软件工程作业讲评

Peking
University

第一次作业

1、分析增量模型和演化模型之间的区别。

- 增量模型意指需求可以分组，形成一个一个的增量，成为一系列增量产品，每一增量可以分别地开发。
- 演化模型是一种有弹性的过程模式，由一些小的开发步组成，每一步历经需求分析、设计、实现和验证，产生软件产品的一个增量。通过这些迭代，完成最终软件产品的开发。
- 演化模型主要针对事先不能完整定义需求的软件开发，而增量模型是在需求确定的情况下，对需求进行分组。

2、简述原型构造在软件开发中的作用。

- 原型是显示地规划如何使用一个或多个演化的增量。作为一个明确的需求揭示工具，原型构造在软件开发中有以下作用：
  - （1）揭示那些以后将在具有完备功能的、可交付的、可支持的增量中予以实现的需求。
  - （2）可以用于为一个项目或一个项目的某些部分，确定技术、成本和进度的可能性。

3、简述软件需求的分类，以你所熟悉的一个软件项目说明该项目中的各种软件需求。

软件需求分为功能、性能、外部接口、设计约束、质量属性。

功能需求规约了系统或系统构件必须执行的功能。

性能需求规约了系统或系统构件必须具有的性能特性。

外部接口需求规约了系统或系统构件必须与之交互的硬件、软件或数据库元素。也可能规约其格式、时间或其他因素。

设计约束限制了系统或系统构件的设计方案。

质量属性规约了软件产品必须具有的一个性质是否达到质量方面一个所期望的水平。

4. 举例说明结构化方法给出的控制复杂性机制。

- 从下面几个角度着手答题
  - 分解与抽象
    - 数据抽象
    - 过程抽象
  - 信息隐蔽原则
    - 模块化原则
  - 开发方法
    - 自顶向下，逐步求精

答题情况：
回答不够全面，没有举例说明

机制：
- 上层数据流打包
- 同层的数据流图可编号对应以免形成的复杂的连线。
- 高层的数据处理可以分解
- MSD中，上层的过程包含底层过程的内容
- 通过“模块”，“数据处理”来封装系统功能
- 设计过程自顶向下，并提供映射机制
- 等等

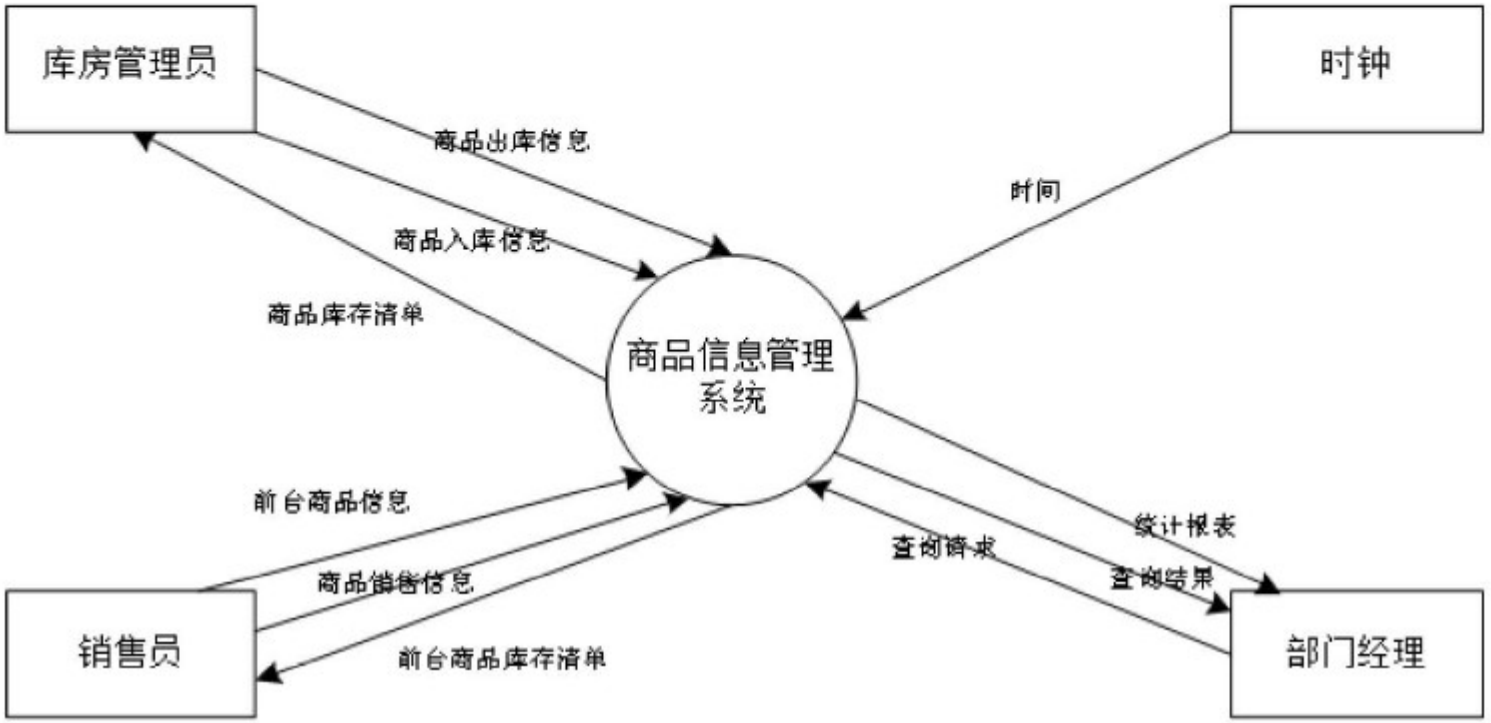
原则：
- 在数据流图中，每一幅图的图元应该个数应该控制在5—9之间
- 模块划分高内聚、低耦合
- 等等

5、软件工程教科书上 P85 页第 8 题，并给出该系统的 use case 图。

- 注意理解和应用结构化方法的建模过程
  - **建立系统的功能模型（数据流图DFD）**
    - 首先：建立系统环境图（顶层数据流图），确定系统边界，及所有的输入、输出
    - 继之：自顶向下，逐层分解，直到所有的功能都足够简单。
  - **建立数据字典**：
    - 以一种准确的和无二义的方式定义所有被加工引用的数据流和数据存储。
    - 定义数据流、定义数据存储、定义数据项
  - **给出加工小说明**：
    - 集中描述一个加工“做什么”，即加工逻辑，也包括其它一些与加工有关的信息，如执行条件、优先级、执行频率、出错处理等。
    - 判定表、判定树

