

无忧考培教育学院2019年正版课程

全国计算机技术与软件专业技术资格(水平)考试

信息系统项目管理师

精讲班*04讲-信息化与信息系统（二）

讲师：朱建军（讲师老师）

1. 4软件工程*考点概述

- 1、软件需求层次P36（掌握）、分类P36、需求获取的方法P37、面向结构化方法需求分析模型3层P37（掌握）、需求规格说明书内容P38、需求验证P39
- 2、UML结构3部分P39、UML的6种关系P40和43/14视图P40/5个系统视图P41（掌握）
- 3、面向对象分析P42（掌握）
- 4、软件架构风格5个及特点P45、软件架构评估P45、敏感点和权衡点P45、软件架构评估方式3个P46（掌握）
- 5、结构化设计特点原则P46-47、面向对象设计的思想和7个原则P47、设计模式P47（掌握）
- 6、软件过程管理：连续式和阶段式P48-49
- 7、测试方法P49-50、测试类型P50-51、面向对象测试P51、软件测试和调试的区别P52（掌握）软件测试管理3个工作及相关内容P52-53（掌握）
- 8、软件集成技术分类及特点P53-56（掌握）

举例如下：

□ **业务需求** 反映企业或客户对系统高层次的目标要求
项目投资人、购买产品的客户、客户单位的管理人员、市场营销部门或产品策划部门

□ **用户需求** 描述用户的具体目标，或用户要求系统必须能完成的任务
描述用户能使用系统来做些什么

□ **系统需求** 从系统的角度来说明软件的需求，包括功能需求、非功能需求和设计约束等
也称为行为需求，规定了开发人员必须在系统中实现的软件功能，用户利用这些功能来完成任务，满足业务需要

□ **功能需求** 通过系统特性的描述表现出来的，所谓特性指一组逻辑上相关的需求，表示系统为用户提供某项功能（服务），使用户的业务目标得以满足

□ **非功能需求** 系统必须具备的属性或品质，例如，可维护性、可靠性、效率等
限制条件或补充规约，通常是对系统的一些约束说明

□ **设计约束** 例如，必须采用国有自主知识产权的数据库系统，必须运行在UNIX操作系统之下等

★ 分类1

① **业务需求** 反映企业或客户对系统高层次的目标要求
项目投资人、购买产品的客户、客户单位的管理人员、市场营销部门或产品策划部门

② **用户需求** 描述用户的具体目标，或用户要求系统必须能完成的任务
描述用户能使用系统来做些什么

③ **系统需求** 从系统的角度来说明软件的需求，包括功能需求、非功能需求和设计约束等
也称为行为需求，规定了开发人员必须在系统中实现的软件功能，用户利用这些功能来完成任务，满足业务需要

★ 分类2

① **功能需求** 通过系统特性的描述表现出来的，所谓特性指一组逻辑上相关的需求，表示系统为用户提供某项功能（服务），使用户的业务目标得以满足

② **非功能需求** 系统必须具备的属性或品质，例如，可维护性、可靠性、效率等
限制条件或补充规约，通常是对系统的一些约束说明

③ **设计约束** 例如，必须采用国有自主知识产权的数据库系统，必须运行在UNIX操作系统之下等

分类3

① **常规需求** 用户认为系统应该做到的功能或性能，实现越多用户会越满意

② **期望需求** 用户想当然认为系统应具备的功能或性能，但并不能正确描述自己想要得到的这些功能或性能需求。如果期望需求没有得到实现，会让用户感到不满意

③ **意外需求** 兴奋需求，是用户要求范围外的功能或性能（但通常是软件开发人员很乐意赋予系统的技术特性），实现这些需求用户会更高兴，但不实现也不影响其购买的决策。

特点 ○ 无二义性、完整性、一致性、可测试性、确定性、可跟踪性、正确性、必要性

获取的方法 ○ 用户访谈、问卷调查、采样、情节串联板、联合需求计划

核心是数据字典

★ SA方法 ○

数据模型 ○

实体联系图
(E-R图)

描述实体、属性，以及实体之间的关系

★ 模型 ○

功能模型 ○

数据流图
(DFD)

从数据传递和加工的角度，利用图形符号通过逐层细分描述系统内各个部件的功能和数据在它们之间传递的情况，来说明系统所完成功能

行为模型
(状态模型)

状态转换图
(STD)

通过描述系统的状态和引起系统状态转换的事件，表示系统的行为，指出作为特定事件的结果将执行哪些动作

软件需求规格
说明书 (SRS) ○

需求开发活动的产物，使项目干系人与开发团队对系统的初始规定有一个共同的理解

内容 ○

①范围②引用文件③需求④合格性规定⑤需求可追踪性⑥尚未解决的问题⑦注解⑧附录

①SRS正确地描述了预期的、满足项目干系人需求的系统行为和特征

②SRS中的软件需求是从系统需求、业务规格和其他来源中正确推导而来的

内容 ○

③需求是完整的和高质量的

④需求的表示在所有地方都是一致的

⑤需求为继续进行系统设计、实现和测试提供了足够的基础

★ 需求验证 ○

需求评审 ○ 技术评审

需求测试



练一练

【例题1-15下】 软件需求包括三个不同的层次，分别为业务需求、用户需求和功能及非功能需求。() 属于用户需求；

- A. 反应了组织机构或客户对系统、产品高层次的目标要求，其在项目视图范围文档中予以说明
- B. 描述用户使用产品必须要完成的任务，其在使用实例文档或方案脚本说明中予以说明
- C. 定义了开发人员必须实现的软件功能，使得用户能完成他们的任务，从中满足了业务需求
- D. 软件产品为了满足用户的使用，对用户并发、处理速度、安全性能等方面需求

【例题2-17上】 以下关于需求分析的叙述中，不正确的是()

- A. 需求分析的目的是确定系统必须完成哪些工作，对目标系统提出完整、准确、清晰、具体的要求
- B. 完整的需求分析过程包括：获取用户需求、分析用户需求、编写需求说明书三个过程
- C. 根据项目的复杂程度，需求分析的工作可以由专门的系统分析人员来做，也可以由项目经理带领技术人员完成
- D. 软件需求分为三个层次：业务需求、用户需求、功能需求与非功能需求

练一练

【例题3-18下】软件需求是多层次的，包括业务需求、用户需求、系统需求，其中业务需求（）

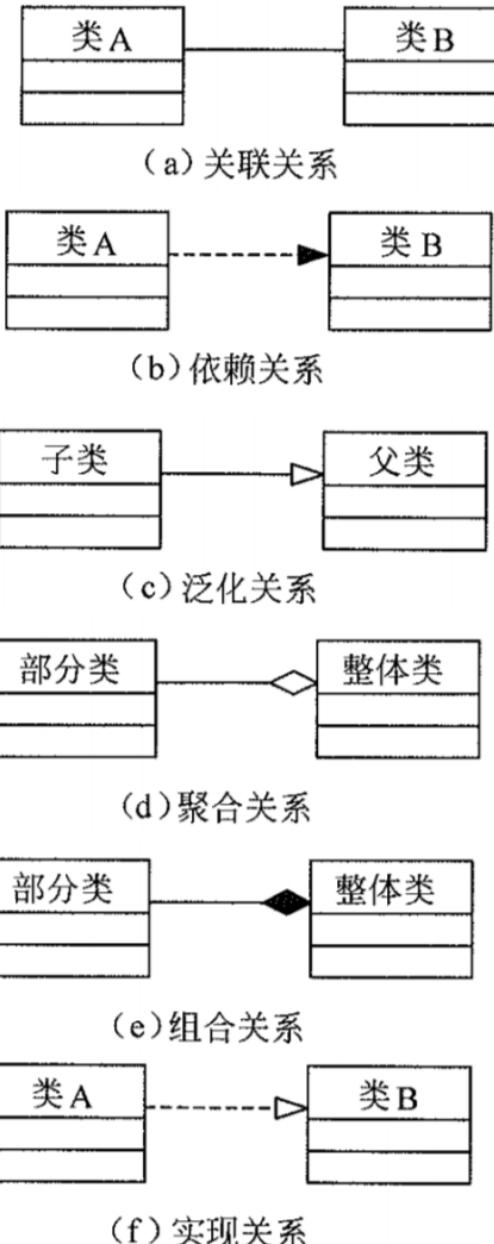
- A. 反应了企业或客户对系统高层次的目标需求
- B. 描述了用户具体目标或者用户要求系统必须完成的任务
- C. 从系统角度来说明软件的需求，包括功能需求、非功能需求和设计约束
- D. 描述了用户任务系统应该具备的功能和性能

1. 4软件工程*UML



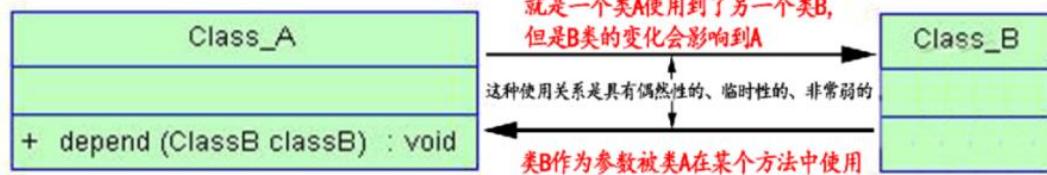
1. 4软件工程*UML

- ①关联  不同类的对象之间的结构关系，它在一段时间内将多个类的实例连接在一起。
- ②依赖  对象实例之间的关系，而不表示两个类之间的关系
- ③泛化  两个类A和B，如果B的变化可能会引起A的变化，则称类A依赖于类B
- 一般事物与该事物中的特殊种类之间的关系，也就是父类与子类之间的关系
- ④共享聚集  继承关系是泛化关系的反关系，也就是说，子类继承了父类，而父类则是子类的泛化
- 聚合关系，它表示类之间的整体与部分的关系
- ⑤组合聚集  “部分”可能同时属于多个“整体”。“部分”与“整体”的生命周期可以不相同。
例：汽车和车轮是聚合关系，车子坏了，车轮还可以用；车轮坏了，可以换一个新的
- 组合关系，它也是表示类之间的整体与部分的关系
- ⑥实现关系  与聚合关系的区别在于，组合关系中的“部分”只能属于一个“整体”，
“部分”与“整体”的生命周期相同，
“部分”随着“整体”的创建而创建，也随着“整体”的消亡而消亡。
- 将说明和实现联系起来。
- 接口是对行为而非实现的说明，而类中则包含了实现的结构。
- 一个或多个类可以实现一个接口，而每个类分别实现接口中的操作



