有部分，在子系统级上涵盖多个系统（族）的特定部分功能。

在特定领域架构中，垂直域关注的是与行业相关的，聚焦于行业特性的内容，而水平域关注的是各行业共性部分的内容。

考点2：领域分析机制

【2014年下】特定领域软件架构 (Domain Specific Software Architecture, DSSA) 是在一个特定应用领域中，为一组应用提供组织结构参考的标准软件体系结构。参加 DSSA 的人员可以划分为多种角色，其中 () 的任务是控制整个领域分析过程，进行知识获取，将获取的知识组织到领域模型中；() 的任务是根据领域模型和现有系统开发出 DSSA，并对 DSSA 的准确性和一致性进行验证。

A. 领域专家 B. 领域分析者 C. 领域设计者 D. 领域实现者

A. 领域专家 B. 领域分析者 C. 领域设计者 D. 领域实现者

【参考答案】 BC

考点3: 三层次模型

【2013年下】特定领域软件架构 (Domain Specific Software Architecture, DSSA) 是在一个特定应用领域中，为一组应用提供组织结构参考的标准软件体系结构。DSSA 通常是一个具有三个层次的系统模型，包括 () 环境、领域特定应用开发环境和应用执行环境，其中 () 主要在领域特定应用开发环境中工作。

A. 领域需求 B. 领域开发 C. 领域执行 D. 领域应用

A. 操作员 B. 领域架构师 C. 应用工程师 D. 程序员

【参考答案】 BC

4.6 基于架构的软件开发方法

知识点1: ABSD

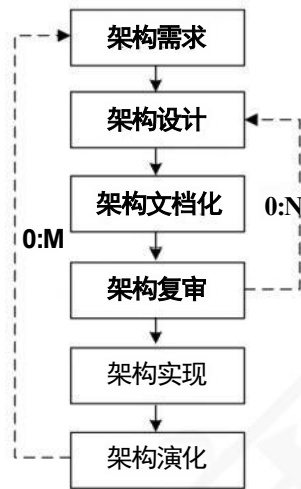
(1) 基于架构的软件设计 (ABSD)

ABSD 方法是架构驱动，即强调由业务【商业】、质量和功能需求的组合驱动架构设计。

ABSD 方法有三个基础。第一个基础是功能的分解。在功能分解中，ABSD 方法使用已有的基于模块的内聚和耦合技术；第二个基础是通过选择架构风格来实现质量和业务需求；第三个基础是软件模板的使用。

视角与视图：从不同的视角来检查，所以会有不同的视图。

用例用来捕获功能需求、特定场景【刺激、环境、响应】用来捕获质量需求。



ABSD 能很好的 **【支持软件重用】**。

ABSD 方法是一个自顶向下，递归细化的方法。

软件系统的体系结构通过该方法得到细化，直到能产生软件构件和类。

架构文档化过程的主要输出结果是架构规格说明和测试架构需求的质量设计说明书这两个文档。

文档的完整性和质量是软件架构成功的关键因素。

关于文档的三大注意事项:

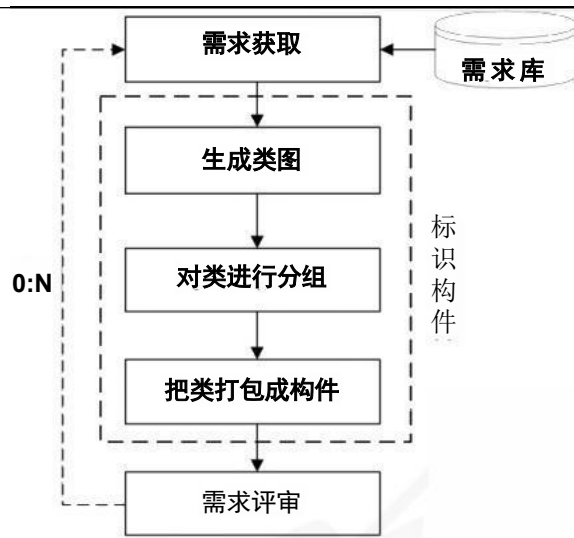
文档要从使用者的角度进行编写

必须分发给所有与系统有关的开发人员

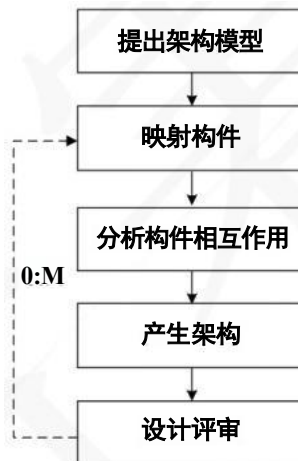
且必须保证开发者手上的文档是最新的

架构复审【架构评估】的目的是标识潜在的风险，及早发现架构设计中的缺陷和错误

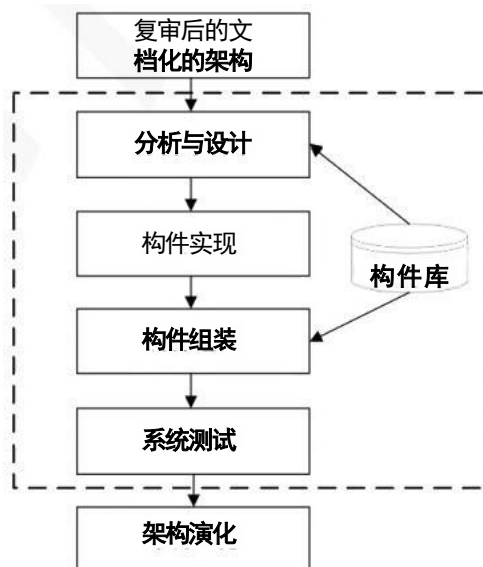
(3) 架构需求过程



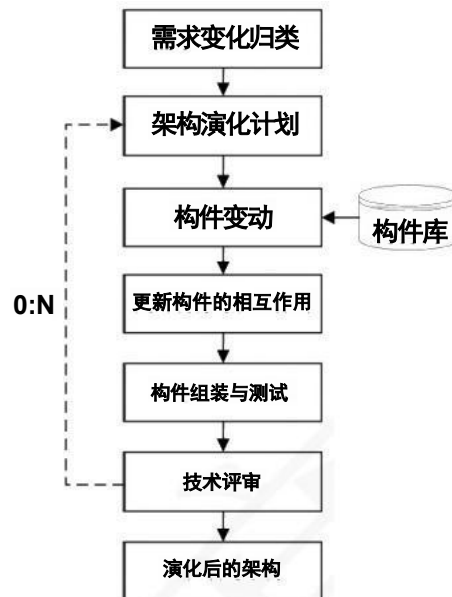
(4) 架构设计过程



(5) 架构实现过程



(6) 架构演化过程



考点1: ABSD 的基本概念

【2009年下】基于架构的软件设计 (ABSD) 强调由商业、质量和功能需求的组合驱动软件架构设计。以下关于ABSD 的叙述中，错误的是（）。

- A. 使用ABSD 方法，设计活动可以从项目总体功能框架明确就开始
- B. ABSD 方法是一个自顶向下，递归细化的过程
- C. ABSD 方法有三个基础：功能分解、选择架构风格实现质量和商业需求以及软件模板的使用
- D. 使用ABSD 方法，设计活动的开始意味着需求抽取和分析活动可以终止

【参考答案】D

【点拨】基于架构的软件设计 (ABSD) 强调由商业、质量和功能需求的组合驱动软件架构设计。使用ABSD 方法，设计活动可以从项目总体功能框架明确就开始，并且设计活动的开始并不意味着需求抽取和分析活动可以终止，而是应该与设计活动并行。

【2010年下】基于软件架构的设计 (Architecture Based Software Design, ABSD) 强调由商业、质量和功能需求的组合驱动软件架构设计。它强调采用（）来描述软件架构，采用（）来描述需求。

- | | | | |
|-----------|----------|------------|------------|
| A. 类图和序列图 | B. 视角与视图 | C. 构件和类图 | D. 构件与功能 |
| A. 用例与类图 | B. 用例与视角 | C. 用例与质量场景 | D. 视角与质量场景 |

【参考答案】BC

【2021年下】基于架构的软件设计 (Architecture Based Software Design, ABSD) 方法是架构驱动的方法, 该方法是一个()的方法, 软件系统的架构通过该方法得到细化, 直到能产生()。

- | | |
|------------|----------------|
| A. 自顶向下 | B. 自底向上 |
| C. 原型 | D. 自顶向下和自底向上结合 |
| A. 软件质量属性 | B. 软件连接性 |
| C. 软件构件或模块 | D. 软件接口 |

【参考答案】AC

考点2: ABSD 的开发过程

【2019年下】ABSDM(Architecture Based Software Design Model) 把整个基于体系结构的软件过程划分为体系结构需求、体系结构设计、体系结构文档化、()、()和体系结构演化等6个子过程。其中, ()过程的主要输出结果是体系结构规格说明和测试体系结构需求的质量设计说明书。