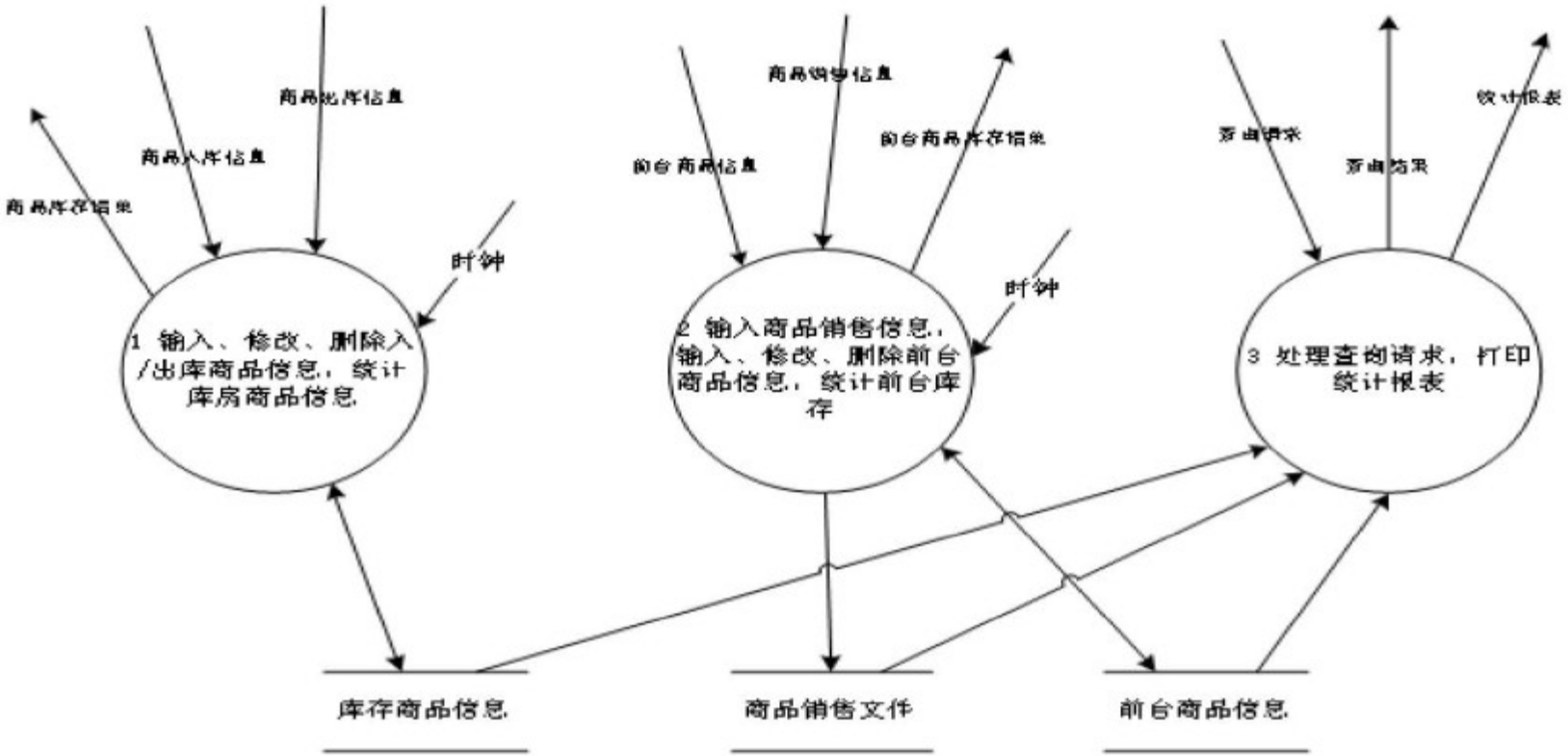
顶层数据流图

- 注意数据源有哪些：库房管理员，销售员，部门经理，系统时钟；数据潭有：库房管理员，销售员，部门经理



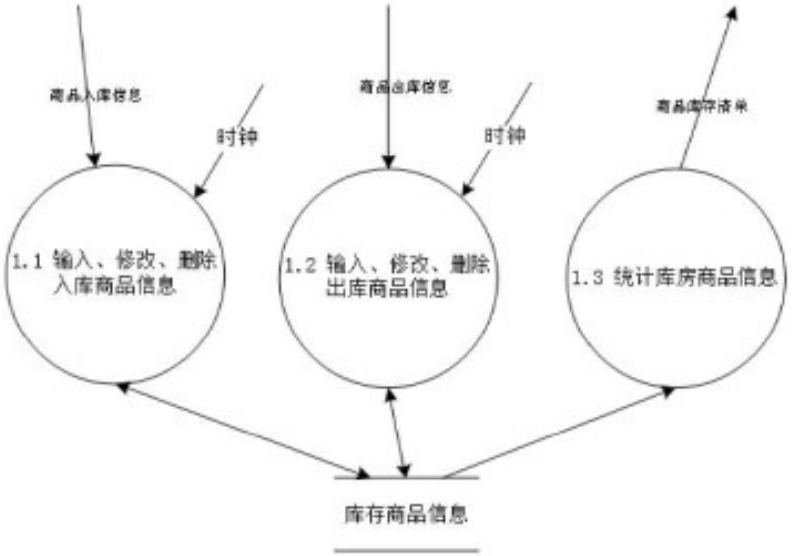
Peking University

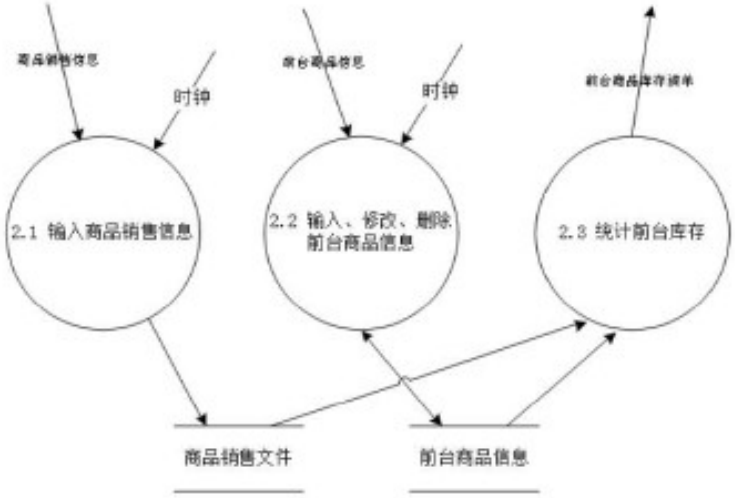
0层数据流图

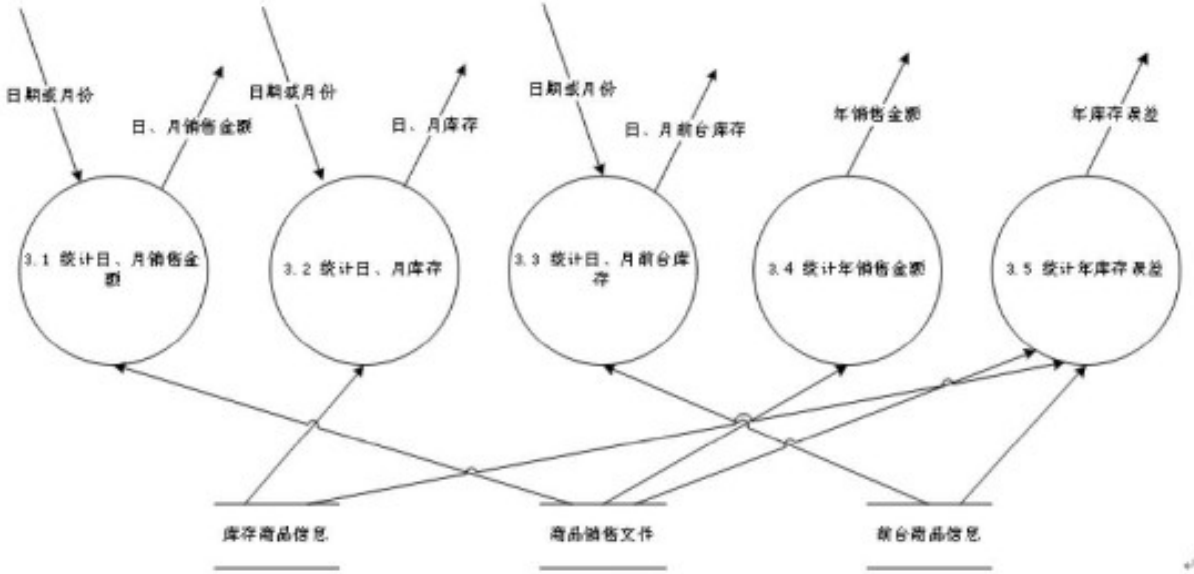


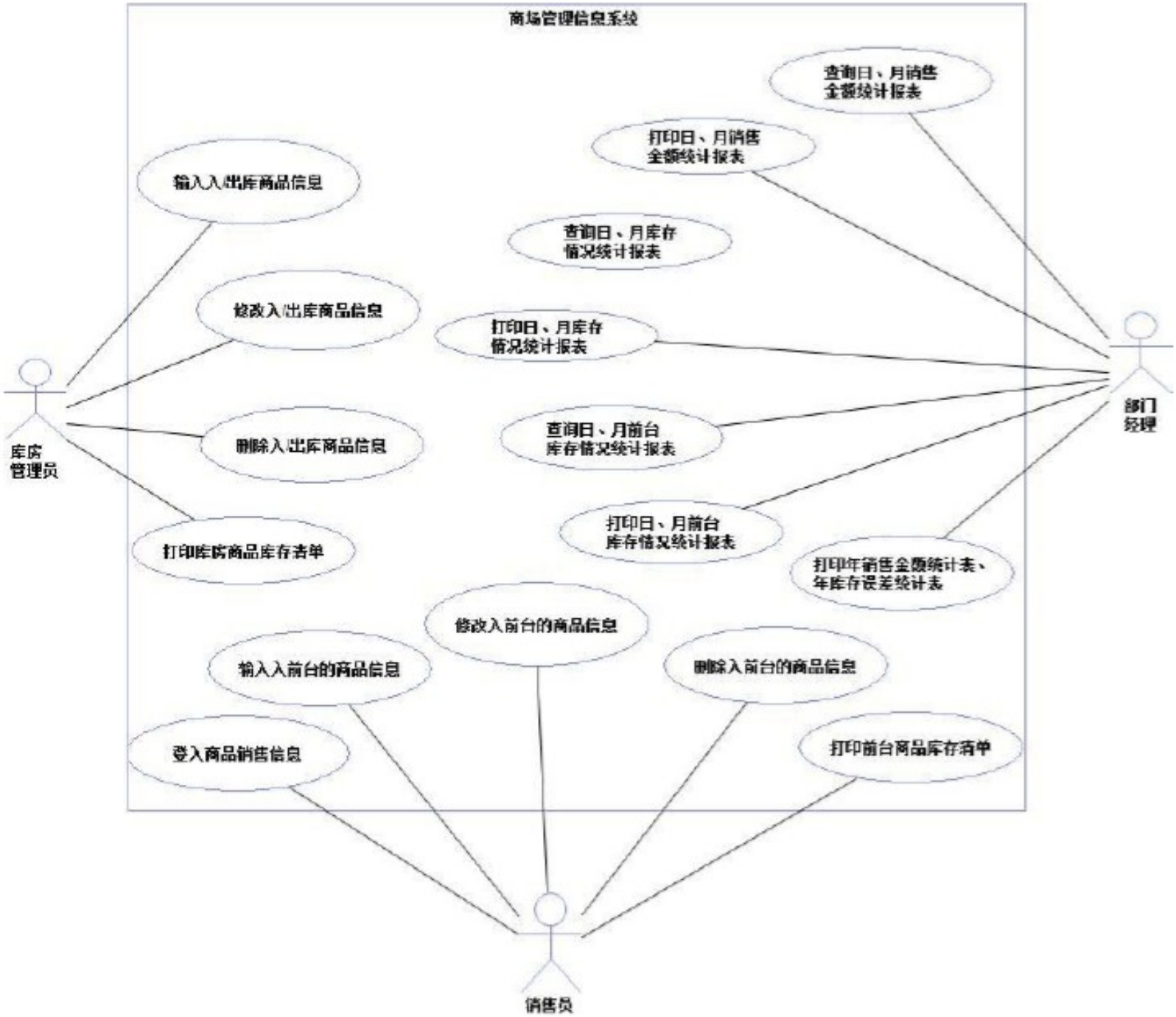
Peking University

1层数据流图


商品入库信息
时钟
1.1 输入、修改、删除入库商品信息
商品出库信息
时钟
1.2 输入、修改、删除出库商品信息
商品库存清单
1.3 统计库房商品信息
库存商品信息


商品销售信息
时钟
2.1 输入商品销售信息
前台商品信息
时钟
2.2 输入、修改、删除前台商品信息
前台商品库存清单
2.3 统计前台库存
商品销售文件
前台商品信息


日期或月份
日、月销售金额
3.1 统计日、月销售金额
日期或月份
日、月库存
3.2 统计日、月库存
日期或月份
日、月前台库存
3.3 统计日、月前台库存
年销售金额
3.4 统计年销售金额
年库存差益
3.5 统计年库存差益
库存商品信息
商品销售文件
前台商品信息


商场管理信息系统
查询日、月销售
金额统计报表
打印日、月销售
金额统计报表
输入入/出库商品信息
查询日、月库存
情况统计报表
修改入/出库商品信息
打印日、月库存
情况统计报表
部门
经理
库房
管理员
删除入/出库商品信息
查询日、月前台
库存情况统计报表
打印日、月前台
库存情况统计报表
打印库房商品库存表单
打印年销售金额统计表、
年库存误差统计表
修改入前台的商品信息
输入入前台的商品信息
删除入前台的商品信息
登入商品销售信息
打印前台商品库存清单
销售员

❑答题情况

- 只画数据流图，没有给出数据字典和加工小说明
- 数据流图的四个基本成分要画全，命名和图形要正确：
  - 数据流，加工，数据存储，数据源和数据潭
- 至少要画到1层数据流图
- 各层数据流图的输入输出要一致
- 加工要编号

总体情况:
第 1 ， 2 题基本都能答出要点

第 3 ， 4 题不少同学没有举例说明（尤其第四题）

第 5 题有少数同学没有审清题目，只给出了 Use Case 图，而题目要求：完成教科书上题目“并”给出 Use Case 图。部分同学在 DFD 图中，层与层之间数据流没有对应；有的同学图元表示没有按照书本讲义上的规范（当然这个与作图工具有关）。

第一次课堂小测练习

- **第 1 题：下面列出一些基于计算机的系统：**
  （1）一个简单的文字处理系统
  （2）实时数据采集系统
  （3）电子邮件系统
  （4）通用的登记系统
  （5）网上订购系统（如预定飞机票）
  （6）其他系统
  请选以上的一个系统，编写一个简要的系统需求说明。
- 同学们基本上上选择了（1）（3）（5）作答
- 大部分同学都涵盖了功能需求、非功能需求，一半以上同学能够按照书本讲义提供的 5 个方面作答，少数同学还按照“软件规格说明书”的格式书写