1. 依赖关系：需要借用一条船，此时人与船之间的关系就是依赖。

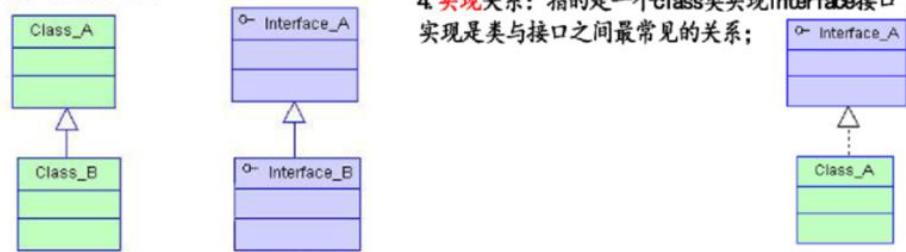
表现在代码层面里：



2. 关联关系：比如我和我的朋友；这种关系比依赖更强、不存在依赖关系的偶然性、关系也不是临时性的，一般是长期性的，而且双方的关系一般是平等的、关联可以是单向、双向的；

3. 泛化关系：一个类或接口继承另外一个类或接口的功能

表现在代码层面里：



4. 实现关系：指的是一个class类实现interface接口（可以是多个）的功能；
实现是类与接口之间最常见的关系；

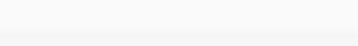
● 依赖(Dependency)



● 泛化 (Generalization)



● 关联 (Association)



组合(Composition)



聚合 (Aggregation)



● 实现 (Realization)



关联	(类之间的结构关系)
依赖	(A 变 B 也变)、
泛化	父类→子类
共享聚合	简称为聚合关系，表示类之间的整体与部分的关系（汽车和轮子就是聚合关系，车子坏了，车轮还可以用；车轮坏了，可以再换一个新的）
组合聚合	简称为组合关系，它也表示类之间的整体与部分的关系，“部分”随着“整体”的创建而创建，也随着“整体”消亡而消亡（一个公司包含多个部门，它们之间的关系就是组合关系，公司一旦倒闭，也就没有部门了）
实现	而每个类分别实现接口中的操作（一个或多个类可以实现一个接口）

□ 关联：连接模型元素及链接实例，用一条实线来表示；

□ 依赖：表示一个元素以某种方式依赖于另一个元素，用一条虚线加箭头来表示；

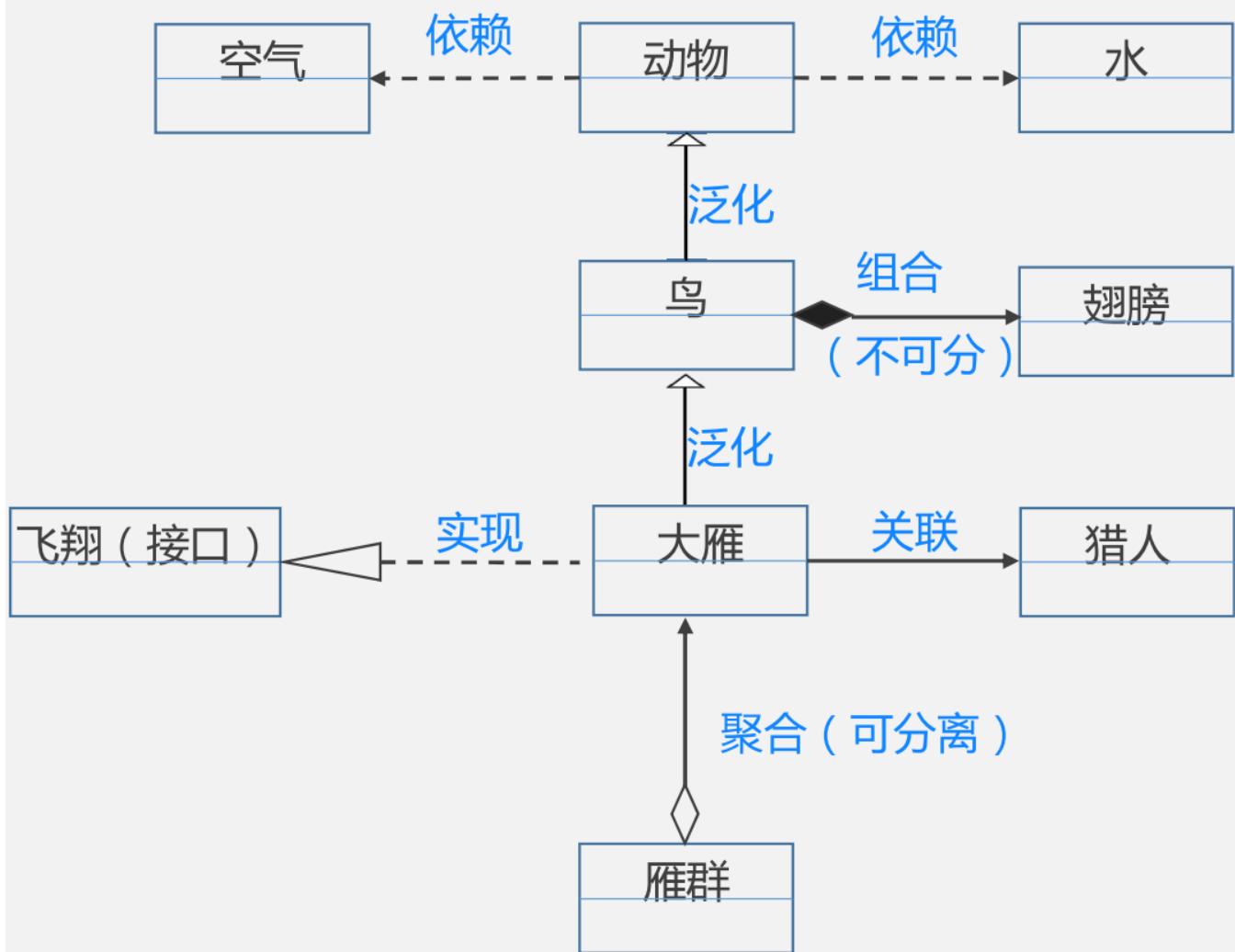
□ 聚合：表示整体与部分的关系，用一条实线加空心菱形来表示；

□ 组成：表示整体与部分的有一关系，用一条实线加实心菱形来表示；

□ 泛化（继承）：表示一般与特殊的关系，用一条实线加空心箭头来表示；

□ 实现：表示类与接口的关系，用一条虚线加空心箭头来表示；

类之间的关系图解



泛化关系：带三角箭头的实线，箭头指向父类。

关联关系：双向的关联可以有两个箭头或者没有箭头，单向的关联有一个箭头。

依赖关系：带箭头的虚线，指向被使用者。

实现关系：带三角箭头的虚线，箭头指向接口。

聚合关系：带空心菱形的实心线，菱形指向整体

组合关系：带实心菱形的实线，菱形指向整体

●实现 (Realization)

几种关系的图形表示

依赖

是一种使用关系，一个类的实现需要另一个类的辅助。这种使用关系一般具有偶然性、临时性、比较弱。

比如一个人要过一条河需要借助一条船，那么人与船的关系就是依赖。再比如人要去旅游，需要用的汽车，那么人和汽车的关系也是一种依赖。

泛化

继承的反关系，描述的是子类继承父类的关系。

实例：龙生九子，九子各不同---龙是父类，九子是子类，九子继承自龙，龙是九子的泛化。

类的几种关系

关联

是一种拥有关系，使得一个类知道另一个类的属性和方法类之间的结构。

双向关联

实例：顾客与产品

单向关联

实例：顾客与地址（购物）

自关联

密码

多重性关联

一个类的一个对象与其他对象的关系描述。

聚合：整体和部分的区别，整体与部分之间可以分离部分脱离整体可单独存在。

实例：车和轮胎，轮胎可以脱离车而单独存在。

组合：整体和部分的区别，整体与部分之间不可以分离，部分脱离整体不可单独存在。

实例：鸟与翅膀，翅膀不能脱离鸟而存在。

实现

一般指的接口和实现接口的类之间的关系。

对象图

描述对象及其关系的图，可以看做类图在某一时刻的实例。

实例：人类是类图，张三、李四、王五对象图。

构件图

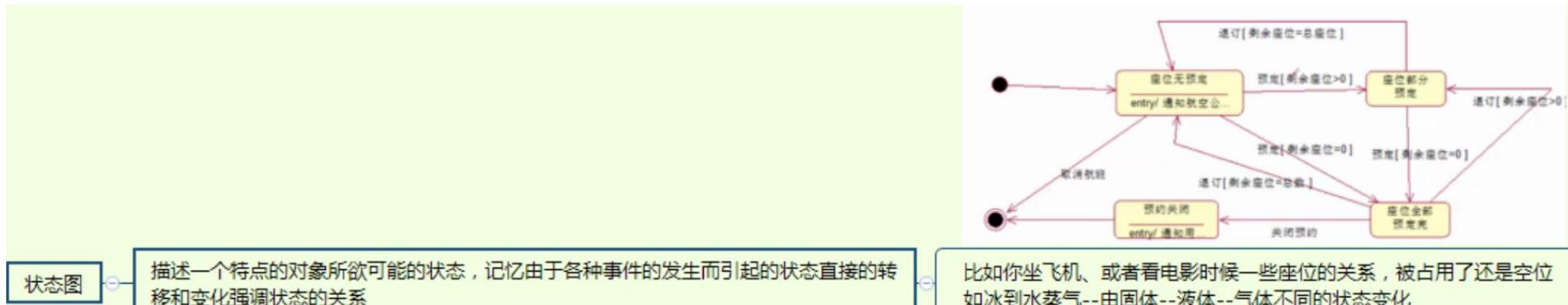
描述的是软件的各种构件（源文件代码、可执行文件或动态链接库DLL）和他们之间的依赖关系。