- | | | | |
|-----------|-----------|------------|-----------|
| A. 体系结构复审 | B. 体系结构测试 | C. 体系结构变更 | D. 体系结构管理 |
| A. 体系结构实现 | B. 体系结构测试 | C. 体系结构建模 | D. 体系结构管理 |
| A. 体系结构设计 | B. 体系结构需求 | C. 体系结构文档化 | D. 体系结构测试 |

【参考答案】AAC

【点拨】ABSDM 模型把整个基于体系结构的软件过程划分为: 体系结构需求、设计、文档化、复审、实现和演化等6个过程。因此第一空选择A 选项体系结构复审, 第二空选择A 选项体系结构实现。

体系结构需求: 需求过程主要是获取用户需求, 标识系统中所要用到的构件。

体系结构设计: 体系结构设计是一个迭代过程, 如果要开发的系统能够从已有的系统中导出大部分, 则可以使用已有系统的设计过程。

体系结构文档化: 绝大多数的体系结构都是抽象的, 由一些概念上的构件组成, 因此要去实现体系结构, 还必须得把体系结构文档化。体系结构文档化过程的主要输出结果是体系结构规格说明和测试体系结构需求的质量设计说明书这2个文档。本题第三空选择C 选项。

体系结构复审: 体系结构设计、文档化和复审是一个迭代过程。复审的目的是表示潜在的风险, 及早发现体系结构设计中的缺陷和错误, 包括体系结构能否满足需求、质量需求是否在设计中得到体现、层次是否清晰、构件划分是否合理、文档表达是否明确、构件设计是否满足功能与性能的要求等。

体系结构实现: 所谓“实现”就是要用实体显示出一个软件体系结构, 即要符合体系结构描述的结构性设计决策, 分割成规定的构件, 按规定的方式互相交互。整个实现过程是以复审后的文档化的体系结构说明书为基础的, 每个构件必须满足软件体系结构中说明的对其他构件的责任。最后一步是测试, 包括单个构件的功能性测试和被组装应用的整体功能和性能测试。

体系结构演化：在构件开发过程中，用户的需求可能还有变动。在软件开发完毕，正常运行后，由一个单位移植到另一个单位，需求也会发生变化。在这两种情况下，就必须相应地修改软件体系结构，以适应新的变化了的软件需求。体系结构演化是使用系统演化步骤去修改应用，以满足新的需求。

【2014年下】在对一个软件系统的架构进行设计与确认之后，需要进行架构复审。架构复审的目的是为了标识潜在的风险，及早发现架构设计中的缺陷和错误。在架构复审过程中，主要由（）决定架构是否满足需求、质量需求是否在设计中得到体现。

- A. 系统分析师与架构师
- B. 用户代表与领域专家
- C. 系统所有者与项目经理
- D. 系统开发与测试人员

【参考答案】B

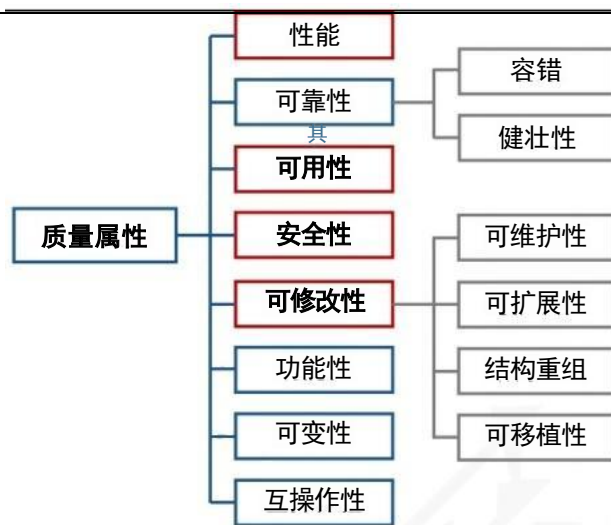
【点拨】架构复审一词来自于ABSD。在ABSD中，架构设计、文档化和复审是一个迭代过程。从这个方面来说，在一个主版本的软件架构分析之后，要安排一次由外部人员（用户代表和领域专家）参加的复审。

复审的目的是标识潜在的风险，及早发现架构设计中的缺陷和错误，包括架构能否满足需求、质量需求是否在设计中得到体现、层次是否清晰、构件的划分是否合理、文档表达是否明确、构件的设计是否满足功能与性能的要求等等。

由外部人员进行复审的目的是保证架构的设计能够公正地进行检验，使组织的管理者能够决定正式实现架构。

4.7 架构评估

知识点1：架构设计重点关注非功能设计（质量属性）



软件系统质量属性是一个系统的可测量或者可测试的属性，用来描述系统满足利益相关者需求的程度。基于软件系统的生命周期，可以将软件系统的质量属性分为开发期质量属性和运行期质量属性2个部分。

开发期质量属性主要指在软件开发阶段所关注的质量属性，主要包含6个方面：

易理解性：指设计被开发人员理解的难易程度。

可扩展性：软件因适应新需求或需求变化而增加新功能的能力，也称为灵活性。

可重用性：指重用软件系统或某一部分的难易程度。

可测试性：对软件测试以证明其满足需求规范的难易程度。

可维护性：当需要修改缺陷、增加功能、提高质量属性时，识别修改点并实施修改的难易程度。

可移植性：将软件系统从一个运行环境转移到另一个不同的运行环境的难易程度。

运行期质量属性主要指在软件运行阶段所关注的质量属性，主要包含7个方面：

性能：性能是指软件系统及时提供相应服务的能力，如速度、吞吐量和容量等的要求。

安全性：指软件系统同时兼顾向合法用户提供服务，以及阻止非授权使用的能力。

可伸缩性：指当用户数和数据量增加时，软件系统维持高服务质量的能力。例如，通过增加服务器来提高能力。

互操作性：指本软件系统与其他系统交换数据和相互调用服务的难易程度。

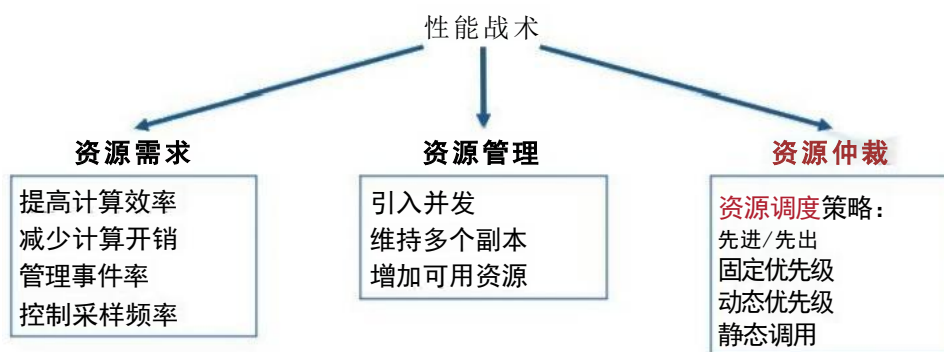
可靠性：软件系统在一定的时间内持续无故障运行的能力。

可用性：指系统在一定时间内正常工作的时间所占的比例。可用性会受到系统错误，恶意攻击，高负载等问题的影响。

鲁棒性：是指软件系统在非正常情况（如用户进行了非法操作、相关的软硬件系统发生了故障等）下仍能够正常运行的能力，也称健壮性或容错性。

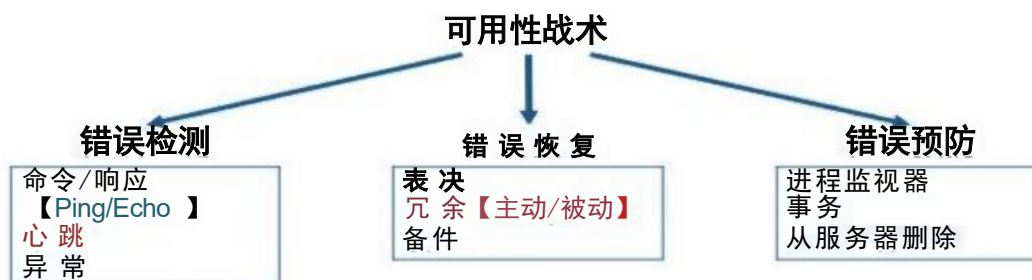
(1) 性能

性能 (performance) 是指系统的响应能力，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。例如： a. 同时支持1000并发； b. 响应时间小于1s；c. 显示分辨率达到4K。