- 存在比较普遍的问题：主要从客户的角度，而没有从系统其他参与者（例如系统管理员）以及系统本身的约束（例如可靠性、响应时间等）去考虑，以下是一些容易被忽略的需求
  - 文字处理系统：响应时间
  - 电子邮件系统：系统并发度、服务器吞吐率
  - 网上订购系统：订票管理者可以获取用户订票信息、生成报表等

第 2 题：如何理解模块独立性？用什么指标来衡量模块独立性？

- 大部分同学都能按照书本讲义解释模块独立性，并都能指出内聚和耦合是衡量模块独立性的两个指标

- 所谓模块的独立性，是指软件系统中每个模块只涉及软件要求的具体的子功能，而和软件系统中其他的模块的接口是简单的。例如，若一个模块只具有单一的功能且与其他模块没有太多的联系，那么，我们称此模块具有模块独立性。
- 一般采用两个准则度量模块的独立性。即模块间的耦合和模块的内聚。
  - 耦合是模块之间的相对独立性（互相连接的紧密程度）的度量。模块之间的连接越紧密，联系越多，耦合性就越高，而其独立性就越弱。
  - 内聚是模块功能强度（一个模块内部各元素彼此结合的紧密程度）的度量。一个模块内部各元素之间的联系越紧密，则它的内聚性就越高，相对地，它与其他模块之间的耦合性就会减少，而模块独立性就越强。
  - 因此，模块独立性比较强的模块应是高内聚低耦合的模块。
- 答题关键：记住概念，模块，模块独立性，耦合，内聚

- **第 3 题：模块独立性与信息隐蔽（反映模块化有效程序的属性）有何联系？**
  - 大部分同学都能回答出模块独立性与信息隐蔽的正相关关系，表述上可能有些差别

- **第 5 题：耦合性的概念和软件的可移植性有什么关系？请举例说明你的论据。**
  - 大部分同学都能指出，耦合性越低的软件，可移植性越强
  - 少数同学没有举例说明

- **第 4 题：模块的内聚性程度与该模块在分层结构中的位置有关系吗？说明你的论据。**
- 此题同学们的意见分成两派，各占一半：
  - 一半的同学认为，内聚性程度与分层位置是无关的
    - 观点 1： 内聚性是模块内部彼此结合的紧密程度，分层位置是模块之间的位置关系，模块内部的度量（内聚性）与模块之间的度量（分层位置）没有必然联系
    - 观点 2： 一个设计良好的软件是高内聚的，因此无论分层位置如何，内聚性程度都是很高的
  - 另一半同学认为，处于分层位置越低的模块，内聚性越强
    - 观点：模块分层位置越低，它们完成的功能就越单一，因此内聚性越强

- 模块的内聚性程度与该模块在分层结构中的位置有关系。在分层结构中，通常随着层次的由高到低，模块的内聚性是由弱到强。
- 例如在一个正文加工系统（P87）最顶层的的主控模块，它调用下层模块完成正文加工的全部功能；第二层的每个模块控制完成正文加工的一个主要功能，例如“编辑”模块通过调用它的下属模块可以完成六种编辑功能中的任何一种。这样，在第三层的六种编辑功能模块，每个模块的内聚性最高，而第二层的各功能模块如“编辑”，它只是完成以下六种模块的协调与调用，所以它的内聚性就稍微差一些，而顶层的主控模块的内聚性就更差。

第二次作业

P119-2-(1) 结构化方法总体设计的任务及目标

- 任务：把系统的功能需求分配给一个特定的软件体系结构
- 目标：建立系统的模块结构图，即系统实现所需要的软件模块——系统中可识别的软件部分，以及这些模块间的调用关系。

