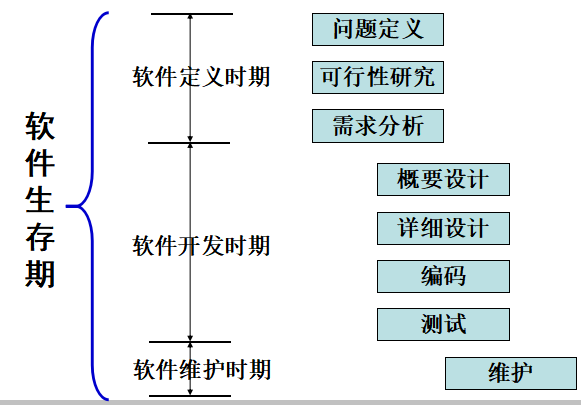
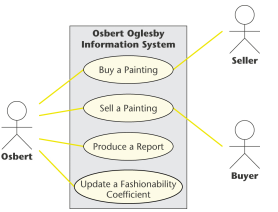
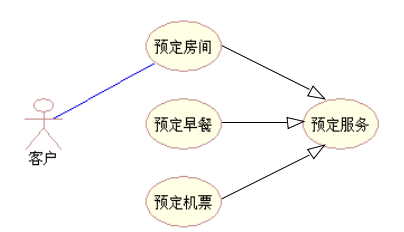
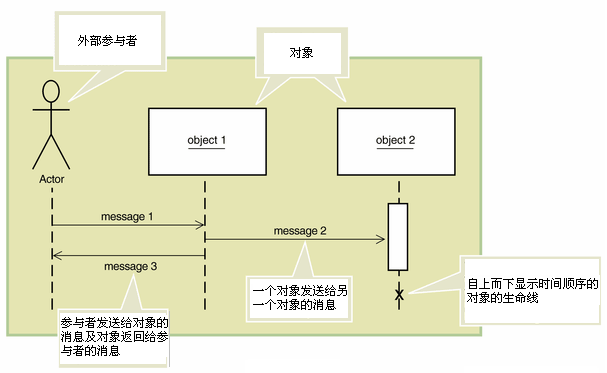
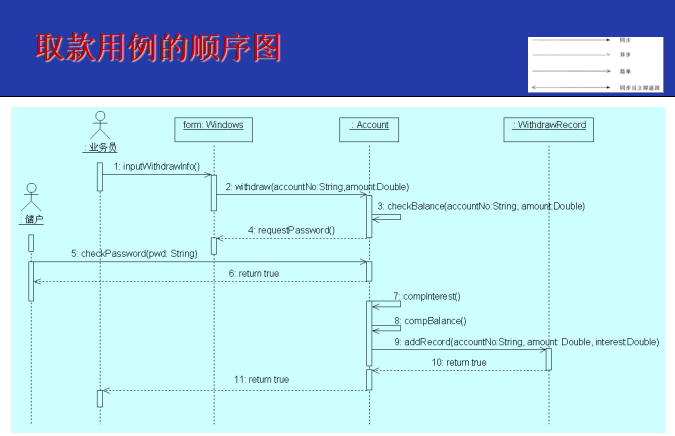
1. 软件7种开发模型及各自特点
   1. 瀑布模型
   2. 快速原型
   3. 增量模型
   4. 螺旋模型
   5. 喷泉模型
   6. 统一过程
   7. 基于构建的开发模型
   8. 敏捷过程
2. 软件生存期：3个
   1. 定义时期
   2. 软件开发时期
   3. 软件维护时期
3. 什么是软件危机？表现？
   1. 软件危机与软件工程：
      1. 20世纪60年代末，由于软件规模扩大、软件复杂性提高导致软件开发开发周期长、成本高、质量差和维护困难，爆发了软件危机，为了克服软件危机，1968年北大西洋公约组织（NATO）提出了软件工程的概念，试图将工程化的方法应用于软件开发。
      2. 许多软件项目不能满足客户的要求
      3. 许多软件项目超出预算和时间安排
   2. 表现：
      1. 对软件开发成本和进度的估计常常很不准确。
      2. 用户对“已完成的”软件系统不满意的现象经常发生。
      3. 软件产品的质量往往靠不住。
      4. 软件常常是不可维护的。
      5. 软件通常没有适当的文档资料。
      6. 软件成本在计算机系统总成本中所占的比例逐年上升。
      7. 软件开发生产率提高的速度，既跟不上硬件的发展速度，也远远跟不上计算机应用迅速普及深入的趋势。
      8. 还有软件开发和维护有关的很多问题
4. 软件组成：
   1. 程序
   2. 文档
   3. 数据
5. 软件生存周期：
   1. 
   2. 软件定义：
      1. 确定总目标和可行性
      2. 导出策略和系统功能
      3. 估计资源和成本
      4. 制定工程进度表
6. 软件工程方法学包括三要素，及其作用
   1. 软件工程**方法**为软件开发提供了 “如何做” 的技术;
   2. 软件**工具**为软件工程方法提供了自动的或半自动的软件支撑环境;
   3. **过程**是为了获得高质量的软件所需要完成的一系列任务框架，它规定了完成各项任务的工作步骤
7. 软件工程的目标？
   1. 运用先进的软件开发技术和管理方法来提高软件的质量和生产率
   2. 也就是要以较短的周期、较低的成本生产出高质量的软件产品，并最终实现软件的工业化生产。
8. 软件质量包括哪些方面？功能性、可靠性、可使用性、效率、可维护性和可移植性
9. 需求分析、概要设计等各阶段在做什么？
   1. 需求分析：
      1. 准确地回答：“目标系统必须做什么”这个问题。
      2. 分析内容：对目标系统提出完整、准确、清晰、具体的要求。对用户提出的需求进行分析并给出详细的定义。
      3. 输出（软件需求说明书、或系统功能说明书、初步的系统用户手册）：提交管理机构评审
   2. 可行性：