构件的思想是复用，拿来就用的一些构件及相互关系组成的图。

部署图

显示系统中软件和硬件的物理架构，即拓扑结构。

如：计算机系统与打印机、数据库、应用服务器是怎样连接和部署的。



1. 4软件工程*UML

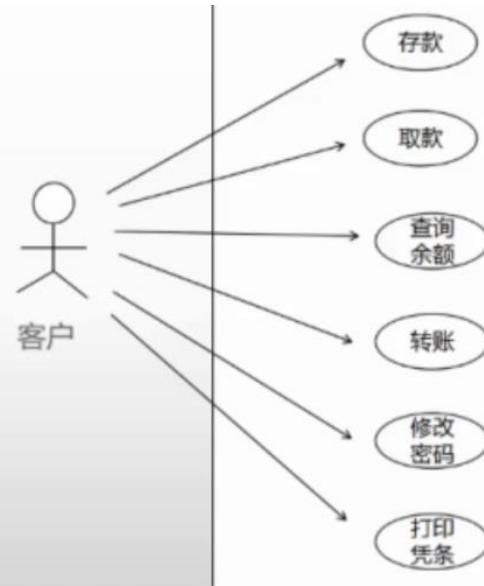
- ★ 14种图
- ①类图 ○ 描述一组类、接口、协作和它们之间的关系，是最常见的图
 - ②对象图 ○ 描述一组对象及它们之间的关系。描述了在类图中所建立的事物实例的静态快照
 - ③构件图 ○ 描述一个封装的类和它的接口、端口，以及由内嵌的构件和连接件构成的内部结构。
 - ④组合结构图 ○ 描述结构化类（例如，构件或类）的内部结构，包括结构化类与系统其余部分的交互点。组合结构图用于画出结构化类的内部内容
 - ⑤用例图 ○ 描述一组用例、参与者及它们之间的关系
 - ⑥顺序图（序列图） ○ 一种交互图，展现了一种交互，由一组对象或参与者以及它们之间可能发送的消息构成
 - ⑦通信图 ○ 一种交互图，它强调收发消息的对象或参与者的结构组织
 - ⑧定时图（计时图） ○ 强调的是时序，强调的是对象之间的组织结构（关系）
 - ⑨状态图 ○ 一种交互图，它强调消息跨越不同对象或参与者的实际时间，而不仅仅只是关心消息的相对顺序
 - ⑩活动图 ○ 描述一个状态机，它由状态、转移、事件和活动组成
 - ⑪部署图 ○ 将进程或其他计算结构展示为计算内部一步一步的控制流和数据流
 - ⑫制品图 ○ 描述对运行时的处理节点及在其中生存的构件的配置
 - ⑬包图 ○ 描述计算机中一个系统的物理结构。制品包括文件、数据库和类似的物理比特集合
 - ⑭交互概览图 ○ 描述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
 - ⑮活动图 ○ 是活动图和顺序图的混合物