P119-2-(2) 结构化方法详细设计的任务及目标

- 任务：详细描述模块
- 目标：将总体设计阶段所产生的系统高层结构，映射为低层结构，也是系统的最终结构。

第二次作业

- **P119-2-(5) 为什么说结构化分析与结构化设计之间存在一条“鸿沟”？**
  - 因为结构化分析和结构化设计采用的是两种完全不同的模型，结构化分析使用的是数据流图 + 数据字典 + 加工小说明，而结构化设计使用的是模块结构图、 PAD 图、 NS 图等等，从分析到设计之间不能无缝连接，还需要一些转化工作，所以说二者之间存在着一条“鸿沟”。

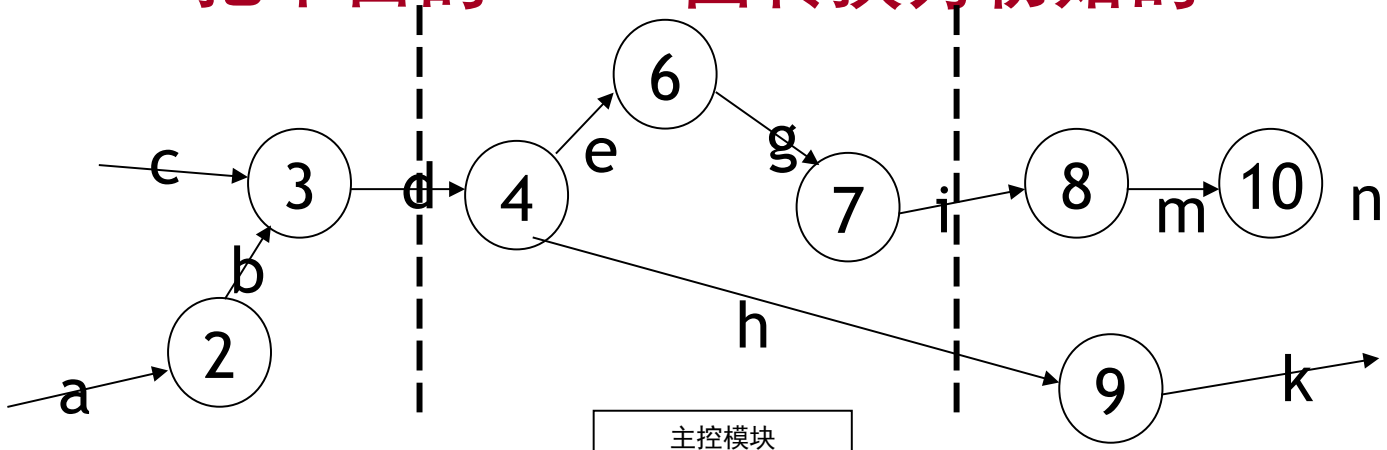
第二次作业

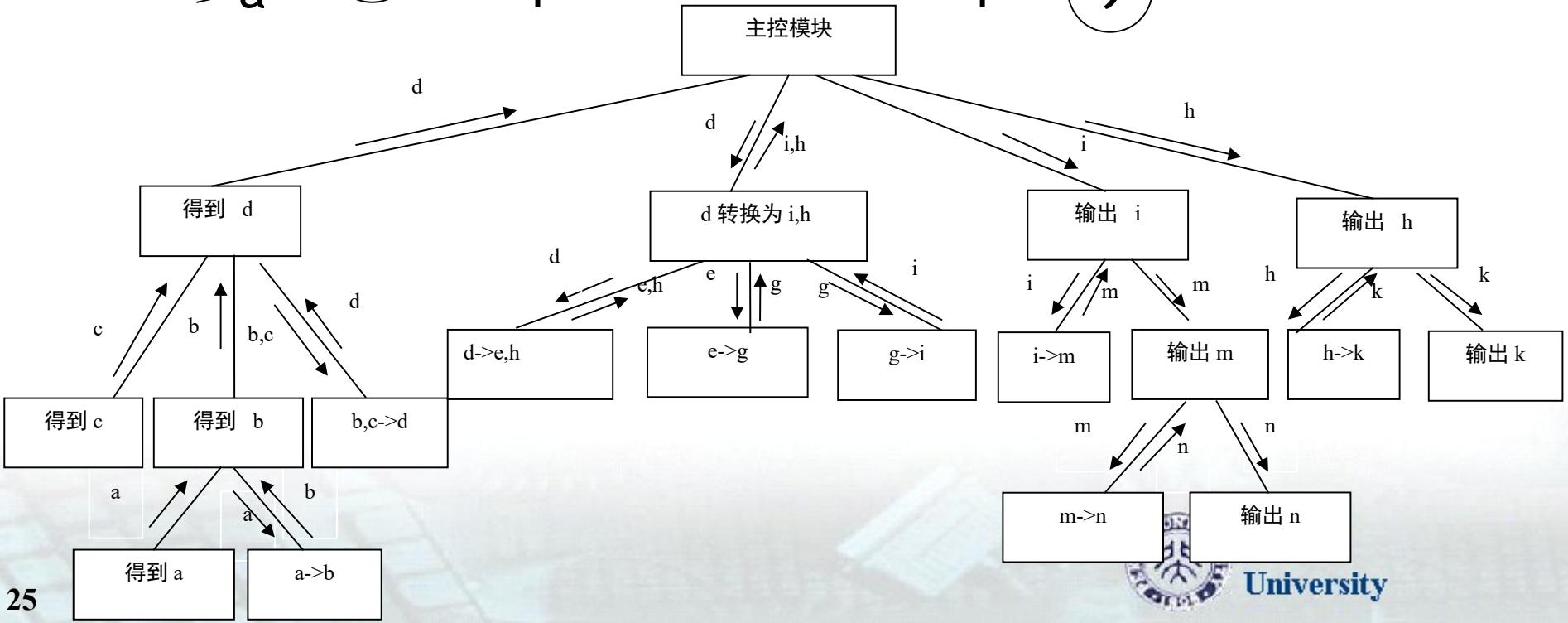
P119-2-(6) 依据一个系统的 DFD，将其转换为 MSD 的基本思路

- 首先利用变换设计，把系统的数据流图分为输入、中心变换和输出 3 个部分，设计上层模块；
- 然后根据各部分数据流图的结构特点，适当地利用变换设计和事务设计进行细化，得到初始的模块结构图；
- 再按照“高内聚低耦合”的原则，对初始模块结构图进行精化，得到最终的模块结构图。