      1. 明确回答：“上一个阶段所确定的要解决的问题是否有行得通的解决办法”。
      2. 输出（可行性研究报告）：从技术、经济和社会因素等方面研究论证各方案的可行性。 为决策提供依据。
   3. 概要设计：
      1. 概括地回答：“怎样实现目标系统?”
      2. 设计内容：设计程序的体系结构，即确定程序的组成模块，确定模块间的关系。
      3. 输出文档：软件的概要设计说明书。
   4. 详细设计：
      1. 详细回答：“应怎样具体实现这个系统？”
      2. 设计内容：详细地设计每个模块，确定实现模块功能所需要的算法和数据结构。
      3. 输出文档：软件的详细设计说明书
   5. 程序编码和单元测试：
      1. 实现：正确的、容易理解的、容易维护的程序模块。
      2. 输出文档：源程序、详尽的程序说明和单元测试报告。
   6. 集成测试和系统测试：
      1. 集成测试实现：对模块的组装和连接测试(及相应的调试)，使软件达到预定的要求。
      2. 系统测试实现：根据需求规格说明，逐项测试并确认
      3. 输出文档：测试计划、详细测试方案以及实际测试结果等。
   7. 维护
      1. 改正性维护：诊断和改正在使用过程中发现的软件错误
      2. 适应性维护：修改软件以适应环境的变化
      3. 完善性维护：根据用户的要求改进或扩充软件，使它更完善
      4. 预防性维护：修改软件为将来的维护活动预先做准备
   8. 开发过程中的典型文档
      1. 软件可行性分析报告
      2. 软件需求规格说明书
      3. 项目计划
      4. 软件设计说明书
      5. 软件测试计划
      6. 用户手册
10. 软件需求的3个层次：
11. 业务需求：客户对软件的高层目标要求。
12. 用户需求：用户使用软件必须达到的要求和完成的任务。通常在用例(use case)或方案脚本(scenario)中加以说明。
13. 功能和非功能需求。规定了开发人员必须实现的需求，它的实现将满足上述业务需求和用户需求。通常以需求规格说明(requirement specification)的形式给以详尽描述
14. 面向对象分析中需要创建的3个模型：用例、对象、动态模型
    1. 用例模型：用例和场景表示的功能模型；（是基础和依据）
    2. 对象模型：用类和对象表示的静态模型；（是核心）
    3. 交互模型：由状态图和顺序图表示的动态模型。（涉及交互时很重要 ）
15. 对象模型的5个层次：主题层(也称为范畴层)、类-对象层、结构层、属性层和服务层
16. 对象模型类分类：实体类、边界类、控制类
    1. 实体类表示系统将跟踪的持久信息；
    2. 边界类表示参与者与系统之间的交互；
    3. 控制类负责用例的实现
17. 对象模型中类之间的关系：普通关联（一对一、一对多、多对多）、聚合、泛化、依赖等
18. 能分析业务画出对应的用例图、顺序图、状态图等
19. 顺序图与通信图的联系与区别
    1. 用例图：
       1. 执行者
       2. （由系统边界（一个矩形）封闭的）一组用例
       3. 执行者和用例之间的关联
       4. 用例间关系