祝您必高分各种培训资料需要购买微信：bigaofen

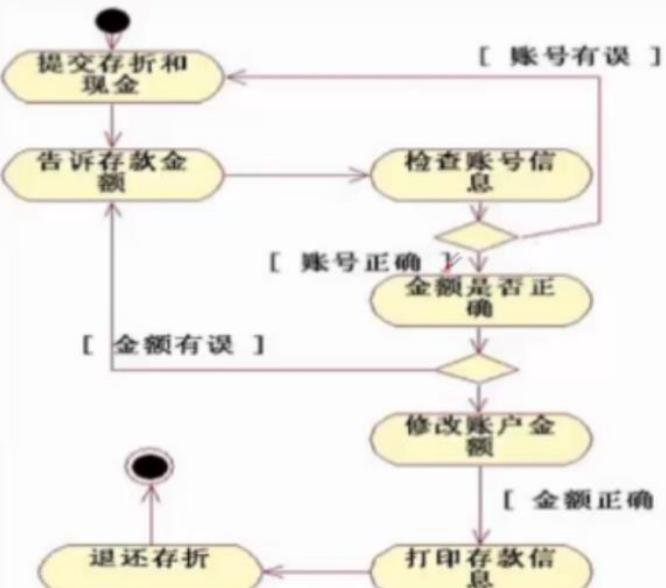
UML2.0包括14种图（10个需要掌握）：

底下的描述也要掌握



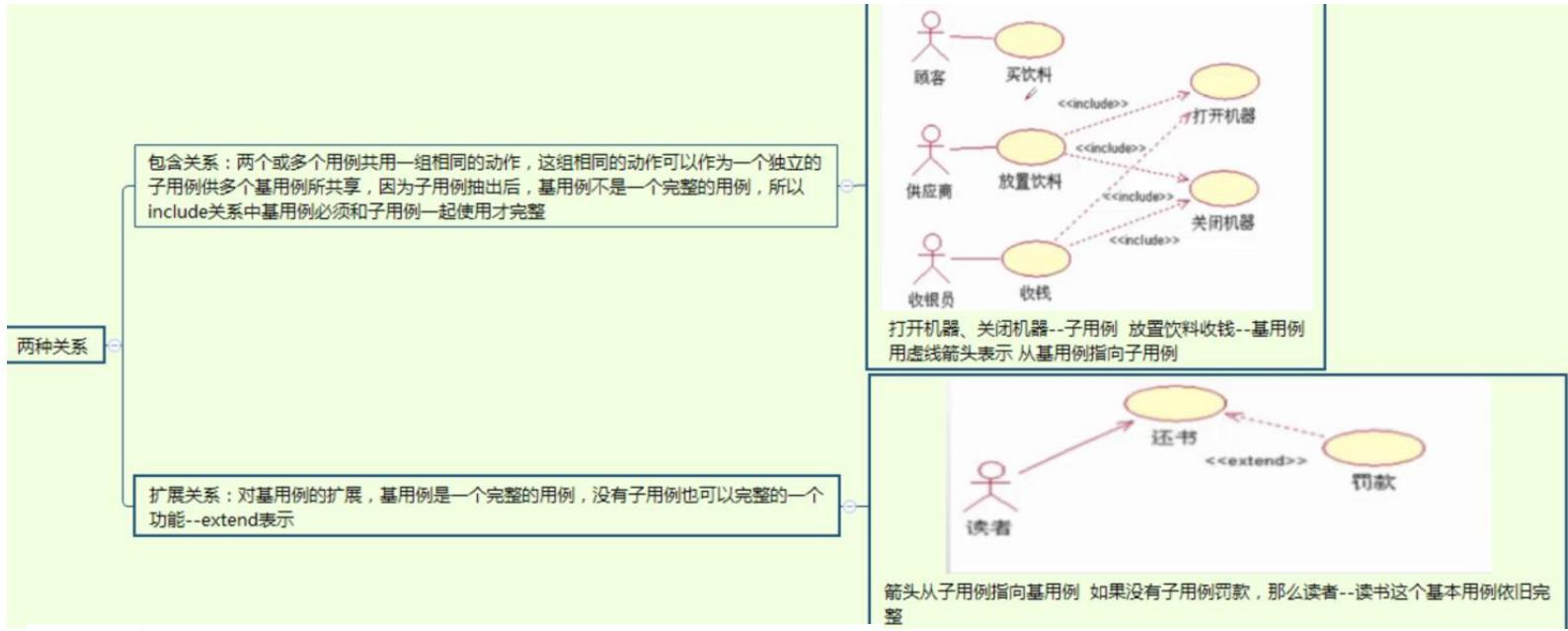


从客户的角度看，atm机有存款、取款、查余额等功能
小人--参与者 椭圆形--用例 箭头表示边界关系



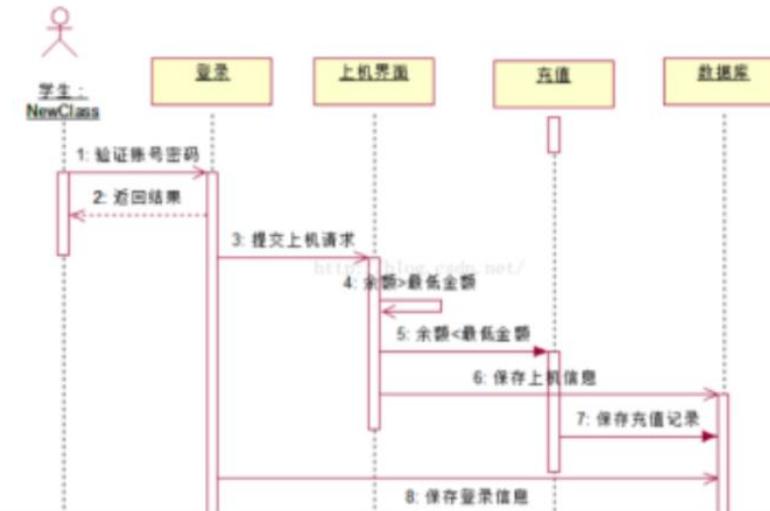
活动图

如你去银行存款的过程活动图

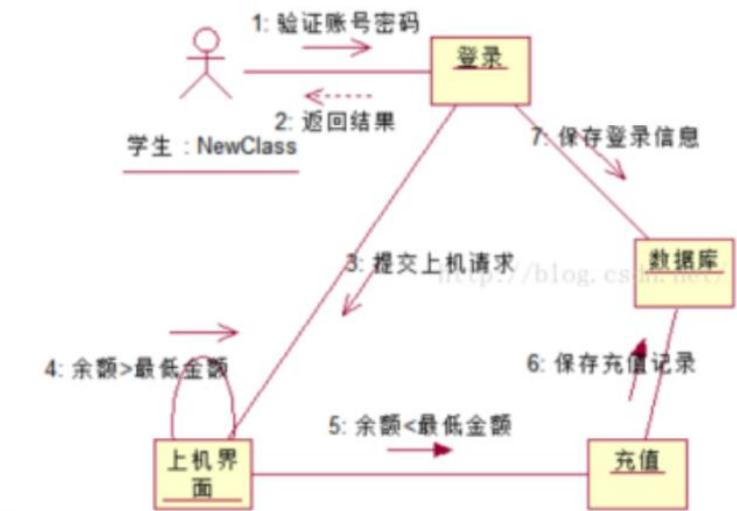


箭头从子用例指向基用例 如果没有子用例罚款，那么读者--读书这个基本用例依旧完整

学生 : NewClass | 登录 | 上机界面 | 充值 | 数据库



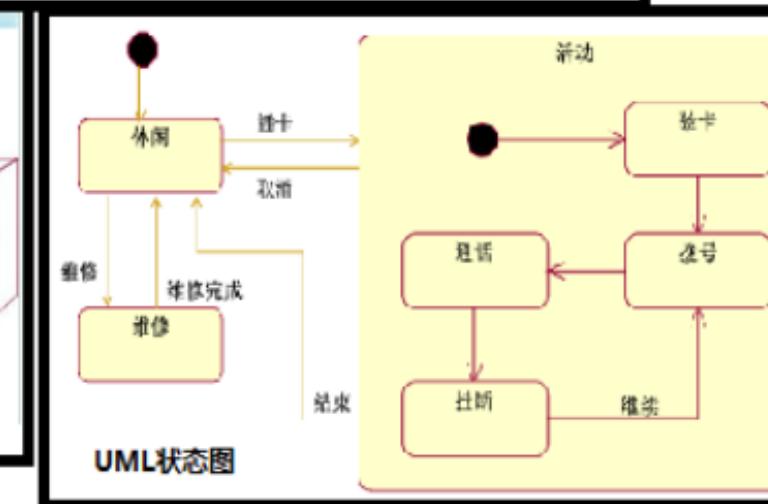
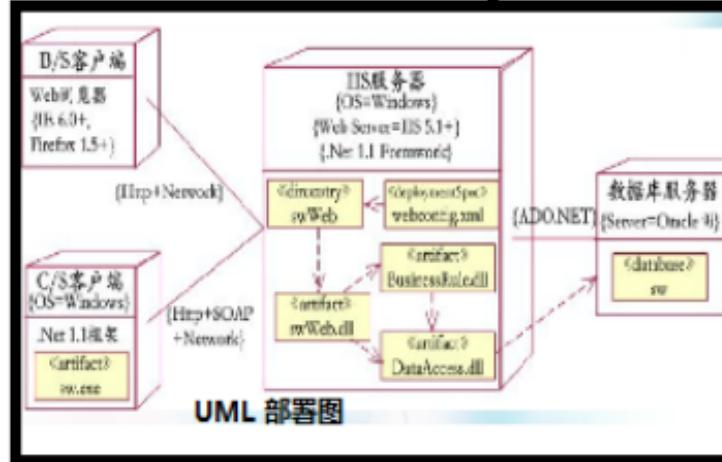
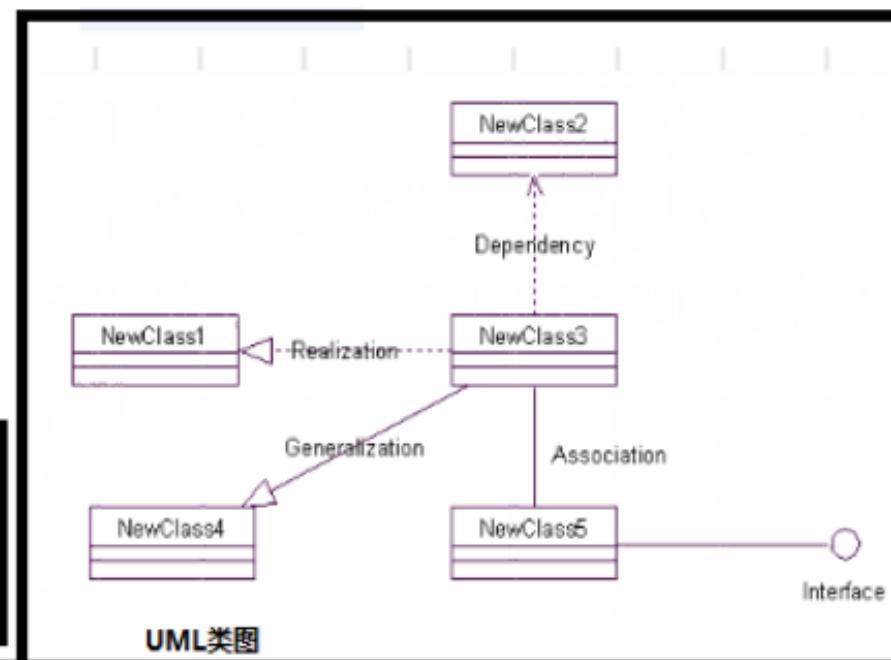
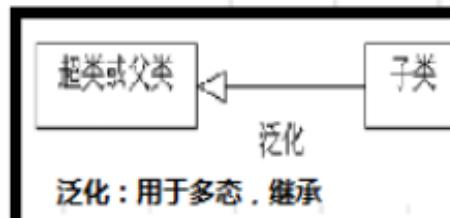
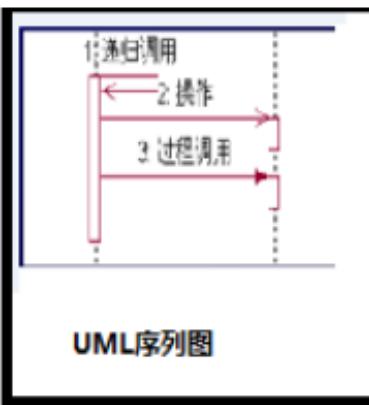
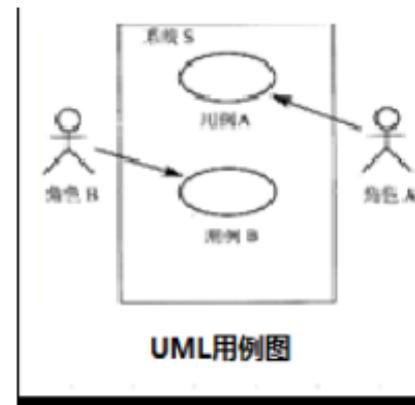
顺序图