第二次作业

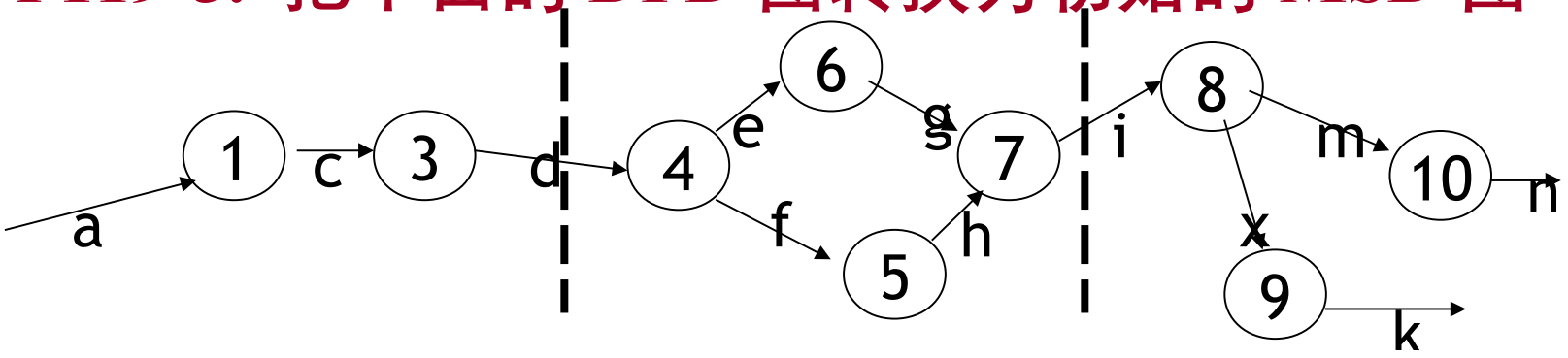
- P119-5. 把下面的 DFD 图转换为初始的 MSD 图

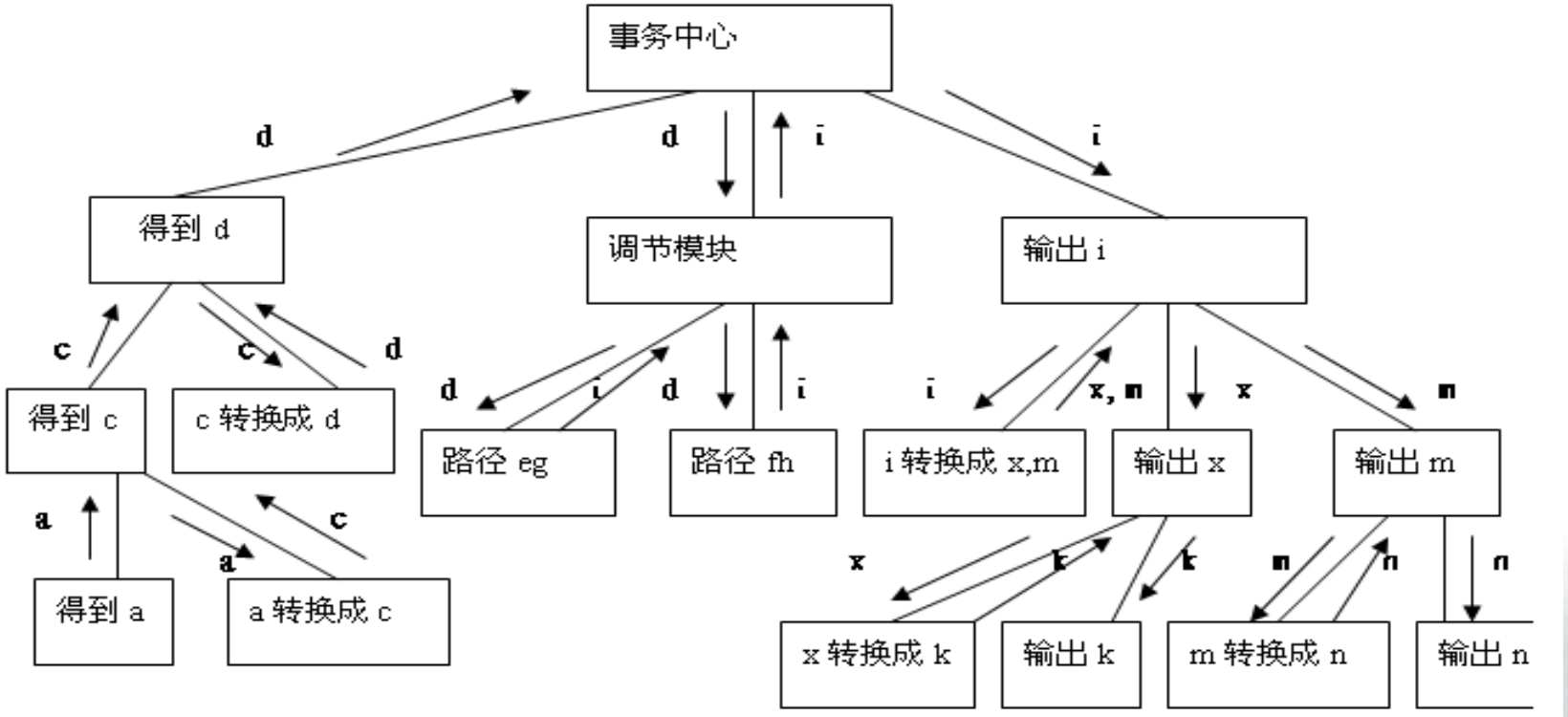




第二次作业

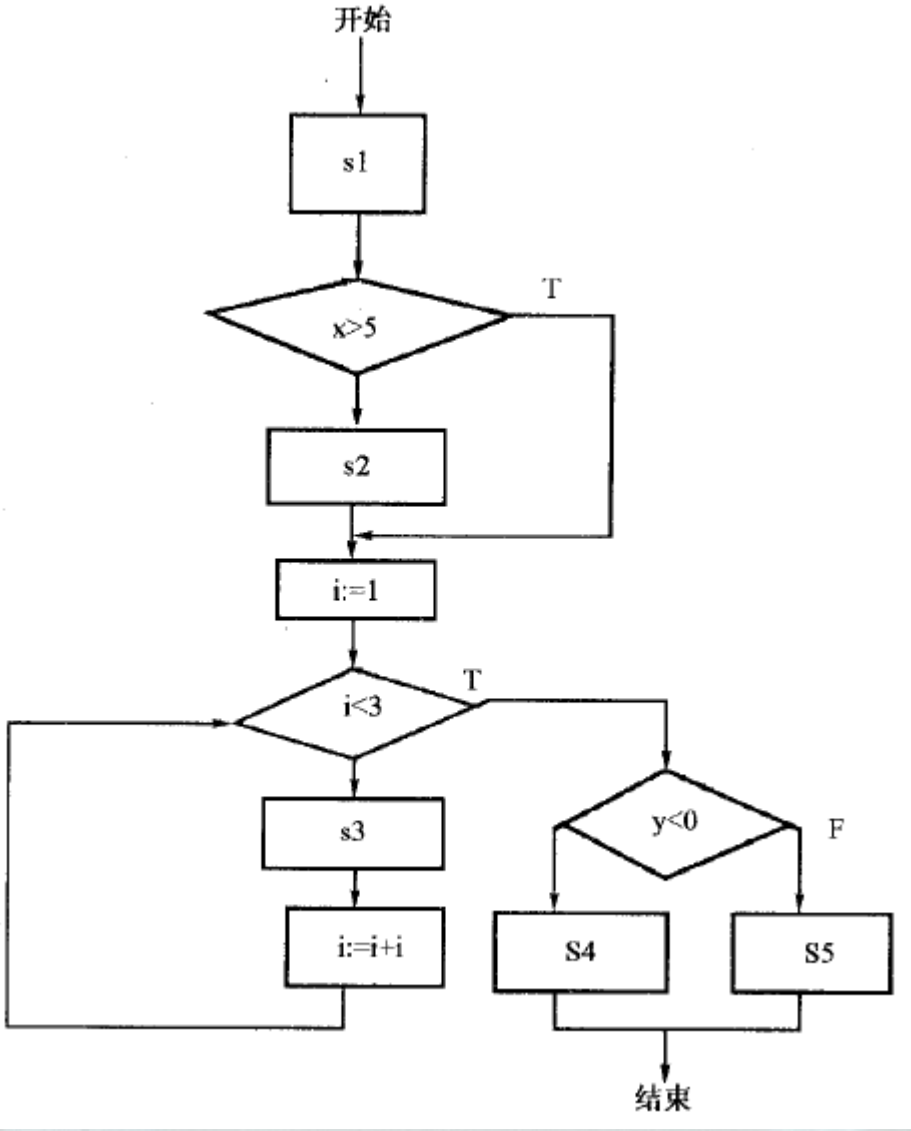
- P119-5. 把下面的 DFD 图转换为初始的 MSD 图





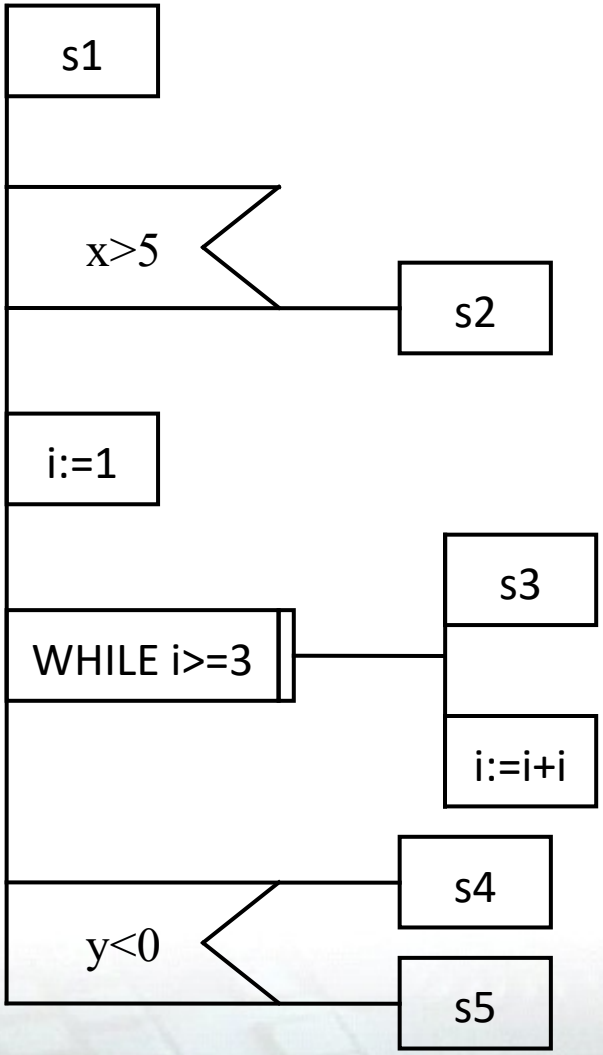
第二次作业

- P119-6. 把下面的程序流程图转换为 PAD, N-S 图和伪码。

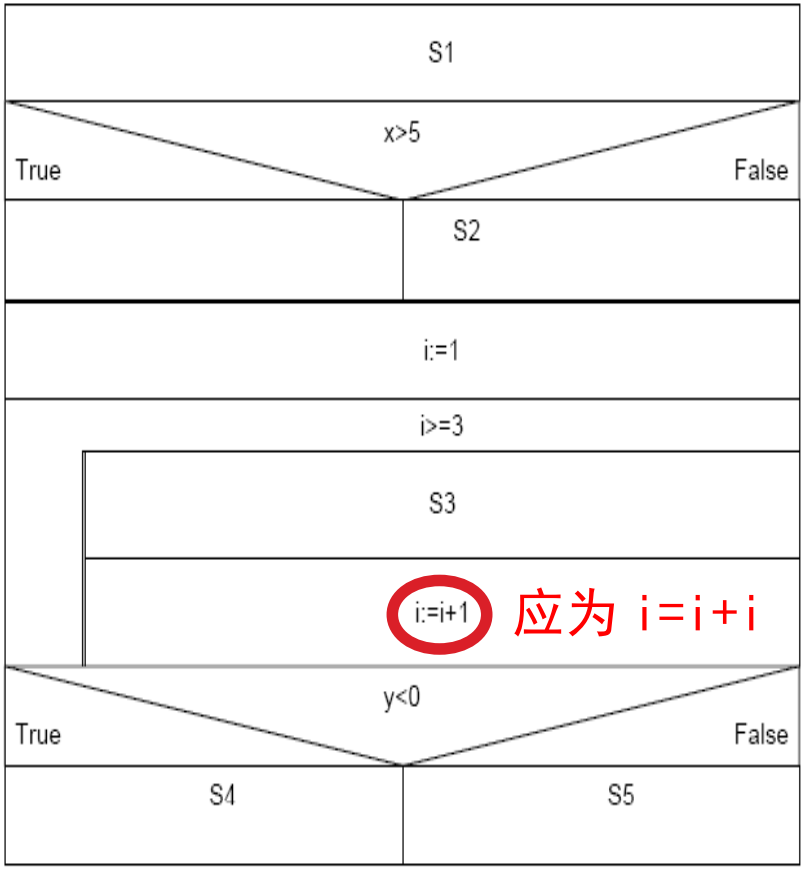


第二次作业

PAD 图



N-S 图



第二次作业

❑伪码

```
begin
    s1;
    if x<=5 then s2;
    i:=1;
    while i>=3 do
                    begin
                        s3;
                        i:=i+i;
                    end
    if y<0 then s4
           else s5;
end
```

注：按题目中流程图所示，第一个判断执行 s2 的条件为 $x > 5$ 不成立，循环执行的条件为 $i < 3$ 不成立，循环体内的赋值语句为 $i := i + i$，虽然都不符合常理，但是书上题目是这么写的，所以 PAD 图，N-S 图及伪代码都按照题目中的写法。

答题情况：

审题不仔细：如 while 循环的条件，以及 i=i+i ;

格式不标准，错用符号

第三次作业

第三次作业

- **P169-1 解释以下术语，并举例说明。**
  - 对象：对象是类的一个实例。
  - 操作：操作是用来描述对象动态特征（行为）的 一个动作序列。
  - 类：是一组具有相同属性、操作、关系和语义的对象的描述。
  - 关联：关联是类目之间的结构关系，描述了一组具有相同结构、相同语义的链（ links ）。
  - 链：链是对象之间的连接（ connection ）
  - 泛化：泛化是一般性事物（称为超类或父类）和它的较为特殊种类（称为子类）之间的一种关系，即“ is-a-kind-of”关系。
  - 聚合：一种特殊形式的关联，表达一种“整体 / 部分”关系。
  - 依赖：依赖是一种使用关系，用于描述一个事物使用另一事物的信息和服务

答题情况：
部分同学没有举例！

P169-2-(5) 描述对象之间的关系所使用的概念

- 关联：关联是类目之间的一种结构关系，是对一组具有相同结构、相同链的描述。
- 泛化：一般性类目（超类 / 父类）和它较为特殊性类目（子类）之间的一种关系，即“ is-a-kind-of”关系。
- 细化 / 实现：细化是类目之间的语义关系，其中一个类目规约了保证另一个类目执行的契约。
- 依赖：依赖是一种使用关系，用于描述一个类目使用另一个类目的信息和服务。

答题情况：
部分同学没有给出详细解释