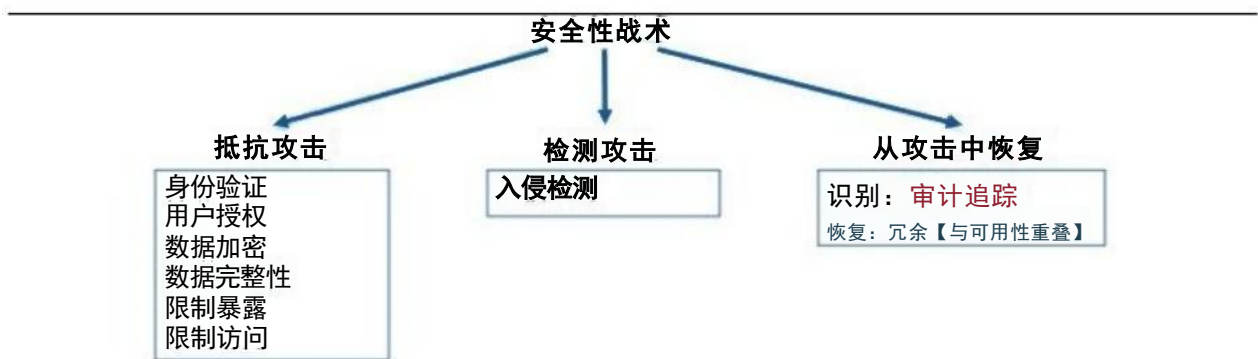
(2) 可用性

可用性 (availability) 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。例如： a. 主服务器故障，1分钟内切换至备用服务器； b. 系统故障，1小时内修复； c. 系统支持7×24小时工作。



(3) 安全性

安全性 (security) 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。安全性又可划分为机密性【信息不泄露给未授权的用户】、完整性【防止信息被篡改】、不可否认性【不可抵赖】及可控性【对信息的传播及内容具有控制的能力】等特性。例如： a. 可抵御SQL注入攻击； b. 对计算机的操作都有完整记录； c. 用户信息数据库授权必须保证99.9%可用。



(4) 可修改性

可修改性 (modifiability) 是指能够快速地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可修改性。(可扩展性与之相近)例如： a. 更改系统报表模块，必须在2人周内完成； b.对 Web 界面风格进行修改，修改必须在4人月内完成。



(5) 易用性

易用性 (usability) 关注的是对用户来说完成某个期望任务的容易程度和系统所提供的用户支持的种类。例如： a. 界面友好； b. 新用户学习使用系统时间不超过2小时。

(6) 可测试性

可测试性 (testability) 是指通过测试揭示软件缺陷的容易程度。 例如： a. 提供远程调试接口，支持远程调试。

考点1: 质量属性

【2021年下】在架构评估过程中，评估人员所关注的是系统的质量属性。其中，()是指系统的响应能力：即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()。

- | | | | |
|--------|-------|--------|--------|
| A. 安全性 | B. 性能 | C. 可用性 | D. 可靠性 |
| A. 个数 | B. 速度 | C. 消耗 | D. 故障率 |

【参考答案】BA

【. 点拨】性能 (performance): 是指系统的响应能力, 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。

可用性 (availability): 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。

安全性 (security): 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。

可靠性 (reliability): 产品在规定的条件和规定的时间内, 完成规定功能的能力。

【2010年下】某服务器软件系统对可用性 (Availability)、性能 (Performance) 和可修改性 (Modification) 的要求较高, () 设计策略能够提高该系统的可用性, () 设计策略能够提高该系统的性能, () 设计策略能够提高该系统的可修改性。

- | | | | |
|--------------|----------|----------|------------|
| A. Ping/Echo | B. 限制访问 | C. 运行时注册 | D. 接口-实现分离 |
| A. 分层结构 | B. 事务机制 | C. 主动冗余 | D. 队列调度 |
| A. 信息隐藏 | B. 记录/回放 | C. 任务模型 | D. 回滚 |

【参考答案】ADA

【. 点拨】本题主要考查质量属性以及实现质量属性的一般策略, 不同策略主要针对一个或多个软件质量属性, 其中Ping/Echo 主要提高系统的可用性; 限制访问主要提高系统的安全性; 运行时注册主要提高系统的可修改性; 接口-实现分离主要提高系统的可修改性; 主动冗余提高系统的可用性; 队列调度主要提高系统的性能; 信息隐藏主要提高系统的可修改性; 记录-回放主要提高系统的可测试性, 等等。

【2012年下】某公司欲开发一个在线交易系统, 在架构设计阶段, 公司的架构师识别出3个核心质量属性场景。其中“在并发用户数量为1000人时, 用户的交易请求需要在0.5秒内得到响应”主要与() 质量属性相关, 通常可采用() 架构策略实现该属性; “当系统由于软件故障意外崩溃后, 需要在0.5小时内恢复正常运行”主要与() 质量属性相关, 通常可采用() 架构策略实现该属性; “系统应该能够抵挡恶意用户的入侵行为, 并进行报警和记录”主要与() 质量属性相关, 通常可采用() 架构策略实现该属性。

- | | | | |
|----------|---------|---------|----------|
| A. 性能 | B. 吞吐量 | C. 可靠性 | D. 可修改性 |
| A. 操作串行化 | B. 资源调度 | C. 心跳 | D. 内置监控器 |
| A. 可测试性 | B. 易用性 | C. 可用性 | D. 互操作性 |
| A. 主动冗余 | B. 信息隐藏 | C. 抽象接口 | D. 记录/回放 |

-
- | | | | |
|---------|---------|--------|----------|
| A.可用性 | B.安全性 | C.可测试性 | D.可修改性 |
| A.内置监控器 | B.记录/回放 | C.追踪审计 | D.维护现有接口 |

【参考答案】ABCABC

【点拨】本题主要考查考生对质量属性的理解和质量属性实现策略的掌握。对于题干描述：“在并发用户数量为1000人时，用户的交易请求需要在0.5秒内得到响应”，主要与性能这一质量属性相关，实现该属性的常见架构策略包括：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。“当系统由于软件故障意外崩溃后，需要在0.5小时内恢复正常运行”主要与可用性质量属性相关，通常可采用心跳、Ping/Echo、主动冗余、被动冗余、选举等架构策略实现该属性；“系统应该能够抵挡恶意用户的入侵行为，并进行报警和记录”主要与安全性质量属性相关，通常可采用入侵检测、用户认证、用户授权、追踪审计等架构策略实现该属性。

【2024年上】基于软件系统的生命周期，可以将软件系统的质量属性分为（）两个部分。

- A.需求分析期质量属性和设计期质量属性
- B.开发期质量属性和运行期质量属性
- C.设计期质量属性和开发期质量属性
- D.设计期质量属性和运行期质量属性

【参考答案】B

知识点2: 架构评估方法

(1) 相关概念

风险点：系统架构风险是指架构设计中潜在的、存在问题的架构决策所带来的隐患。

非风险点：是指不会带来隐患，一般以“XXX 要求是可以实现【或接受】的”方式表达。

敏感点：敏感点是一个或多个构件(和/或构件之间的关系)的特性，它能影响系统的某个质量属性。

权衡点：影响多个质量属性的特性，是多个质量属性的敏感点。

场景：场景是从风险承担者的角度与系统交互的简短描述。场景可从六个方面进行描述：刺激源、刺激、制品、环境、响应、响应度量。



刺激源 (Source): 这是某个生成该刺激的实体(人、计算机系统或者任何其他刺激器)。

刺激 (Stimulus): 该刺激是当刺激到达系统时需要考虑的条件。

环境 (Environment): 该刺激在某些条件内发生。当激励发生时，系统可能处于过载、运行或者其他情况。

制品 (Artifact): 某个制品被激励。这可能是整个系统，也可能是系统的一部分。

响应 (Response): 该响应是在激励到达后所采取的行动。

响应度量 (Measurement): 当响应发生时，应当能够以某种方式对其进行度量，以对需求进行测试。

(2) 评估方式

评估方式	调查问卷或检查表		场景	度量
	调查问卷	检查表		
通用性	通用	特定领域	特定系统	通用或特定领域
评估者对架构的了解程度	粗略了解	无限制	中等了解	精确了解
实施阶段	早	中	中	中
客观性	主观	主观	较主观	较客观