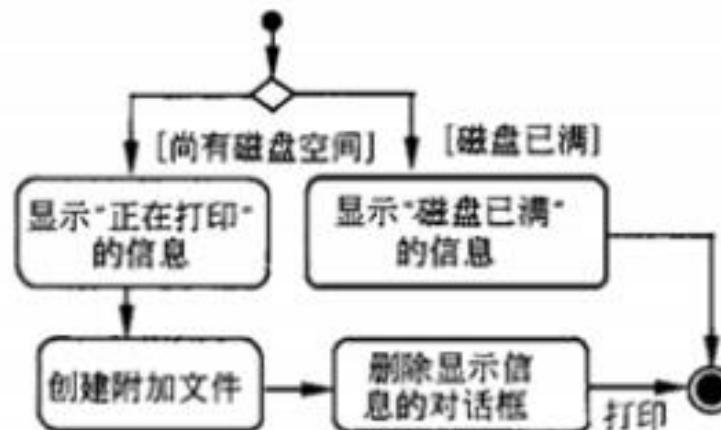
强调了组织结构非线性的操作比如登陆、数据库等的关系

协作图

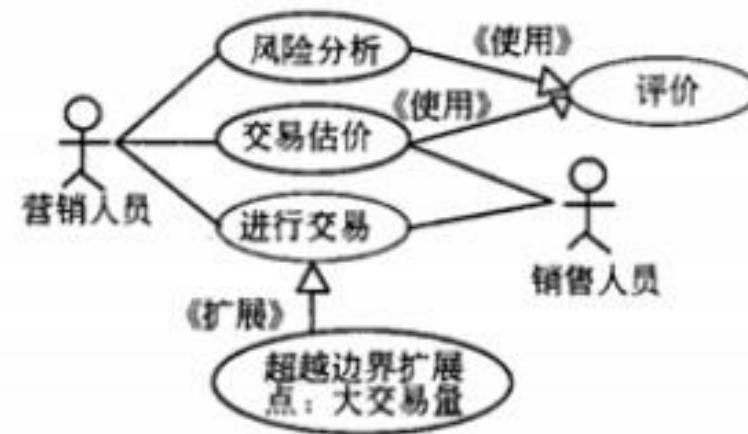
1. 4软件工程*UML



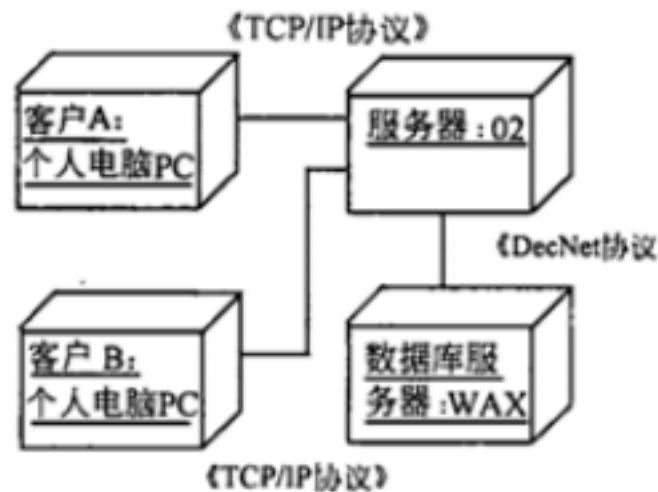
1. 4软件工程*UML



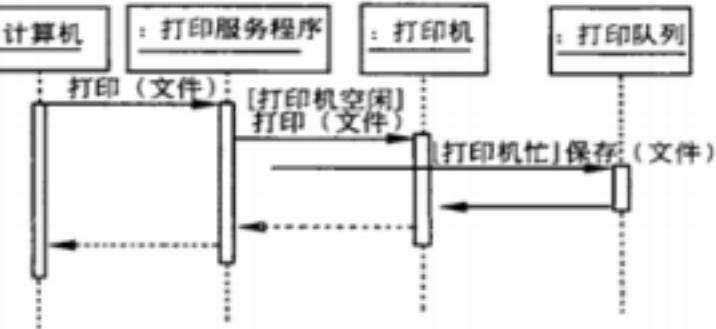
活动图



用例图

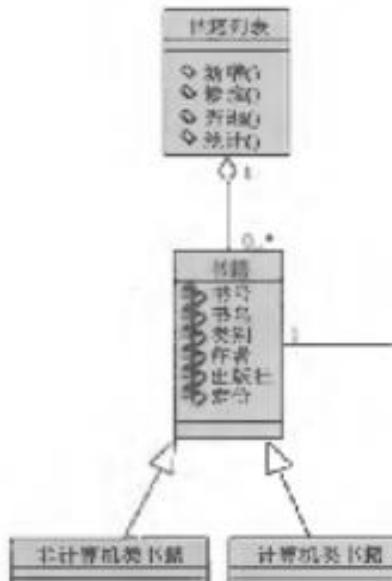


部署图

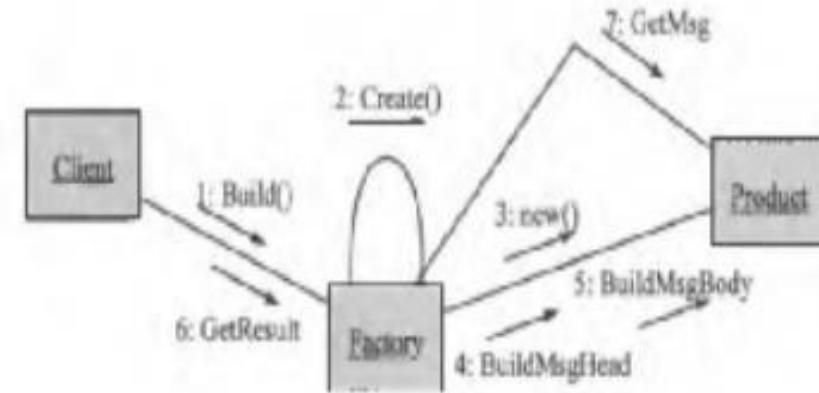


顺序图

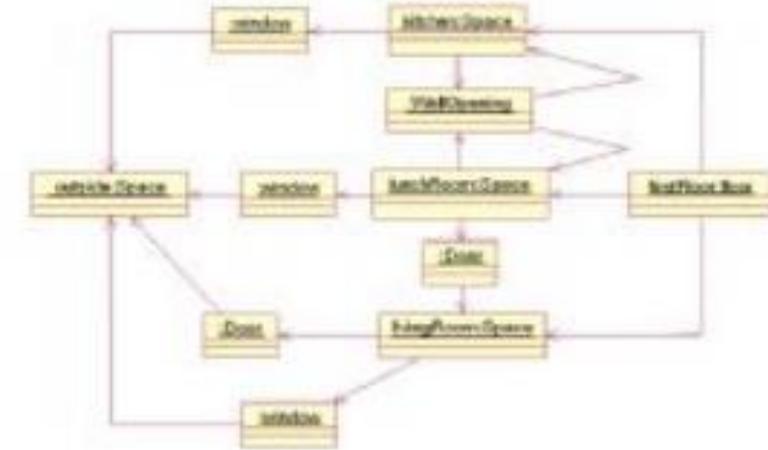
1. 4软件工程*UML



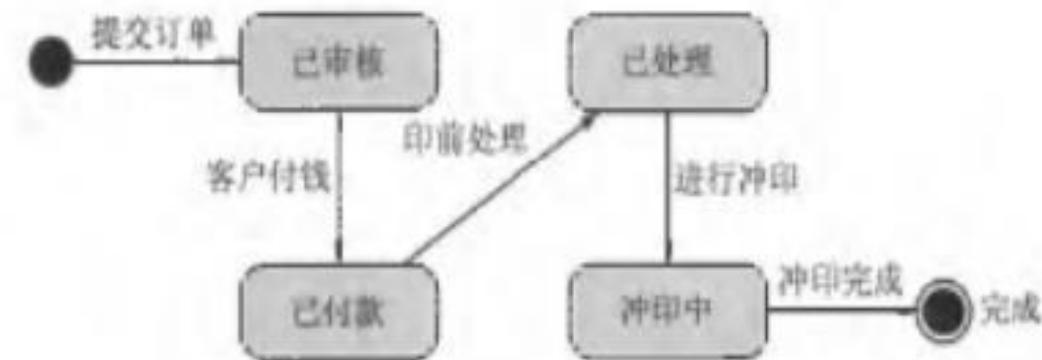
类图



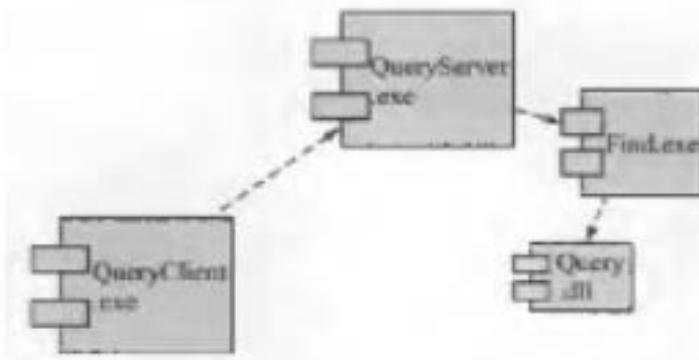
协作图



对象图



状态图



构件图

练一练

【例题4-17下】 UML中共包括14种图，其中（）属于交互图。

- A. 类图
- B. 定时图
- C. 状态图
- D. 对象图

【例题5-17下】（）有称为设计视图，它表示了设计模型中在架构方面具有重要意义的部分，即类、子系统、包和用例实现的子集。

- A. 逻辑视图
- B. 进程视图
- C. 实现视图
- D. 用例视图

【例题6-16上】在用UML对信息系统建模过程中，（）来描述用户需求，主要从用户的角度描述系统的功能。

- A. 用例图
- B. 类图
- C. 对象图
- D. 部署图

【例题7-16下】以下关于UML的叙述中，不正确的是（）。

- A. UML适用于各种开发方法
- B. UML适用于软件生命周期的各个阶段
- C. UML是一种可视化的建模语言
- D. UML也是一种编程语言

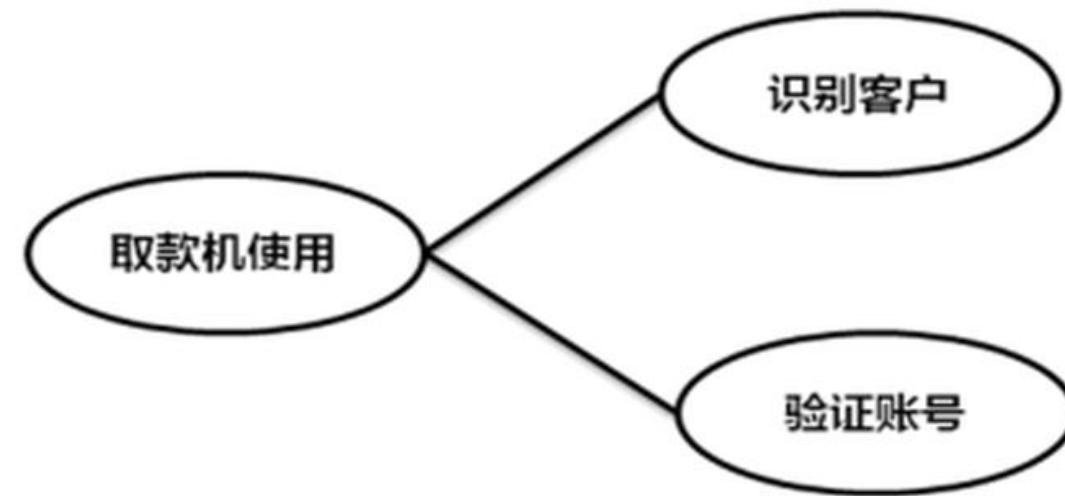
练一练

【例题8-16下】在面向向象系统中，（）关系表示一个较大的“整体”类包含一个或多个“部分”类。

- A. 概化
- B. 合成
- C. 泛化
- D. 聚合

【例题9-17上】面向对象的软件开发过程是用例驱动的，用例是UML的重要部分，用例之间存在着一定的关系，下图表示的是用例之间的（）关系。

- A. 泛化
- B. 包含
- C. 扩展
- D. 等同



练一练

【例题10-18上】关于UML的描述，不正确的是（）

- A. UML是一种可视化编程语言
- B. UML适用于各种软件开发方法
- C. UML用于对软件进行可视化描述
- D. UML适用于软件生命周期的各个阶段