第三次作业

P169-2-(6) 用况之间有哪几种关系

- 泛化：用况 A 和用况 B 之间具有一般 / 特殊关系。
- 扩展：一个用况 A 的实例在特定的条件下可以由另一用况 B 所规约的行为予以扩展，并依据定义的扩展点位置，B 的行为被插入到 A 的实例中。
- 包含：用况 A 的一个实例包含用况 B 所规约的行为。

答题情况：
（1）不少同学回答关联关系。关联是参与者和用况之间的关系，但不是用况之间的关系。
（2）部分同学没有给出解释。

P169-2-(7) 面向对象为什么要从多个侧面建立系统模型

- 支持软件开发人员从不同目的（静态、动态）、针对不同粒度（系统、子系统、类目等），从不同抽象层和从不同视角来创建模型

P169-3-(4) 在什么情况下需要建立状态图？

- 状态图是显示一个状态机的图，其中强调了从一个状态到另一状态的控制流。一个状态机是一种行为，归约了一个对象在其生存期间因响应事件并作出响应而经历的状态。
- 当需要创建系统的行为生存周期模型，表达系统的动态结构，给出系统在生存期间可有哪些阶段、每一阶段可从事的活动以及对外所呈现的特征等方面的信息时，就需要建立状态图。

P169-3-(8) 面向对象方法与结构化方法在控制信息组织复杂性方面引入的机制？

- 面向对象方法
  - 信息组织的复杂性：
    - **抽象**：从许多事物中舍弃个别的、非本质的特征，抽取共同的、本质性的特征.
      - 系统中的对象是对现实世界中事物的抽象；
      - 类是对象的抽象；
      - 一般类是对特殊类的抽象；
      - 属性是事物静态特征的抽象；
      - 操作是事物动态特征的抽象。
    - **分类机制**：把具有相同属性和操作的对象划分为一类，用类作为这些对象的抽象描述。
    - **继承**：特殊类的对象拥有其一般类的全部属性和服务（一般 - 特殊结构）
    - **聚合**：把一个复杂的事物看成若干比较简单的事物的组装体，从而简化对复杂事物的描述。（整体 - 部分结构）
    - **多个视图**：从多个角度认识系统
  - 文档组织的复杂性—控制机制
    - **包**：使模型具有大小不同的粒度层次，以利于控制复杂性

第三次作业

- 结构化方法
  - 原则：
    - 分解与抽象：数据抽象、过程抽象
    - 信息隐蔽原则：模块化原则
    - 开发方法：自顶向下，逐步求精
  - 机制
    - 上层数据流打包
    - 同层的数据流图可编号对应以免形成的复杂的连线。
    - 高层的数据处理可以分解
    - MSD 中，上层的过程包含底层过程的内容
    - 通过“模块”，“数据处理”来封装系统功能
    - 设计过程自顶向下，并提供映射机制

答题情况：

面向对象方法中，很少有同学这道题想到了除包以外的机制

第三次作业

- P169-1-(1) 假定教务管理包括以课程为中心进行资源（教师、教室、学生）配置，并根据各科考试成绩进行教学分析。在这一假定下，结合实际情况，给出教务管理系统的需求陈述，建立该系统的类图，并选取其中一个典型的对象类，给出它的状态图。

第三次作业　P169-1-(1)