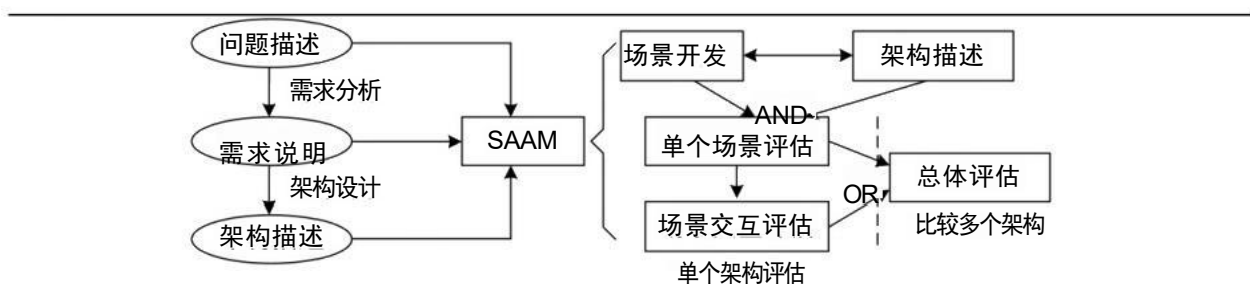
(3) 基于场景的评估方法

过程：

- 确定应用领域的功能和软件架构的结构之间的映射；
- 设计用于体现待评估质量属性的场景；
- 分析软件架构对场景的支持程度。

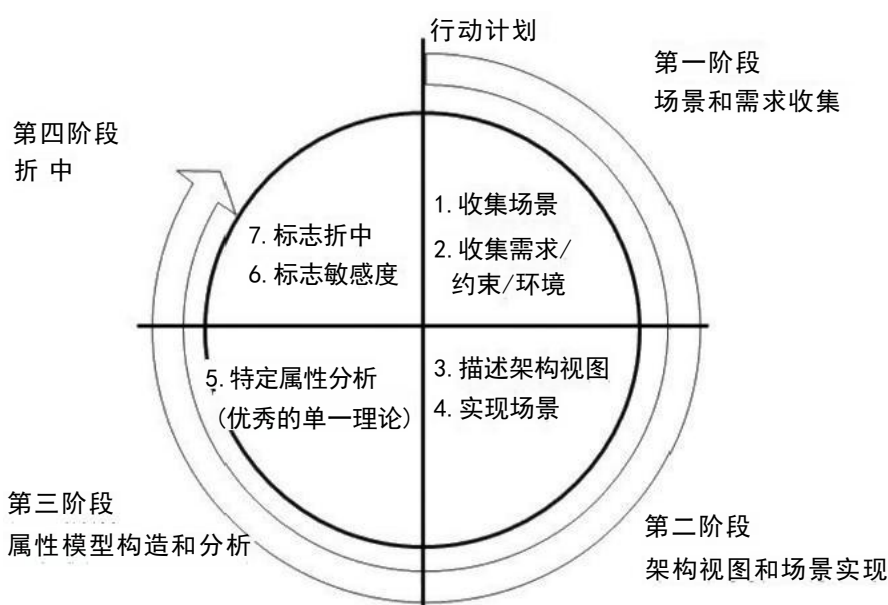
软件架构分析法 (SAAM)

(最初用于分析架构可修改性，后扩展到其它质量属性。)



架构权衡分析法 (ATAM)

(在SAAM 的基础上发展起来的, 主要针对性能、实用性、安全性和可修改性, 在系统开发之前, 对这些质量属性进行评价和折中。)



质量效用树

ATAM 方法采用效用树 (Utility tree) 这一工具来对质量属性进行分类和优先级排序。效用树的结构包括: 树根—质量属性—属性分类—质量属性场景(叶子节点)。

得到初始的效用树后, 需要修剪这棵树, 保留重要场景(通常不超过50个), 再对场景按重要性给定优先级(用H/M/L 的形式), 再按场景实现的难易度来确定优先级(用H/M/L 的形式), 这样对所选定的每个场景就有一个优先级对(重要度、难易度), 如 (H、L) 表示该场景重要且易实现。

ATAM方法架构评估实践:

①描述和介绍阶段

介绍ATAM 方法

描述商业目标(主要利益相关者包括: 最终用户、架构师和应用程序开发人员)

描述体系结构(侧重于体系结构、时间可用性以及体系结构的质量要求。关键问题包括: 技术约束及与当前系统交互的其它系统)

②调查和分析阶段

标识体系结构步骤

产生质量属性树

分析体系结构步骤(四个阶段：调查架构方法、创建分析问题、分析问题的答案、找出风险、非风险、敏感点和权衡点)

③测试阶段

讨论质量需求的次序(使用头脑风暴的三种场景：【用例场景】在这种情况下，利益相关者就是最终用户。【增长情景】代表了架构发展的方式。【探索性场景】代表架构中极端的生长形式。)

分析体系结构步骤

④报告阶段

提交结果

考点1: 架构评估方法相关概念

【2011年下】识别风险点、非风险点、敏感点和权衡点是软件架构评估过程中的关键步骤。针对某系统所作的架构设计中，“系统需要支持的最大并发用户数量直接影响传输协议和数据格式”描述了系统架构设计中的一个()；“由于系统的业务逻辑目前尚不清楚，因此现有系统三层架构中的第二层可能会出现功能重复，这会影响系统的可修改性”描述了系统架构设计中的一个()。

- | | | | |
|--------|--------|---------|--------|
| A.敏感点 | B.风险点 | C.非风险点 | D.权衡点 |
| A. 敏感点 | B. 风险点 | C. 非风险点 | D. 权衡点 |

【参考答案】AB

【. 点拨】本题考查架构设计中的一些基本概念。

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。

敏感点是一个或多个构件(和/或构件之间的关系)的特性。研究敏感点可使设计人员或分析员明确在搞清楚如何实现质量目标时应注意什么。

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。例如，改变加密级别可能会对安全性和性能产生非常重要的影响。提高加密级别可以提高安全性，但可能要耗费更多的处理时间，影响系统性能。如果某个机密消息的处理有严格的时间延迟要求，则加密级别可能就会成为一个权衡点。

【2014年下】识别风险点、非风险点、敏感点和权衡点是进行软件架构评估的重要过程。“改变业务数据编码方式会对系统的性能和安全性产生影响”是对()的描述,“假设用户请求的频率为每秒1个,业务处理时间小于30毫秒,则将请求响应时间设定为1秒钟是可以接受的”是对()的描述。

- | | | | |
|--------|---------|--------|--------|
| A. 风险点 | B. 非风险点 | C. 敏感点 | D. 权衡点 |
| A. 风险点 | B. 非风险点 | C. 敏感点 | D. 权衡点 |

【参考答案】DB

【点拨】敏感点是一个或多个构件(和/或构件之间的关系)的特性。研究敏感点可使设计人员或分析员明确在搞清楚如何实现质量目标时应注意什么。

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。例如,改变加密级别可能会对安全性和性能产生非常重要的影响。提高加密级别可以提高安全性,但可能要耗费更多的处理时间,影响系统性能。如果某个机密消息的处理有严格的时间延迟要求,则加密级别可能就会成为一个权衡点。

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。

【2021年下】在架构评估中,场景是从()的角度对与系统交互的描述,一般采用()三方面来对场景进行描述。

- | | | | |
|----------------|---------------|---------------|----------------|
| A.系统设计者 | B.系统开发者 | C. 风险承担者 | D.系统测试者 |
| A. 刺激源, 制品, 响应 | B. 刺激, 制品, 响应 | C. 刺激, 环境, 响应 | D. 参与者, 制品, 环境 |

【参考答案】CC

【点拨】场景(scenarios): 在进行体系结构评估时,一般首先要精确地得出具体的质量目标,并以之作为判定该体系结构优劣的标准。为得出这些目标而采用的机制叫做场景。场景是从风险承担者的角度对与系统的交互的简短描述。在体系结构评估中,一般采用刺激(stimulus)、环境(environment)和响应(response)三方面来对场景进行描述。

考点2: 架构评估方法

【2011年下】架构权衡分析方法(ATAM)是一种常用的软件架构评估方法,下列关于该方法的叙述中,正确的是()。

- | | |
|----------------------|---------------------------|
| A. ATAM 需要对代码的质量进行评估 | B. ATAM 需要对软件系统需求的正确性进行评价 |
|----------------------|---------------------------|

C.ATAM 需要对软件系统进行集成测试

D.ATAM 需要对软件质量属性进行优先级排序

【参考答案】D

【点拨】架构权衡分析方法 (Architecture Tradeoff Analysis Method, ATAM), 它是评价软件架构的一种综合全面的方法。这种方法不仅可以揭示出架构满足特定质量目标的情况, 而且 (因为它认识到了架构决策会影响多个质量属性) 可以使我们更清楚地认识到质量目标之间的联系——即如何权衡诸多质量目标。

ATAM 是针对软件架构的评估方法, 其层次较高, 不会涉及具体代码质量的评估, 所以A 选项不正确。而对于软件系统需求的正确性评价, 应是需求验证的主要工作, 也非ATAM 所关注的内容。集成测试是在软件开发的测试阶段需要完成的任务, 此时, 架构设计、架构评审 (即用ATAM, SAAM 进行软件架构评审)、软件详细设计、编码、单元测试工作都已完成, 所以该工作也非ATAM 所关注的内容。只有D 选项的属性优先级排序是ATAM 所要做的。

【2012年下】基于场景的架构分析方法 (Scenarios-based Architecture Analysis Method, SAAM) 是卡耐基梅隆大学软件工程研究所的Kazman 等人于1983年提出的一种非功能质量属性的架构分析方法, 是最早形成文档并得到广泛应用的软件架构分析方法。SAAM 的主要输入是问题描述、() 和架构描述文档, 其分析过程主要包括场景开发、()、单个场景评估、场景交互和总体评估。