【例题11-18上】UML图不包括（）。

- A. 用例图
- B. 序列图
- C. 组件图
- D. 继承图

【例题12-18下】UML的（）描述了一个特定对象的所有可能状态以及由于各种事件的发生而引起的状态之间的转移。

- A. 控制图
- B. 状态图
- C. 协作图
- D. 序列图

1. 4软件工程*软件架构

核心

能否达到架构级的软件复用

也就是说，能否在不同的系统中，使用同一个软件架构

★ 软件架构风格

① 数据流风格

批处理序列架构风格知管道/过滤器

② 调用/返回风格

主程序/子程序、数据抽象和面向对象，以及层次结构

③ 独立构件风格

进程通信和事件驱动的系统

④ 虚拟机风格

解释器和基于规则的系统

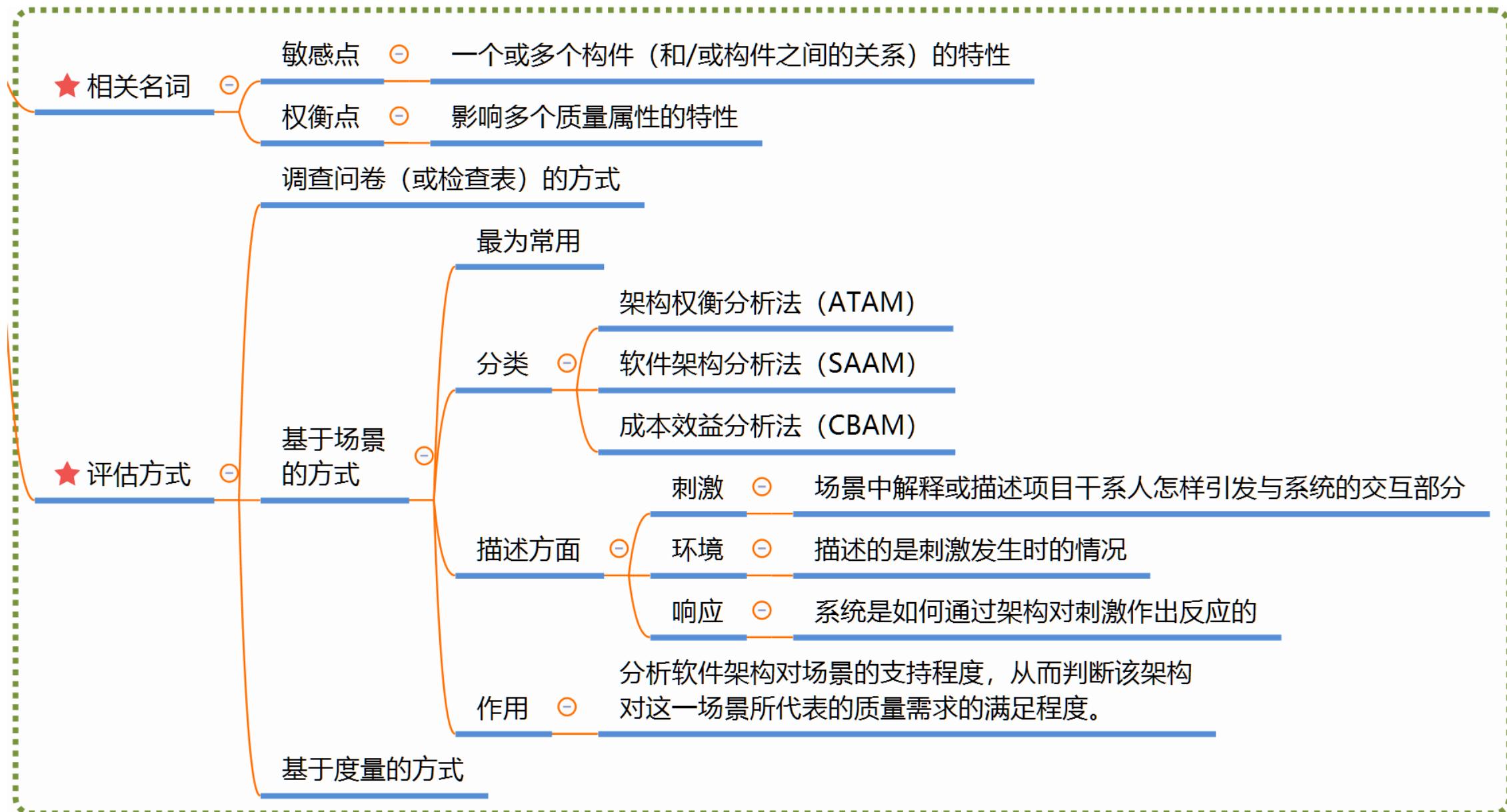
⑤ 仓库风格

数据库系统、黑板系统和超文本系统

★ 评估人员关注

质量属性

1. 4 软件工程*软件架构



练一练

【例题13-17下】研究软件架构的根本目的是解决软件的复用、质量和维护问题，软件架构设计是软件开发过程中关键的一步，因此需要对其进行评估，在这一活动中，评估人员关注的是系统的（）属性。

- A. 功能
- B. 性能
- C. 质量
- D. 安全

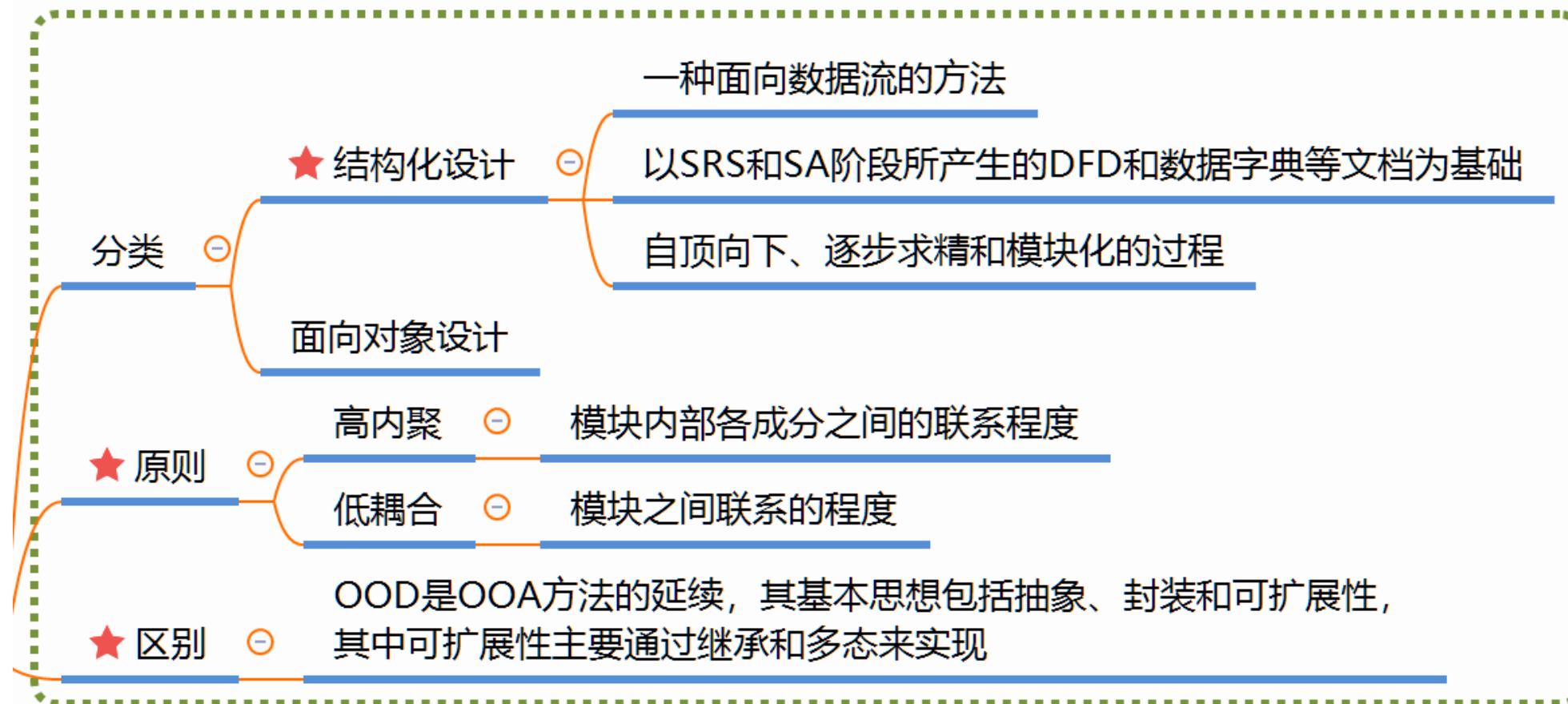
【例题14-11下】用一系列软件的修改来评价软件体系结构的易修改性，属于基于（）的软件体系结构评估方式。