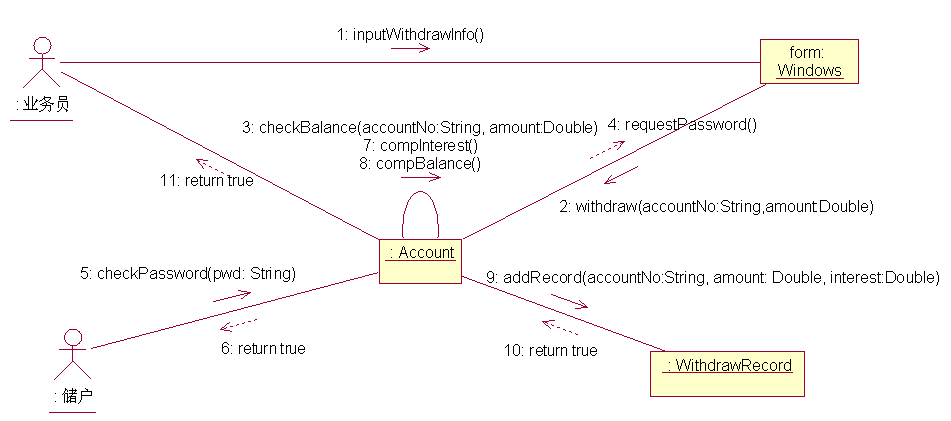
 

* 1. 顺序图：描述对象之间的动态交互关系，着重表现对象间消息传递的时间顺序。

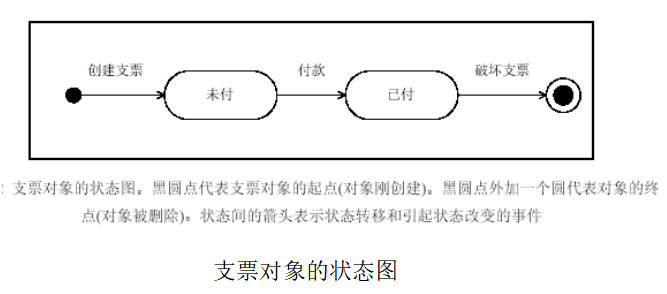




* 1. 通信图是顺序图的一种变化形式，用于描述相互协作的对象间的交互关系和链接关系。

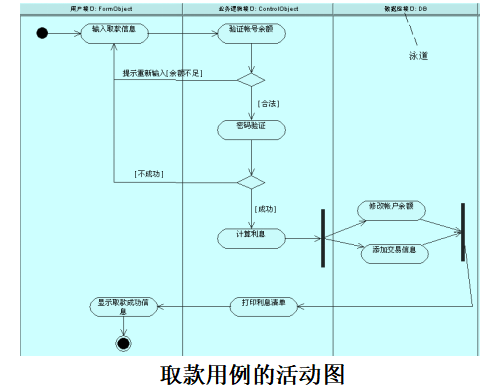


* 1. 状态图：描述一个特定对象的所有可能的状态以及引起状态转换的事件。大多数面向对象技术都用状态图表示单个对象在其生命期中的行为。一个状态图包括一系列状态、事件以及状态之间的转移。

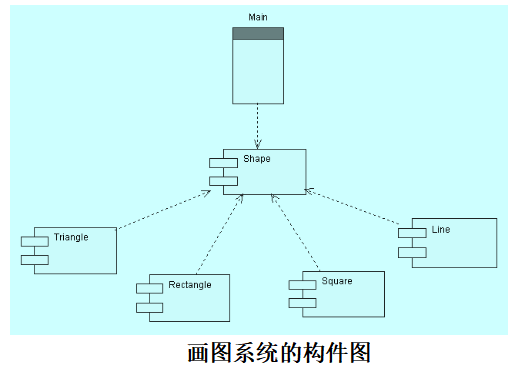


* 1. 活动图用来捕捉用例的活动，使用框图的方式显示动作及其结果

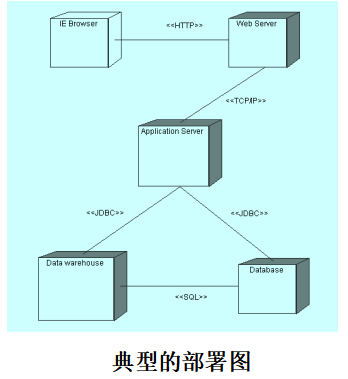
1. 活动图是一个流图，描述了从活动到活动的流。
2. 它是另一种描述交互的方式，它描述采取何种动作，动作的结果是什么(动作状态改变)，何时发生(动作序列)，以及在何处发生(泳道)。