A. 问题说明

B.问题建模

C. 需求说明

D.需求建模

A.架构需求

B.架构描述

C. 架构设计

D.架构实现

【参考答案】CB

【模拟题练习】以下() 不是ATAM 方法的主要活动阶段。

A.场景和需求收集

B. 体系结构视图和场景实现

C. 单个场景评估

D.属性模型构造和分析

【参考答案】C

【点拨】ATAM 方法的主要阶段包括: 场景和需求收集、体系结构视图和场景实现、属性模型构造和分析、折中。

【2024年上】在ATAM 评估方法设计之初, 其主要关注的4种质量属性, 分别为()。

A. 性能、安全性、可修改性和可测试性

- B. 性能、安全性、可修改性和可用性
- C. 性能、可修改性、可用性和可测试性
- D. 安全性、可测试性、可用性和可测试性

【参考答案】 B

考点3: 质量效用树

【2024下】在软件架构评估中，（）方法采用效用树这一工具来对质量属性进行分类和优先级排序。效用树的结构包括：（）。

- | | | | |
|--------|--------|--------|--------|
| A.SAEM | B.ATAM | C.SAAM | D.CBAM |
|--------|--------|--------|--------|

- A. 树根——质量属性——属性分类——质量属性场景(叶子节点)
- B. 树根——属性分类——属性描述——质量属性场景(叶子节点)
- C. 树根——质量属性——属性描述——质量属性场景(叶子节点)
- D. 树根——功能需求——需求描述——质量属性场景(叶子节点)

【参考答案】 BA

【点拨】 ATAM 方法采用效用树这一工具来对质量属性进行分类和优先级排序。效用树的结构包括：树根——质量属性——属性分类——质量属性场景(叶子节点)。需要注意的是， ATAM 主要关注4类质量属性：性能、安全性、可修改性和可用性，这是因为这4个质量属性是利益相关者最为关心的。

4. 8构件与中间件

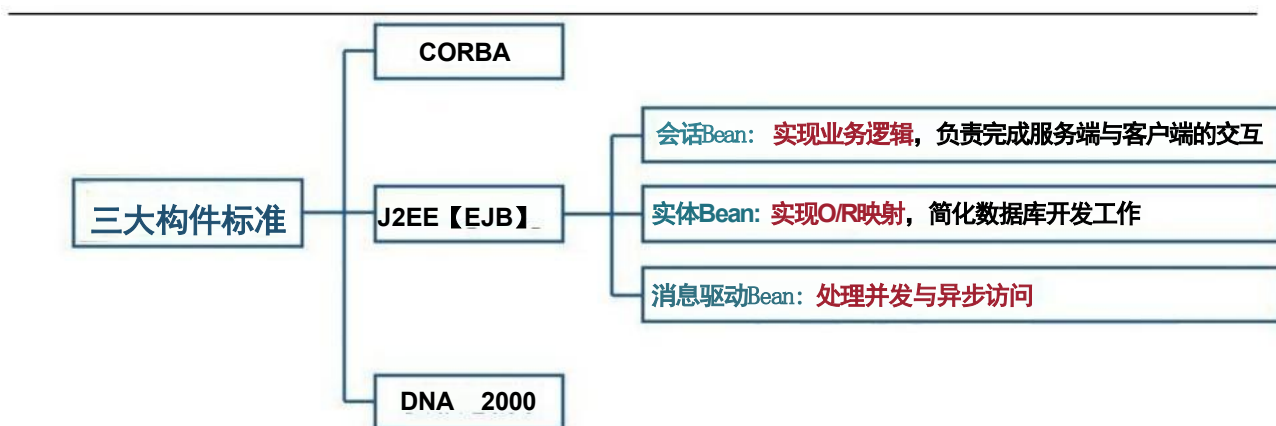
知识点1: 构件

(1) 构件的概念

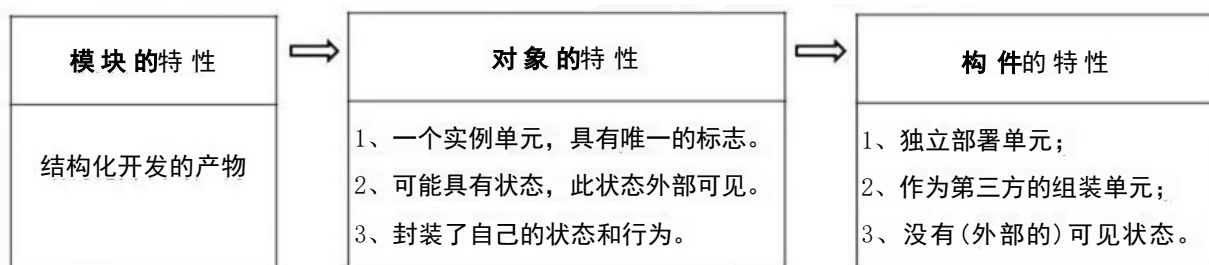
定义1: 软件构件是一种组装单元，它具有规范的接口规约和显式的语境依赖。软件构件可以被独立地部署并由第三方任意地组装。

定义2: 构件是某系统中有价值的、几乎独立的并可替换的一个部分，它在良好定义的体系结构语境境内满足某清晰的功能。

定义3: 构件是一个独立发布的功能部分，可以通过其接口访问它的服务。



(2) 构件、对象、模块的对比



(3) 构件的复用

软件复用【重用】是多次不同的软件开发过程中重复使用相同或相似【软件元素】的过程。软件元素包括：需求分析文档、设计过程、设计文档、程序代码、测试用例、领域知识。

复用的历史发展线路

框架【多个类的组合】



复用的维度

水平复用：不分行业领域，通用

垂直复用：分行业领域，专用

检索与提取构件

理解与评价构件

修改构件

组装构件

a. 检索与提取构件

基于关键字的检索。系统在图形用户界面上将构件库的关键字树形结构直观地展示给用户，复用者通过对树形结构的逐级浏览，寻找需要的关键字并提取相应的构件。（特点：树形或有向无回路图结构）

剖面检索法。该方法基于剖面分类法，由三步构成，分别是构造查询、检索构件和对构件进行排序。这种方法的优点是它易于实现相似构件的查找，但复用者在构造查询时比较麻烦。（特点：利用 Facet 描述构件执行的功能、被操作的数据、构件应用的语境或任意其他特征。分多个剖面，例如：应用领域、使用环境、功能）

超文本检索法。复用者首先给出一个或数个关键字，系统在构件的说明文档中进行精确或模糊的语法匹配，匹配成功后，向复用者列出相应的构件说明。这种方法的优点是用户界面友好，但在某些情况下复用者难以在超文本浏览过程中正确选取构件。（特点：按照人类的联想思维方式任意跳转到包含相关概念或构件的文档）

b. 理解与评价构件

要复用构件，准确地理解构件至关重要。特别是对构件修改使用时。

为达到目的，必须要求构件的开发过程遵循公共标准。

一般构件库的文档中全面而准确地说明以下内容：构件的功能与行为、相关的领域知识、可适应性约束条件与例外情形、可以预见的修改部分及修改方法。

c. 修改构件

理想状态是直接复用构件库中现成的构件，但大多数情况下，必须对构件进行或多或少的修改，以应对新需求。

为了减少构件修改的工作量，要求开发人员尽量使构件的功能、行为和接口设计更为抽象化、通用化和参数化。这样，复用者即可通过对实参的选取来调整构件的功能或行为。如果这种调整仍不足以使构件适用于新系统，复用者就必须借助设计信息和文档来修改构件。

构件库中若无可修改使用的构件，则按新需求开发构件，并存入构件库。

d. 组装构件

基于功能的组装技术。基于功能的组装技术采用子程序调用和参数传递的方式将构件组装起来。它要求库中的构件以子程序/过程/函数的形式出现，并且接口说明必须清晰。当使用这种组装技术进行软件开发时，开发人员首先要对新系统进行功能分解，将系统分解为强内聚、松耦合的功能模块；然后根据各模块的功能需求提取构件，进行适应性修改后，再挂接在上述功能分解框架中。

基于数据的组装技术。基于数据的组装技术首先根据当前软件问题的核心数据结构设计出一个框架，然后根据框架中各结点的需求提取构件并进行适应性修改，再将构件逐个分配至框架中的适当位置。此后，构件的组装方式仍然是传统的子程序调用与参数传递。这种组装技术也要求库中构件以子程序形式出现，但它所依赖的软件设计方法不再是功能分解，而是面向数据的设计方法，例如，Jackson 系统开发方法。

面向对象的组装技术。由于封装和继承特征，面向对象方法比其他软件开发方法更适合支持软件复用。在面向对象的软件开发方法中，如果从类库中检索出来的基类能够完全满足新系统的需求，则可以直接应用。否则，必须以基类为父类，生成相应的子类，以满足新系统的需求。

e. 在构件组装阶段失配问题

由构件引起的失配，包括由于系统对构件基础设施、构件控制模型和构件数据模型的假设存在冲突引起的失配；

由连接子引起的失配，包括由于系统对构件交互协议、连接子数据模型的假设存在冲突引起的失配；

由于系统成分对全局体系结构的假设存在冲突引起的失配等。要解决失配问题，首先需要检测出失配问题，并在此基础上通过适当的手段消除检测出的失配问题。

考点1: 构件的概念

【2018年下】 EJB 是企业级 Java 构件，用于开发和部署多层结构的、分布式的、面向对象的Java应用系统。其中，（）负责完成服务端与客户端的交互；（）用于数据持久化来简化数据库开发工作；（）主要用来处理并发和异步访问操作。