- A. 调查问卷或检查表
- B. 场景
- C. 度量
- D. 实际运行

【例题15-18上】老于是某银行的系统架构师，他为银行投资管理系统设计的软件架构包括进程通信和事件驱动的系统，该软件架构风格属于（）

- A. 数据流风格
- B. 独立构件风格
- C. 仓库风格
- D. 虚拟机风格

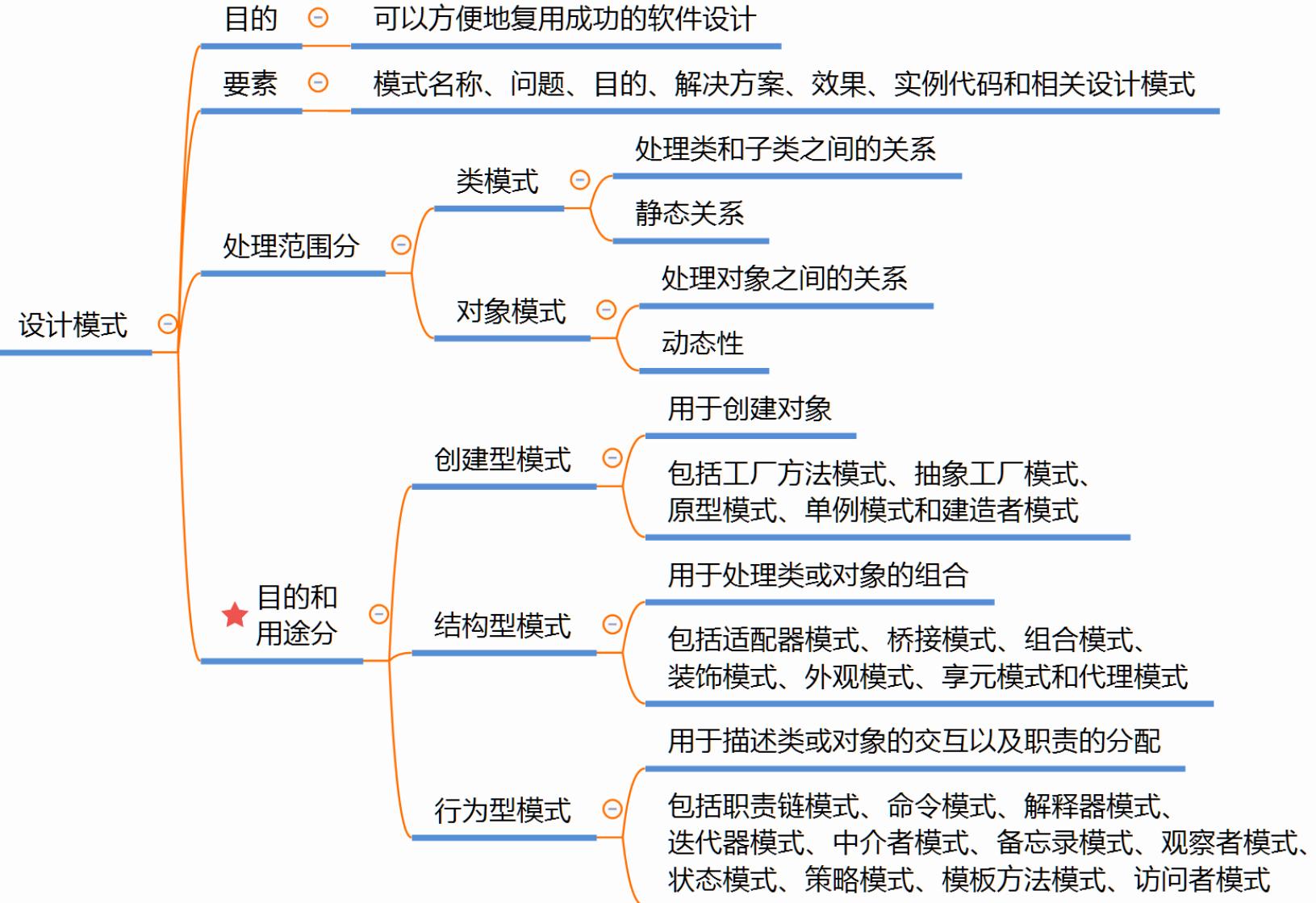
1. 4软件工程*软件设计



【例题16-15上】（）是软件系统结构中各个模块之间相互联系紧密程度的一种度量。

- A. 内聚性
- B. 耦合性
- C. 层次性
- D. 并联性

1. 4软件工程*软件设计

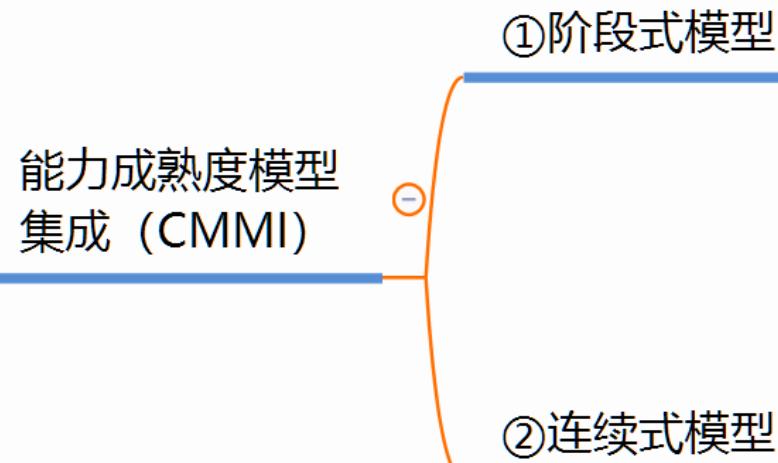


练一练

【例题17-18下】关于设计模式的描述，不正确的是（）。

- A. 设计模式包括模式名称、问题、目的、解决方案、效果、实例代码和相关设计模式等基本要素
- B. 根据处理范围不同，设计模式分为类模式和对象模式
- C. 根据目的和用途不同，设计模式分为创建型模式、结构型模式和行为型模式
- D. 对象模式处理对象之间的关系，这些关系通过继承建立，在编译的时刻就被确定下来，属于静态关系

1.4 软件工程*能力成熟度模型集成



成熟度等级	过程域
可管理级	需求管理、项目计划、配置管理、项目监督与控制、供应商合同管理、度量和分析、过程和产品质量保证
已定义级	需求开发、技术解决方案、产品集成、验证、确认、组织级过程焦点、组织级过程定义、组织级培训、集成项目管理、风险管理、集成化的团队、决策分析和解决方案、组织级集成环境
量化管理级	组织级过程性能、定量项目管理
优化管理级	组织级改革与实施、因果分析和解决方案

连续式分组	过程域
过程管理	组织级过程焦点、组织级过程定义、组织级培训、组织级过程性能、组织级改革与实施
项目管理	项目计划、项目监督与控制、供应商合同管理、集成项目管理、风险管理、集成化的团队、定量项目管理
工程	需求管理、需求开发、技术解决方案、产品集成、验证、确认
支持	配置管理、度量和分析、过程和产品质量保证、决策分析和解决方案、组织级集成环境、因果分析和解决方案

【例题18-18下】CMMI的连续式表示法与阶段式表示法分别表示：（ ）。

- A. 项目的成熟度和组织的过程能力
- B. 组织的过程能力和组织的成熟度
- C. 项目的成熟度和项目的过程能力
- D. 项目的过程能力和组织的成熟度

★ 测试方法分

静态测试

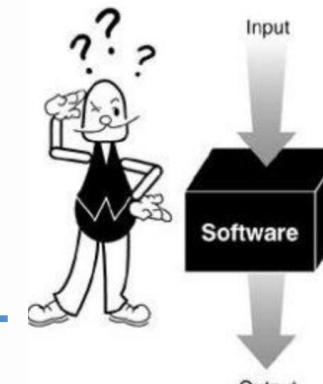
被测试程序不在机器上运行，而采用人工检测和计算机辅助静态分析的手段对程序进行检测

包括对文档的静态测试和对代码的静态测试

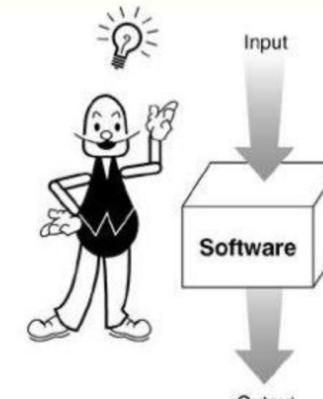
对文档的静态测试主要以检查单的形式进行

对代码的静态测试一般采用桌前检查、代码走查和代码审查

定义 在计算机上实际运行程序进行软件测试



Black-Box Testing



White-Box Testing

动态测试

结构测试

用于软件单元测试中

将程序看作是一个透明的白盒，测试人员完全清楚程序的结构和处理算法，按照程序内部逻辑结构设计测试用例，检测程序中主要执行通路是否都能按预定要求正确工作

方法有控制流测试、数据流测试和程序变异测试

使用人工检查代码的方法来检查代码的逻辑问题

逻辑覆盖 语句覆盖、判定覆盖、条件覆盖、条件/判定覆盖、

条件组合覆盖、修正的条件/判定覆盖和路径覆盖

功能测试

用于集成测试、确认测试和系统测试中

将程序看作是一个不透明的黑盒，完全不考虑程序的内部结构和处理算法，只检查程序功能是否能按照SRS的要求正常使用，程序是否能适当地接收输入数据并产生正确的输出信息，程序运行过程中能否保持外部信息（如文件和数据库等）的完整性等