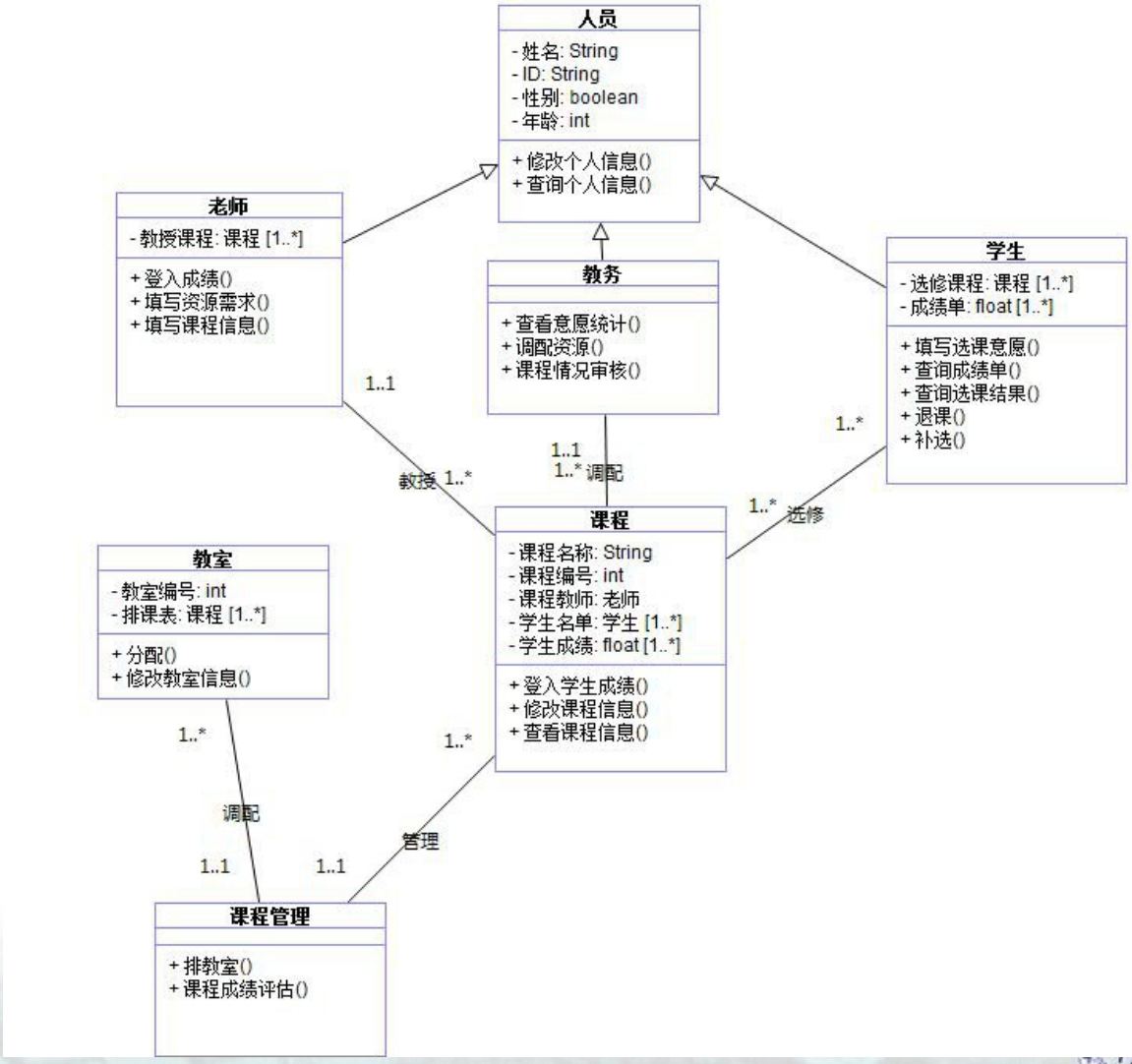
- 需求陈述（可在文字叙述的基础上配用况图）
  - 要点：功能需求和非功能需求
  - 功能性需求
    - 考虑系统的用户有哪些
      - 教师，学生，教务人员，系统管理员（超级用户）
    - 考虑系统有哪些功能
      - 共有的功能
        - 登陆：用户名，密码
      - 与教师有关的功能
        - 授课申请、填写资源需求，填写课程信息，录入成绩…
      - 与学生有关的功能
        - 选课，退课，查询成绩单，查询课程信息…
      - 与教务人员有关的功能
        - 审核（教师的授课申请，学生的选退课申请），教室分配…
      - 与系统管理员有关的功能（不是必须的）
        - 管理系统用户，录入、审核用户信息

第三次作业　P169-1-(1)

- 非功能性需求
  - 性能需求
    - 承受压力：选课高峰期要同时承受几千用户的并行访问
    - 查询课表等的响应时间
    - …
  - 外部接口
    - 用户接口？系统接口？软件接口？（与学校人员管理系统，选课系统等连接）
  - 设计规约
  - 质量属性
    - 可靠性：选课失败后，能对之前所选课程进行保存?
    - 存活性
    - 可维护性
    - 用户友好性

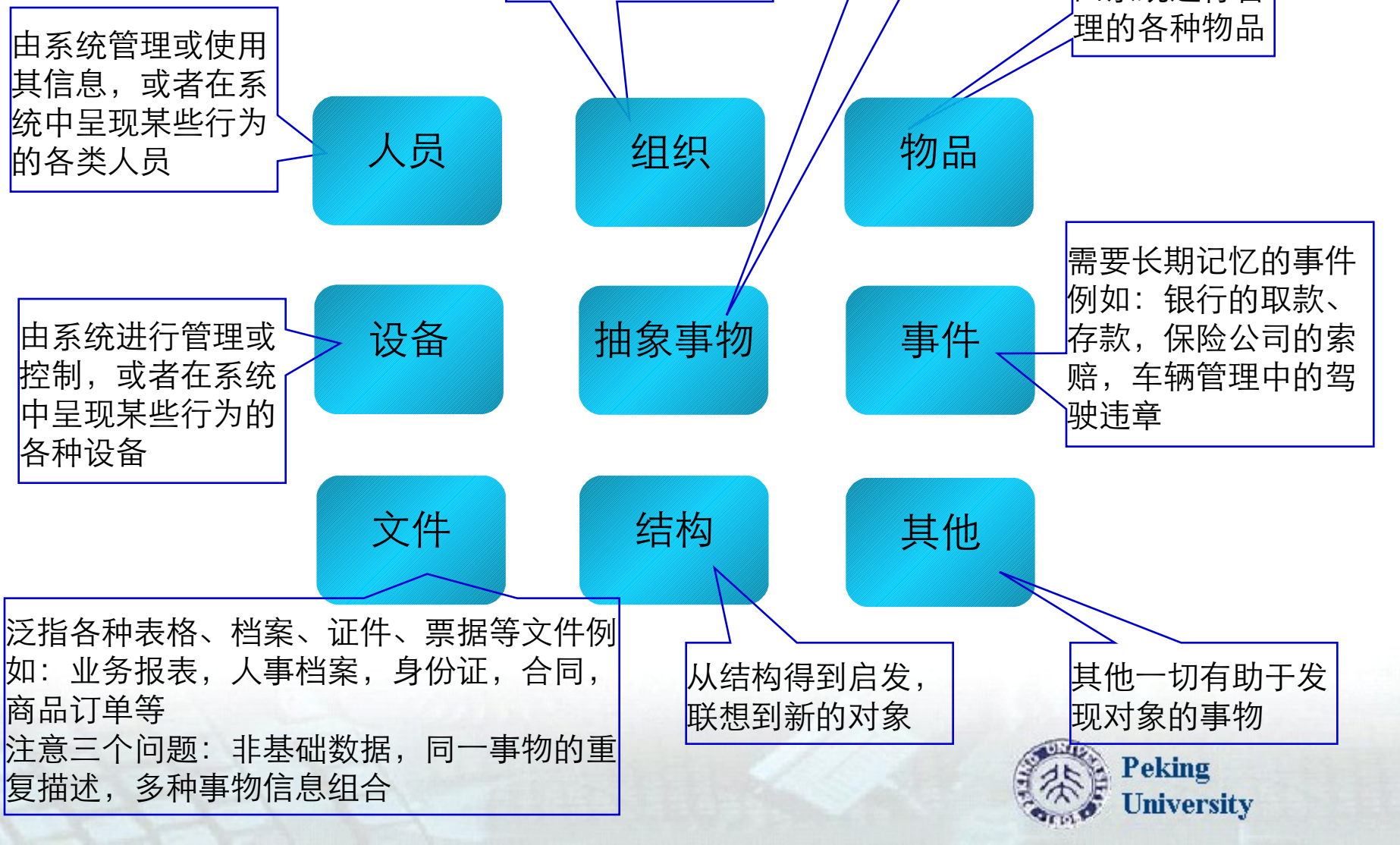
第三次作业　P169-1-(1)