- | | | | |
|----------|----------|-----------|-----------|
| A. 会话型构件 | B. 实体型构件 | C. COM 构件 | D. 消息驱动构件 |
| A. 会话型构件 | B. 实体型构件 | C. COM 构件 | D. 消息驱动构件 |
| A. 会话型构件 | B. 实体型构件 | C. COM 构件 | D. 消息驱动构件 |

【参考答案】 ABD

考点2: 构件的特性

【2022年下】 有关构件特性的描述中，说法不正确的是（）。

- A. 构件是独立部署单元
- B. 构件可作为第三方的组装单元
- C. 构件没有外部的可见状态
- D. 构件作为部署单元，是可拆分的

【参考答案】 D

【. 点拨】 构件的特性：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。构件的特性只有以上3点， D 选项说法不正确，所以答案选D。

【2016年下】 关于构件的描述，正确的是（）。

- A. 构件包含了一组需要同时部署的原子构件

-
- B. 构件可以单独部署，原子构件不能被单独部署
 - C. 一个原子构件可以同时多个构件家族中共享
 - D. 一个模块可以看作带有单独资源的原子构件

【参考答案】A

【点拨】构件是一组通常需要同时部署的原子构件。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。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。

考点3: 构件的复用-检索与提取构件

【2019年下】基于构件的软件开发中，构件分类方法可以归纳为三大类：()根据领域分析的结果将应用领域的概念按照从抽象到具体的顺序逐次分解为树形或有向无回路图结构；()利用 Facet 描述构件执行的功能、被操作的数据、构件应用的语境或任意其他特征；()使得检索者在阅读文档过程中可以按照人类的联想思维方式任意跳转到包含相关概念或构件的文档。

- | | | | |
|----------|----------|----------|----------|
| A.关键字分类法 | B. 刻面分类法 | C. 语义匹配法 | D. 超文本方法 |
| A.关键字分类法 | B.刻面分类法 | C. 语义匹配法 | D.超文本方法 |
| A.关键字分类法 | B. 刻面分类法 | C. 语义匹配法 | D. 超文本方法 |

【参考答案】ABD

考点4: 构件的复用-组装构件

【2019年下】构件组装是指将库中的构件经适当修改后相互连接构成新的目标软件。()不属于构件组装技术。

- A. 基于功能的构件组装技术
- B. 基于数据的构件组装技术
- C. 基于实现的构件组装技术
- D. 面向对象的构件组装技术

【参考答案】C

【. 点拨】构件组装是指将构件库中的构件经过适当修改后相互连接，或者将它们与当前开发项目中的构件元素相连接，最终构成新的目标软件。

构件组装技术大致可分为基于功能的组装技术、基于数据的组装技术和面向对象的组装技术。本题C选项不属于构件组装技术。

考点5: 构件的复用- 失配问题

【2014年下】在构件组装过程中需要检测并解决架构失配问题。其中()失配主要包括由于系统对构件基础设施、控制模型和数据模型的假设存在冲突引起的失配。()失配包括由于系统对构件交互协议、构件连接时数据格式的假设存在冲突引起的失配。

- | | | | |
|-------|-------|-------|--------|
| A. 构件 | B. 模型 | C. 协议 | D. 连接子 |
| A. 构件 | B. 模型 | C. 协议 | D. 连接子 |

【参考答案】 AD

知识点2: 中间件



(1) 概念

中间件：中间件是一种独立的系统软件或服务程序，可以帮助分布式应用软件在不同的技术之间共享资源。

中间件是一类构件，中间件是一类系统软件。

(2) 采用中间件技术的优点

(简化结构、屏蔽差异、利于复用)

面向需求。即设计师集中精力于业务逻辑本身。

业务的分隔和包容性。应用开发人员可以按照不同的业务进行功能的划分，体现为不同的接口或交互模式。

设计与实现隔离。构件对外发生作用或构件间的交互，都是通过接口进行的，构件使用者只需要知道构件的接口，而不必关心其内部实现，这是设计与实现分离的关键。

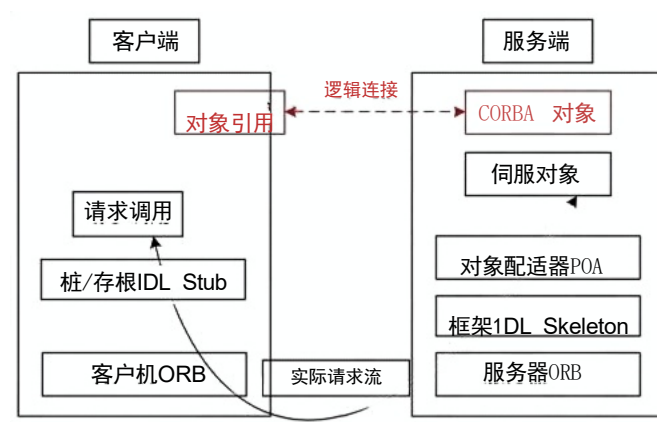
隔离复杂的系统资源。架构很重要的一个功能就是将系统资源与应用构件隔离，这是保证构件可复用甚至“即插即用”的基础，与中间件的意图也是一致的。

符合标准的交互模型。中间件则实现了架构的模型，实现了标准的协议。

软件复用。中间件提供了构件封装、交互规则、与环境的隔离等机制，这些都为软件复用提供了方便解决方案。

提供对应用构件的管理。基于中间件的软件可以方便地进行管理，因为构件总可以通过标识机制进行划分。

(3) 中间件应用-Corba（公共对象请求代理体系结构）（代理模式）



伺服对象 (Servant) :CORBA 对象的真正实现，负责完成客户端请求。

对象适配器 (Object Adapter):用于屏蔽ORB内核的实现细节，为服务器对象的实现者提供抽象接口，以便他们使用ORB 内部的某些功能。

对象请求代理 (Object Request Broker) :解释调用并负责查找实现该请求的对象，将参数传给找到的对象，并调用方法返回结果。客户方不需要了解服务对象的位置、通信方式、实现、激活或存储机制。

考点1: 中间件

【2012年下】以下关于软件中间件的叙述，错误的是()。

- A. 中间件通过标准接口实现与应用程序的关联，提供特定功能的服务
- B.使用中间件可以提高应用软件可移植性
- C.使用中间件将增加应用软件设计的复杂度
- D.使用中间件有助于提高开发效率

【参考答案】 C

【. 点拨】中间件是一种独立的系统软件或服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源，中间件位于客户机服务器的操作系统之上，管理计算资源和网络通信。

软件中间件的作用是为处于自己上层的应用软件提供运行与开发的环境，帮助用户开发和集成应用软件。它不仅仅要实现互连，还要实现应用之间的互操作。

【2018年下】CORBA服务端构件模型中，（）是CORBA对象的真正实现，负责完成客户端请求。

- A. 伺服对象 (Servant)
- B. 对象适配器 (Object Adapter)
- C. 对象请求代理 (Object Request Broker)
- D. 适配器激活器 (Adapter Activator)

【参考答案】A

【. 点拨】伺服对象 (Servant): CORBA对象的真正实现，负责完成客户端请求。

对象适配器 (Object Adapter): 用于屏蔽ORB内核的实现细节，为服务器对象的实现者提供抽象接口，以便他们使用ORB内部的某些功能。

对象请求代理 (Object Request Broker): 解释调用并负责查找实现该请求的对象，将参数传给找到的对象，并调用方法返回结果。客户方不需要了解服务对象的位置、通信方式、实现、激活或存储机制。

4.9 MDA(Model Driven Architecture)

知识点1: MDA

Model: 客观事物的抽象表示。

Architecture: 构成系统的部件、连接件及其约束的规约。

Model-Driven: 使用模型完成软件的分析、设计、构建、部署、维护等各开发活动。

MDA起源于分离系统规约和平台实现的思想。

MDA的主要目标: Portability (可移植性)、 interoperability (互通性)、 Reusability (可重用性)

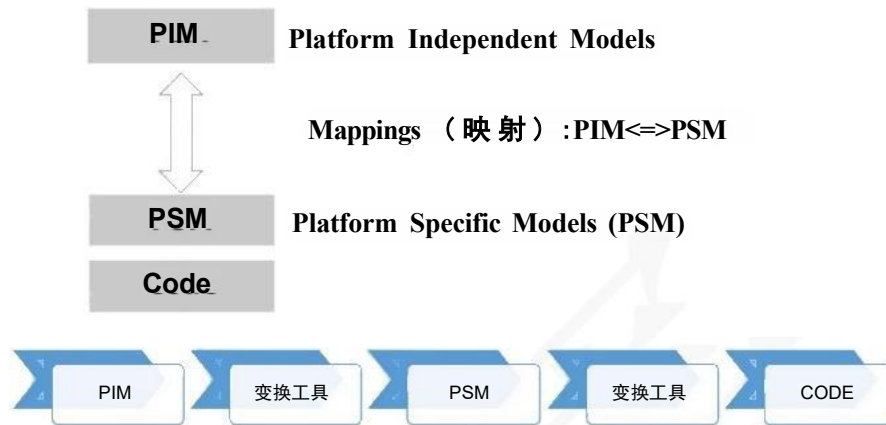
MDA的核心模型:

计算无关模型 (CIM): 对某具体行业内一个项目的业务需求及其系统功能需求进行分析。

平台独立模型 (PIM): 具有高抽象层次、独立于任何实现技术的模型。

平台相关模型 (PSM): 为某种特定实现技术量身定做, 让你用这种技术中可用的实现构造来描述系统的模型。PIM 会被变换成一个或多个 PSM。

代码 Code: 用源代码对系统的描述(规约)。每个 PSM 都将被变换成代码。



考点1: MDA的3种核心模型

【2022年下】使用模型驱动的软件开发方法, 软件系统被表示为一组可以被自动转换为可执行代码的模型。其中, () 在不涉及实现的情况下对软件系统进行建模。

- A.平台无关模型
- B.计算无关模型
- C.平台相关模型
- D.实现相关模型

【参考答案】 A

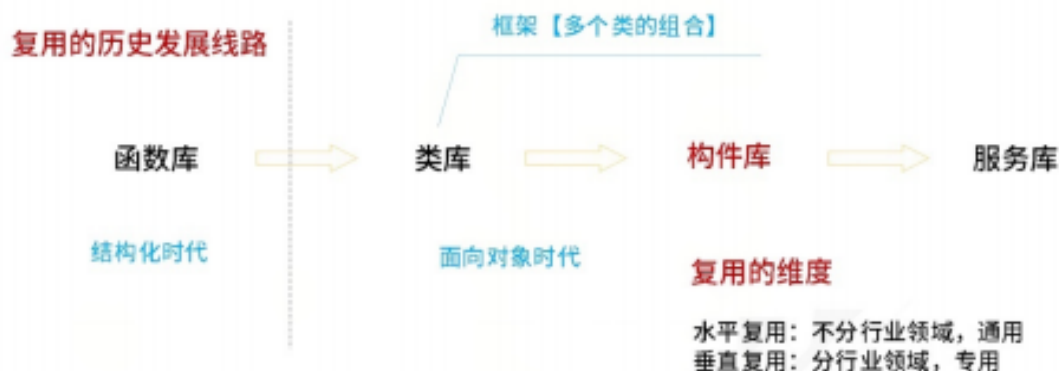
【. 点拨】模型驱动的体系结构是一种关注模型的软件设计和实现方法, 使用了UML 模型的一个子集来描述系统, 其中会创建不同抽象层次上的模型。模型驱动的体系结构 (MDA) 方法建议应当产生以下3种类型的抽象系统模型: 1. 计算无关模型 (Computation Independent Model, CIM)。CIM 对系统中使用的重要的领域抽象进行建模, 因此有时被称为领域模型。2. 平台无关模型 (Platform-Independent Model, PIM)。PIM 在不涉及实现的情况下对系统的运转进行建模。3. 平台相关模型 (Platform-Specific Model, PSM)。PSM是对平台无关模型转换后得到的, 对于每个应用平台都有一个单独的 PSM。 答案选择A 选项。

4. 10软件架构复用

知识点1: 软件架构复用

【软件复用】是一种系统化的软件开发过程，通过识别、分析、分类、获取和修改软件实体，以便在不同软件开发过程中重复使用它们。

软件开发过程中重复使用相同或相似【软件元素】的过程。



软件架构 复用类型	机会复用	开发过程中，只要发现有可复用资产，就对其进行复用
	系统复用	开发之前，要进行规划，以决定哪些需要复用

软件架构复用的基本过程：复用的基本过程主要包括3个阶段：首先构造/获取可复用的软件资产，其次管理这些资产，最后针对特定的需求，从这些资产中选择可复用的部分，以开发满足需求的应用系统。

可复用资产范围：需求、架构设计、元素、建模与分析、测试、项目规划、过程、方法和工具、人员、样本系统、缺陷消除。

考点1: 软件架构的复用类型

【2022年下】在软件架构复用中，()是指开发过程中，只要发现有可复用的资产，就对其进行复用。()是指在开发之前，就要进行规划，以决定哪些需要复用。

- A.发现复用 B.机会复用 C.资产复用 D.过程复用
- A.预期复用 B.计划复用 C.资产复用 D.系统复用

【参考答案】BD

考点2: 软件架构复用的基本过程

【模拟练习题】软件架构复用的基本过程包括获取可复用的软件资产、()和使用可复用资产，其中实现第二个阶段的过程中存在两个关键问题，一是构件分类，二是()。

- A.分析可复用资产 B.存储可复用资产 C.管理可复用资产 D.修改可复用资产
- A.构件检索 B.构件复制 C.构件设计 D.构件实现

【参考答案】 CA

【点拨】复用的基本过程主要包括3个阶段：首先构造/获取可复用的软件资产，其次管理这些资产，最后针对特定的需求，从这些资产中选择可复用的部分，以开发满足需求的应用系统。

1. 复用的前提：获取可复用的软件资产首先需要构造恰当的、可复用的资产，并且这些资产必须是可靠的、可被广泛使用的、易于理解和修改的。2. 管理可复用资产该阶段最重要的是：构件库 (Component Library)，由于对可复用构件进行存储和管理，它是支持软件复用的必要设施。构件库中必须有足量的可复用构件才有意义，构件库应提供的主要功能包括构件的存储、管理、检索以及库的浏览与维护等，以及支持使用者有效地、准确地发现所需的可复用构件。在这个过程中，存在两个关键问题：一是构件分类，构件分类是指将数量众多的构件按照某种特定方式组织起来；二是构件检索，构件检索是指给定几个查询需求，能够快速准确地找到相关构件。3. 使用可复用资产在最后阶段，通过获取需求，检索复用资产库，获取可复用资产，并定制这些可复用资产修改、扩展、配置等，最后将它们组装与集成，形成最终系统。

【2024年上】软件复用的基本过程可以划分为三个阶段，其中，（）阶段主要是构造恰当的、可复用的资产。

- A. 获取可复用资产
- B. 分析可复用资产
- C. 管理可复用资产
- D. 使用可复用资产

【参考答案】 A

第五章 人工智能

5.1 信息安全基础

知识点1: 人工智能

(1) 人工智能概述

定义：人工智能是利用数字计算机或者数字计算机控制的机器模拟、延伸和扩展人的智能，感知环境、获取知识并使用知识获得最佳结果的理论、方法、技术及应用系统。

目标：了解智能的实质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器。【图灵测试】

分类：弱人工智能和强人工智能【是否能真正实现推理、思考和解决问题】

关键技术：自然语言处理、计算机视觉、知识图谱、人机交互、虚拟现实/增强现实、机器学习。

关键技术	描述	说明
自然语言处理 NLP	人与计算机之间用自然语言进行有效通信的各种理论和方法。	机器翻译、语义理解和问答系统。
计算机视觉 Computer Vision	模仿人类视觉系统的科学，让计算机拥有类似人类提取、处理、理解和分析图像以及图像序列的能力，将图像分析任务分解为便于管理的小块任务。	自动驾驶、机器人、智能医疗等领域均需要通过计算机视觉技术从视觉信号中提取并处理信息。近年来随着深度学习的发展，预处理、特征提取与算法处理渐渐融合，形成端到端的人工智能算法技术。
知识图谱 Knowledge Graph	本质上是结构化的语义知识库，是一种由节点和边组成的图数据结构，以符号形式描述物理世界中的概念及其相互关系。知识图谱就是把所有不同种类的信息连接在一起而得到的一个关系网络，提供了从“关系”的角度去分析问题的能力。	知识图谱可用于反欺诈、不一致性验证、组团欺诈等对公共安全保障形成威胁的领域，需要用到异常分析、静态分析、动态分析等数据挖掘方法。知识图谱在搜索引擎、可视化展示和精准营销方面有很大的优势，已成为业界的热门工具。
人机交互 HCI	人机交互主要研究人和计算机之间的信息交换，包括人到计算机和计算机到人的两部分信息交换，是人工智能领域的重要的外围技术。人机交互是与认知心理学、人机工程学、多媒体技术、虚拟现实技术等密切相关的综合学科的交叉。	传统的人与计算机之间的信息交换主要依靠交互设备进行，主要包括键盘、鼠标、操纵杆、数据服装、眼动跟踪器、位置跟踪器、数据手套、压力笔等输入设备，以及打印机、绘图仪、显示器、头盔式显示器、音箱等输出设备。人机交互技术除了传统的基本交互和图形交互外，还包括语音交互、情感交互、体感交互及脑机交互等技术。
虚拟现实/增强现实 VR/AR	虚拟现实或增强现实是以计算机为核心的新型视听技术。结合相关科学技术，在一定范围内生成与真实环境在视觉、听觉等方面高度近似的数字化环境。	用户借助必要的装备与数字化环境中的对象进行交互，相互影响，获得近似真实环境的感受和体验，通过显示设备、跟踪定位设备、触力觉交互设备、数据获取设备、专用芯片等实现。
机器学习	是一门涉及统计学、系统辨识、逼近理	具体来说，机器学习是以数据为基础，通