方法

白盒测试

方法 等价类划分、边界值分析、判定表、因果图、状态图、随机测试、猜错法和正交试验法

1. 4软件工程*软件测试

也称为模块测试，测试的对象是可独立编译或汇编的程序模块、软件构件或OO软件中的类（统称为模块）

单元测试

目的是检查每个模块能否正确地实现设计说明中的功能、性能、接口和其他设计约束等条件，发现模块内可能存在的各种差错

集成测试

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

目的 主要用于验证软件的功能、性能和其他特性是否与用户需求一致

确认测试

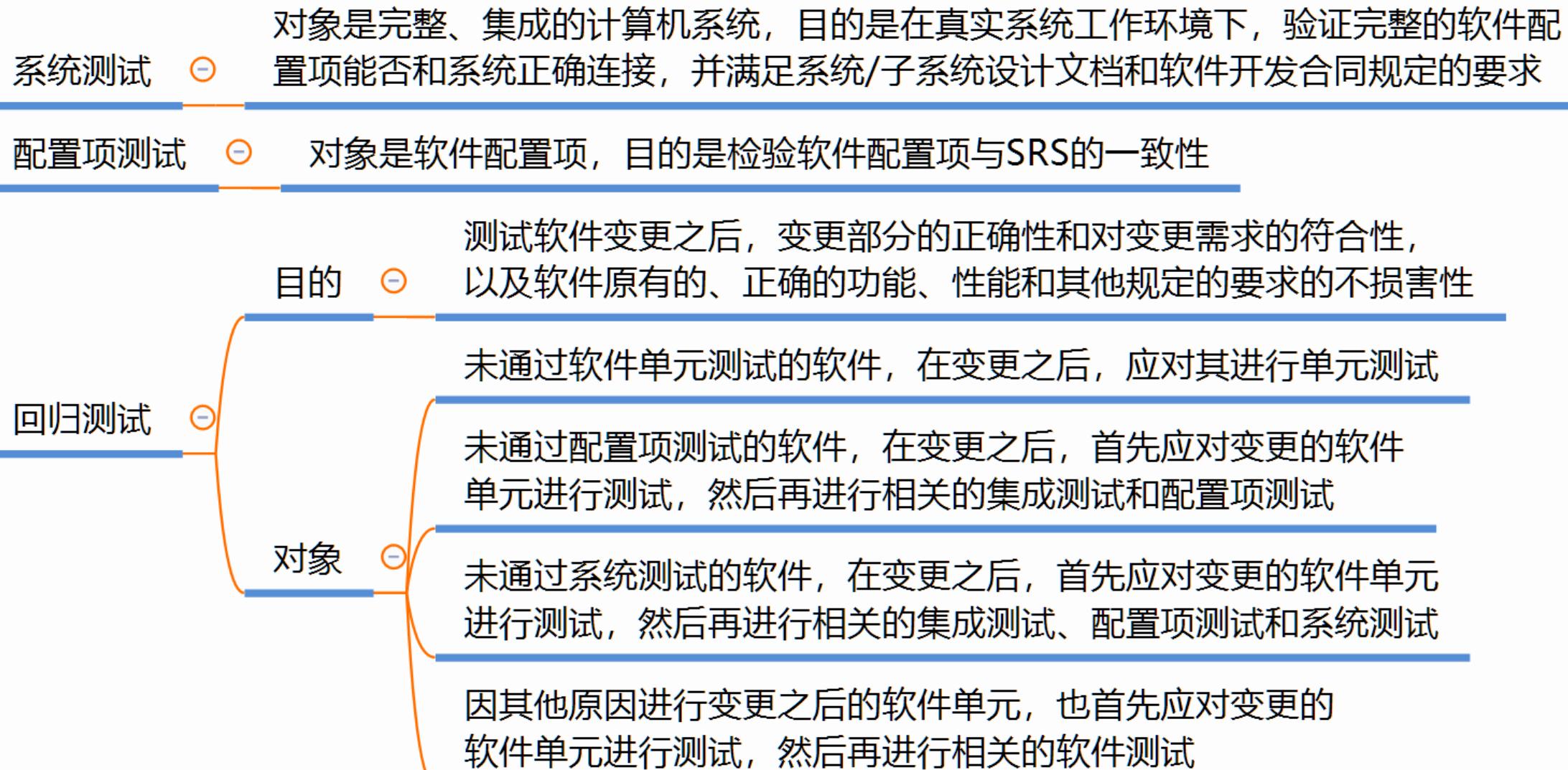
内部确认测试 由软件开发组织内部按照SRS进行测试

分类 Alpha测试 由用户在开发环境下进行测试

Beta测试 由用户在实际使用环境下进行测试

通过后，才能把产品发布或交付给用户

1. 4软件工程*软件测试





补充：

1、软件测试是为了发现错误而执行程序的过程，是根据程序开发阶段的规格说明及程序内部结构而精心设计的一批测试用例（输入数据及其预期结果的集合），并利用这些测试用例去运行程序，以发现程序错误的过程。故软件测试应尽可能在实际运行使用环境下进行。软件测试不再只是一种仅在编码阶段完成后才开始的活动，而是应该包括在整个开发和维护过程中的活动，它本身也是实际产品构造的一个组成部分。

2、软件测试有如下几大原则：

- (1) 软件开发人员即程序员应当避免测试自己的程序（单元测试除外）
- (2) 应尽早地和不断地进行软件测试
- (3) 对测试用例要有正确的态度：第一，测试用例应当由测试输入数据和预期输出结果这两部分组成；第二，在设计测试用例时，不仅要考虑合理的输入条件，更要注意不合理的输入条件。
- (4) 要充分注意软件测试中的群集现象，也可以认为是“**80-20原则**”。不要以为发现几个错误并且解决这些问题之后，就不需要测试了。反而这里是错误群集的地方，对这段程序要重点测试，以提高测试投资的效益。
- (5) 严格执行测试计划，排除测试的随意性，以避免发生疏漏或者重复无效的工作。
- (6) 应当对每一个测试结果进行全面检查。
- (7) 妥善保存测试用例、测试计划、测试报告和最终分析报告，以备回归测试及维护之用。

练一练

【例题19-15下】（）指在软件维护阶段，为了检测由于代码修改而可能引入的错误所进行的测试活动

- A. 回归测试
- B. 修复测试
- C. 集成测试
- D. 冒烟测试

【例题20-17上】以下关于软件测试的描述，不正确的是（）。

- A. 是为评价和改进产品质量进行的活动
- B. 是必须在编码阶段完成后才开始的活动
- C. 是为识别产品的缺陷而进行的活动
- D. 一般分为单元测试、集成测试、系统测试等阶段

练一练

【例题21-18上】软件测试是实现现软件错误（缺陷）的主要手段，软件测试方法可分为静态测试和动态测试，其中（）属于静态测试

- A. 代码走查
- B. 功能测试
- C. 黑盒测试
- D. 白盒测试

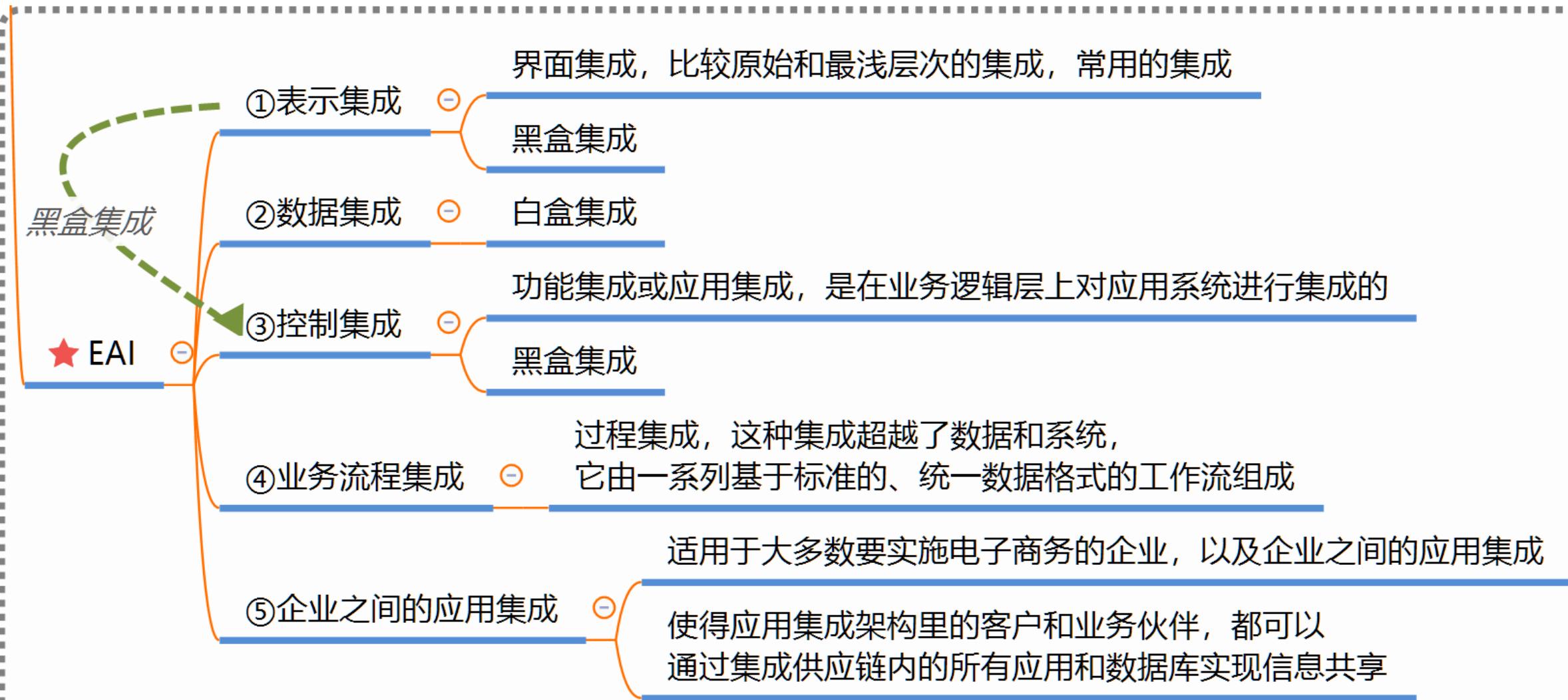
【例题22-18上】结束软件测试工作时，应按照软件配置管理的要求，将（）纳入配置管理。

- A. 全部测试工具
- B. 被测试软件
- C. 测试支持软件
- D. 以上都是

【例题23-18下】软件测试可分为单元测试、集成测试、确认测试、系统测试、配置测试、回归测试等类别。（）主要用于检测软件的功能、性能、和其他特性是否与用户需求一致。

- A. 单元测试
- B. 集成测试
- C. 确认测试
- D. 系统测试

1. 4软件工程*EAI



练一练

【例题24-17下】关于企业应用集成（EAI）技术，描述不正确的是（）

- A. EAI可以实现表示集成、数据集成、控制集成、应用集成等
- B. 表示集成和数据集成是白盒集成，控制集成是黑盒集成
- C. EAI技术适用于大多数实施电子商务的企业以及企业之间的应用集成
- D. 在做数据集成之前必须首先对数据进行标识并编成目录

【例题25-18上】企业应用集成技术（EAI）可以消除信息孤岛，将多个企业信息系统连接起来，实现无缝集成。EAI包括多个层次和方面，其中在业务逻辑层上对应用系统进行黑盒集成的，属于（）

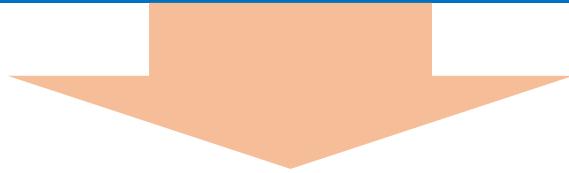
- A. 数据集成
- B. 控制集成
- C. 表示集成
- D. 业务流程集成

参考答案

1	2	3	4	5	6	7	8	9	10
B	B	A	B	A	A	D	D	B	A
11	12	13	14	15	16	17	18	19	20
D	B	C	B	B	B	D	B	A	B
21	22	23	24	25	26	27	28	29	30
A	D	C	B	B					

非常感谢您的聆听

加入正版课程获得VIP全套增值服务



问题咨询联系江山老师 QQ/微信：915446173



江山老师答疑微信



无忧官方公众号



知识分享公众号

扫一扫
加关注
抢先学
早拿证