* 1. 构件图描述软件构件及构件之间的依赖关系，显示代码的静态结构。



* 1. 部署图描述处理器、设备和连接，它显示系统硬件的物理拓扑结构及在此结构上执行的软件



1. 软件设计模式：例举2-3个，并说明其用途。23种设计模式
2. 面向对象模块设计准则：

（1）模块化

（2）抽象

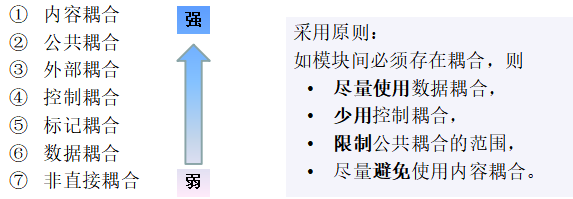
（3）信息隐藏

（4）弱耦合 （可参见教材P80）

（5）强内聚 （可参见教材P82）

（6）可重用

1. 各种内聚及耦合分类
   1. 强内聚 （可参见教材P82）
   2. 弱耦合 （可参见教材P80）
   3. 耦合
      1. 什么是耦合：耦合是软件各模块之间相互关联的度量。它取决于各个模块之间接口的复杂程度、调用模块的方式以及哪些信息通过接口。耦合是影响软件复杂程度和设计质量的一个重要因素。
      2. 耦合可以分为以下几种，它们之间的耦合度由高到低排列：



* + 1. 交互耦合、继承耦合

1. 代码规范问题

规范涉及：版面、注释、标识符命名、变量使用、代码可测性、程序效率、质量保证、代码编译、单元测试、程序版本与维护。

1. 项目管理涉及哪4个方面？

（1）People 人员

（2）Product 产品

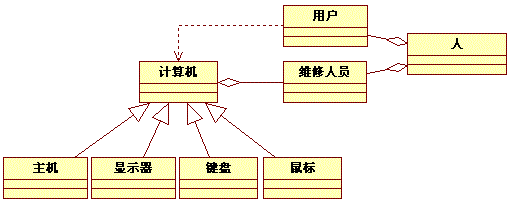
（3）Process 过程

（4）Project 项目

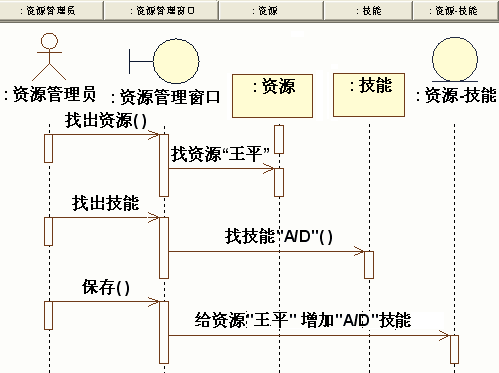
1. 进度管理：甘特、PERT、时标网状图
2. 里程碑

**综合类：**

1. 找出并说明下面类图中的错误。



1. 参考下图，说明“资源管理窗口”和“资源-技能”类至少应包含哪些操作。



资源管理窗口类包含操作：

1. 找出资源（）
2. 找出技能（）
3. 保存（）

资源-技能类包含操作：

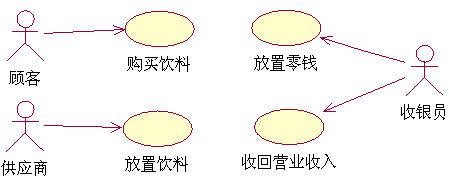
1. 给资源”王平“能加”A/D“技能

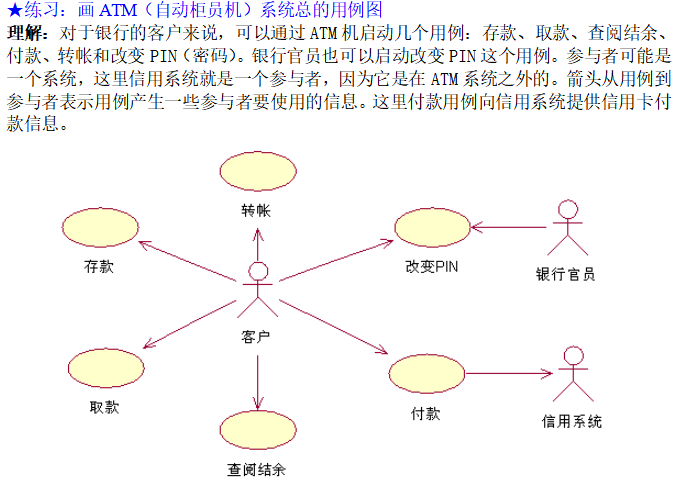
3.画出能正确表示下面叙述的类图（括号内为类名）。

 “一个雇员（Employee）最多由一个经理（Manager）管理，某些经理管理多个雇员，某些经理不管理任何雇员”。



4.设计一个饮料自动售货机系统，其主要功能是向顾客出售饮料，同时供应商需要向其中放置饮料，收银员需要向其中放置零钱和收回营业收入。画出该系统的用例图。





5.根据如下Java代码画出相应的顺序图。

请分析创建类EOutMessage对象并调用该对象的do2()方法的执行逻辑，并画出顺序图（时序图）。（本小题8分）