ML	论、神经网络、优化理论、计算机科学、脑科学等诸多领域的交叉学科。其最初的研究动机是为了让计算机系统具有人的学习能力以便实现人工智能。	过研究样本数据寻找规律，并根据所得规律对未来数据进行预测。目前，机器学习广泛应用于数据挖掘、计算机视觉、自然语言处理、生物特征识别等领域。
	<p>机器学习分类：</p> <p>首先，按照学习模式的不同，机器学习可分为监督学习、无监督学习、半监督学习、强化学习。其中，监督学习需要提供标注的样本集，无监督学习不需要提供标注的样本集，半监督学习需要提供少量标注的样本，而强化学习需要反馈机制。</p> <p>按照学习方法的不同，机器学习可分为传统机器学习和深度学习。区别在于，传统机器学习的领域特征需要手动完成，且需要大量领域专业知识；深度学习不需要人工特征提取，但需要大量的训练数据集以及强大的GPU服务器来提供算力。</p> <p>机器学习的常见算法还包括迁移学习、主动学习和演化学习。</p>	

(2)AI 芯片

人工智能 (Artificial Intelligence, AI) 芯片的定义：从广义上讲只要能够运行人工智能算法的芯片都叫做AI 芯片。但是通常意义上的AI 芯片指的是针对人工智能算法做了特殊加速设计的芯片，现阶段，这些人工智能算法一般以深度学习算法为主，也可以包括其它机器学习算法。

人工智能芯片四大类（按技术架构分类）：

- GPU
- FPGA （现场可编程门阵列）
- ASIC （专用集成电路）
- 类脑芯片

AI 芯片的关键特征：

新型的计算范式： AI 计算既不脱离传统计算，也具有新的计算特质。

训练和推断： AI 系统通常涉及训练和推断过程。

大数据处理能力： 满足高效能机器学习的数据处理要求。

数据精度： 降低精度的设计。

可重构的能力： 针对特定领域而不针对特定应用的设计，可以通过重新配置，适应新的AI 算法、架构和任务。

开发工具： AI 芯片需要软件工具链的支持。

考点1：人工智能概述

【模拟题练习】2020年，在全球抗击疫情的背景下，人工智能被赋予了更多期待和重任，它在信息收集、数据汇总及实时更新、流行病调查、疫苗药物研发、新型基础设施建设等领域大显身手。随着新技术新业态的不断涌现，人工智能凝聚全球智慧、助力全球经济复苏的力量更加凸显，人工智能在我们的生活中无处不在。以下关于人工智能的叙述中，正确的是（）。

A. 人工智能是利用数字计算机或者数字计算机控制的机器模拟、延伸和扩展人的智能，感知环境、获取知识并使用知识获得最佳结果的理论、方法、技术及应用系统。

B. 人工智能的目标是了解智能的实质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器。根据人工智能是否能真正实现推理、思考和解决问题，可以将人工智能分为半人工智能和人工智能。

C. 人工智能关键技术包括程序设计语言处理、计算机视觉、知识图谱、人机交互、虚拟现实/增强现实、机器学习等。

D. 按照学习方法的不同，机器学习可分为监督学习、无监督学习、半监督学习、强化学习。其中，监督学习需要提供标注的样本集，无监督学习不需要提供标注的样本集，半监督学习需要提供少量标注的样本，而强化学习需要反馈机制。

【参考答案】 A

【点拨】人工智能的目标是了解智能的实质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器。根据人工智能是否能真正实现推理、思考和解决问题，可以将人工智能分为弱人工智能和强人工智能，B选项“半人工智能和人工智能”说法是错误的。

人工智能关键技术包括自然语言处理、计算机视觉、知识图谱、人机交互、虚拟现实/增强现实、机器学习等。C选项“程序设计语言处理”说法是错误的。

按照学习模式的不同，机器学习可分为监督学习、无监督学习、半监督学习、强化学习。其中，监督学习需要提供标注的样本集，无监督学习不需要提供标注的样本集，半监督学习需要提供少量标注的样本，而强化学习需要反馈机制。按照学习方法的不同，机器学习可分为传统机器学习和深度学习。区别在于，传统机器学习的领域特征需要手动完成，且需要大量领域专业知识；深度学习不需要人工特征提取，但需要大量的训练数据集以及强大的GPU服务器来提供算力。D选项机器学习分类方法依据错误。

A选项“人工智能是利用数字计算机或者数字计算机控制的机器模拟、延伸和扩展人的智能，感知环境、获取知识并使用知识获得最佳结果的理论、方法、技术及应用系统。”描述正确，本题选择A选项。

考点2：AI芯片

【2022年下】 AI 芯片是当前人工智能技术发展的核心技术，其能力要支持训练和推理，通常， AI 芯片的技术架构包括（）等三种。

- A.GPU、FPGA、ASIC
- B.CPU、FPGA、DSP
- C.GPU、CPU、ASIC
- D.GPU、FPGA、SOC

【参考答案】 A

【2021年下】人工智能技术已成为当前国际科技竞争的核心技术之一，AI 芯片是占据人工智能市场的法宝。 AI 芯片有别于通常处理器芯片，它应具备四种关键特征。（）是AI 芯片的关键特点。

- A. 新型的计算范式、信号处理能力、低精度设计、专用开发工具
- B. 新型的计算范式、训练和推断、大数据处理能力、可重构的能力
- C.训练和推断、大数据处理能力、可定制性，专用开发工具
- D. 训练和推断、低精度设计、新型的计算范式、图像处理能力

【参考答案】 B

知识点2:人工智能应用场景

关键技术	应用场景
自然语言处理 NLP	教育行业：语言学习助手、智能辅导、自动作业批改 医疗行业：医疗问答、智能导诊、病历分析 政府办公系统：智能政策咨询、信访内容自动分类 电商行业：智能客服、评论情感分析 军事系统：语音指挥、多语种情报分析、战报自动生成
计算机视觉 Computer Vision	教育行业：课堂行为监控、在线考试监考、AR教学 医疗行业：CT/MRI影像分析、手术辅助、患者监控 政府办公系统：人脸识别安防、档案数字化、城市监控、证件真伪识别 电商行业：商品图像识别、物流监控 军事系统：卫星图像目标识别、无人机自主导航、战场感知
知识图谱 Knowledge Graph	教育行业：学科知识体系、知识点关系呈现、课程推荐 医疗行业：疾病-基因-药物关系网络构建 政府办公系统：跨部门数据关联分析(如企业信用评估) 电商行业：商品属性关系推理(如搭配推荐) 军事系统：作战知识库构建(地形/装备/敌情)
人机交互 HCI	教育行业：智能白板、语音交互学习系统 医疗行业：智能诊断界面、患者自助系统 政府办公系统：智能政务终端、语音交互系统 电商行业：智能搜索界面、语音购物助手 军事系统：智能指挥界面、语音控制系统
虚拟现实/增强现实 VR/AR	教育行业：虚拟实验室、历史场景沉浸式教学 医疗行业：手术模拟、康复训练 政府办公系统：城市规划模拟、应急演练 电商行业：虚拟商店、AR商品展示、虚拟试穿 军事系统：作战模拟、战术训练
机器学习 ML	教育行业：学习行为预测、个性化学习计划、知识点掌握度分析 医疗行业：慢性病风险预测、药物分子筛选 政府办公系统：舆情风险预警、反欺诈 电商行业：销量预测、动态定价

	军事系统：战场态势预测、装备故障预判
--	--------------------

知识点3: 人工智能技术应用开发

【技术与解决方案】

概念	说明
大模型	如基于Transformer的通用语言模型，能解决多任务语言处理问题 (DeepSeek、ChatGPT)。 用途：对话、翻译、文本生成、代码生成。
提示词	引导模型输出的指令，解决任务定制和输出控制问题。
智能体 (AI Agent) 平台	自主决策系统，解决自动化任务和复杂决策问题Coze、Dify、FastGPT、manus。
MCP	多模态处理技术，解决跨模态理解和复杂场景推理问题 (CLIP、Flamingo)。
模型微调	是指在预训练大模型 (如GPT-4、Llama等) 的基础上，用特定领域或任务的数据进行二次训练，使模型适应新场景的过程。它是迁移学习的核心方法，解决了通用大模型在垂直领域表现不足的问题。
RAG检索增强生成引擎	结合检索与生成，解决知识更新和领域特定问答问题 (RAGFlow)。 用途：建立本地知识库，优先用本地知识库解决问题，以解决AI幻觉问题。