❑建立类图



策略与启发

（1）考虑问题域：



（2）考虑系统边界：

考察在系统边界以外与系统交互的各类参与者
考虑通过那些对象处理这些参与者的交互

人员　　设备　　外系统

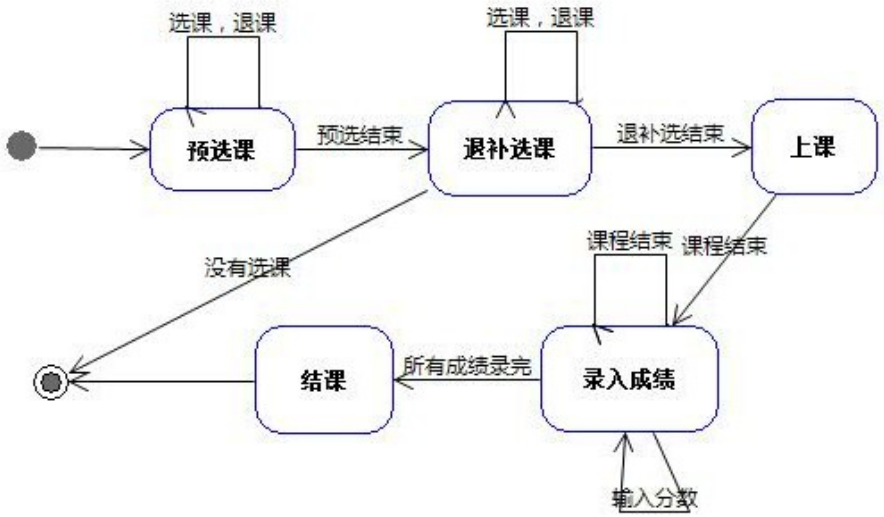
（3）考虑系统责任：

检查每一项功能需求是否已有相应的对象提供，
发现遗漏的对象

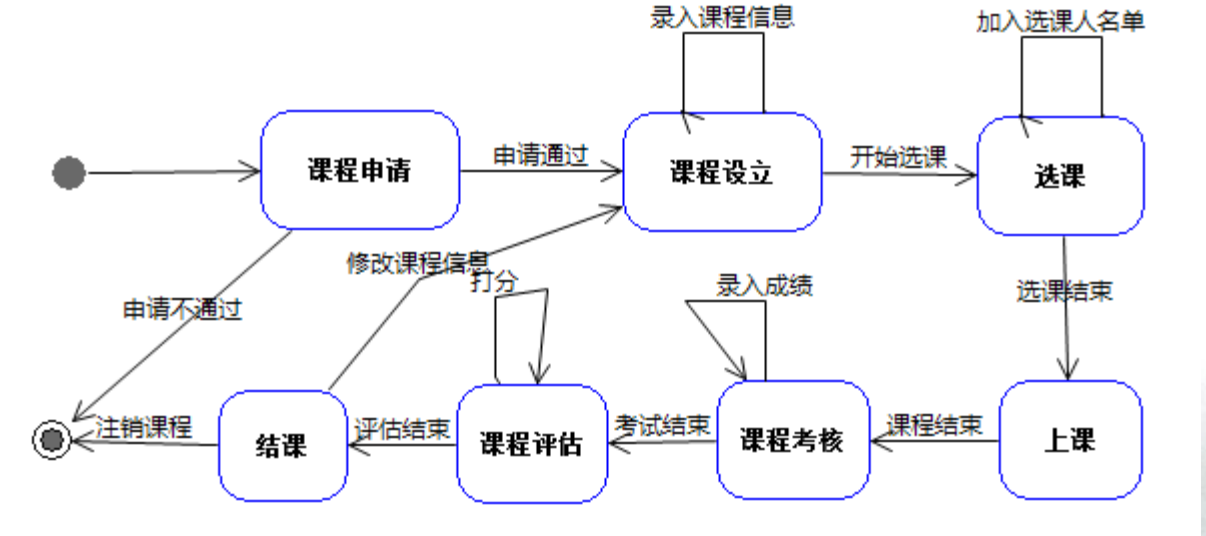
第三次作业　P169-1-(1)

❑建立状态图

- 学生对象



- 课程对象



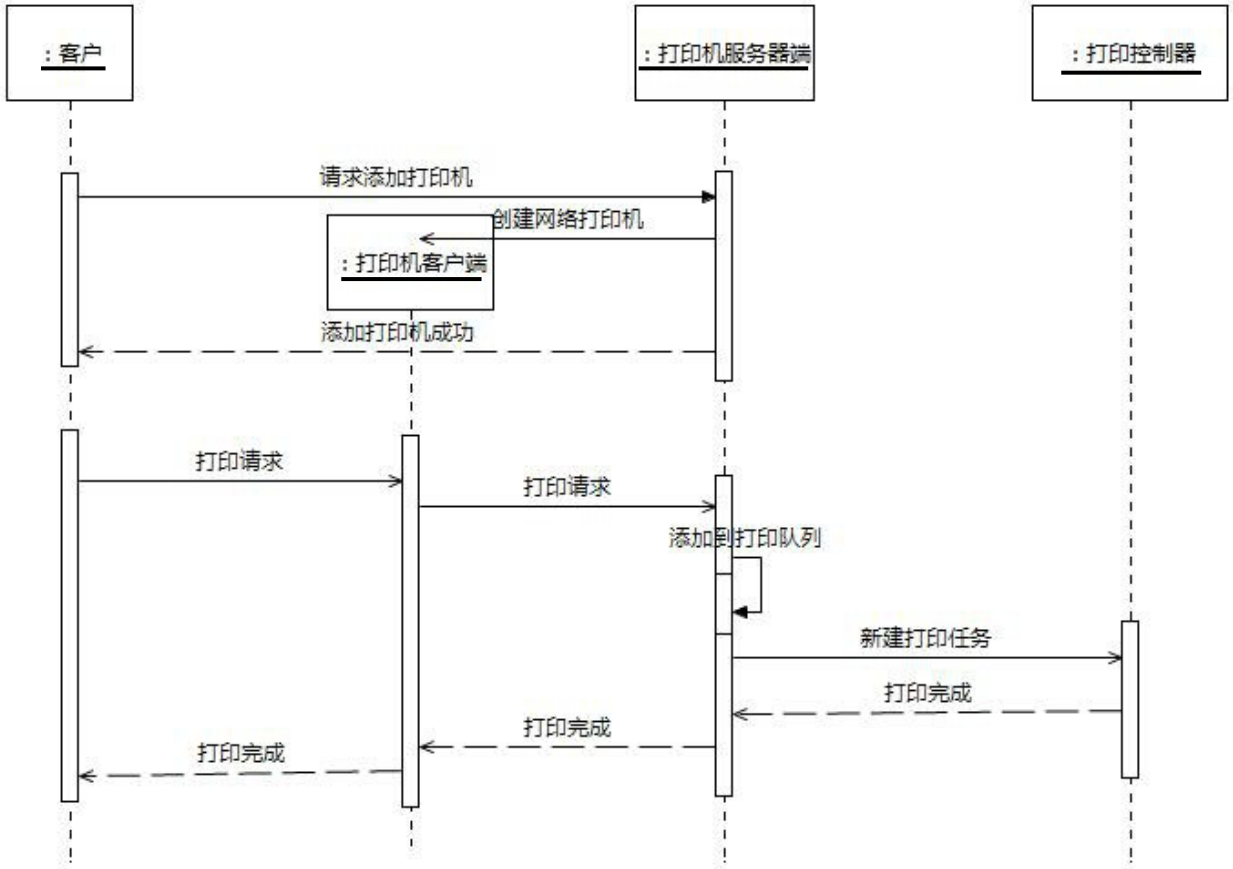
第三次作业　P169-1-(1)

❑存在问题

- 需求陈述太简单：没有明确划分系统边界，没有列举出系统的完整功能；只有很少的同学画了用况图；没有写非功能需求。
- 类图：与需求分析结果对应不好：需求中的功能没有在类图中体现。画得不详细，有的缺少属性，有的缺少方法操作，还有同学都没有写；缺少类之间的关系
- 对状态图理解不到位，不是针对一个类对象进行建模，而是画了整个系统的状态图。有些状态未必是对象会出现的状态，有些状态转移未必合理
- 有些状态不是系统功能范围内会出现的状态

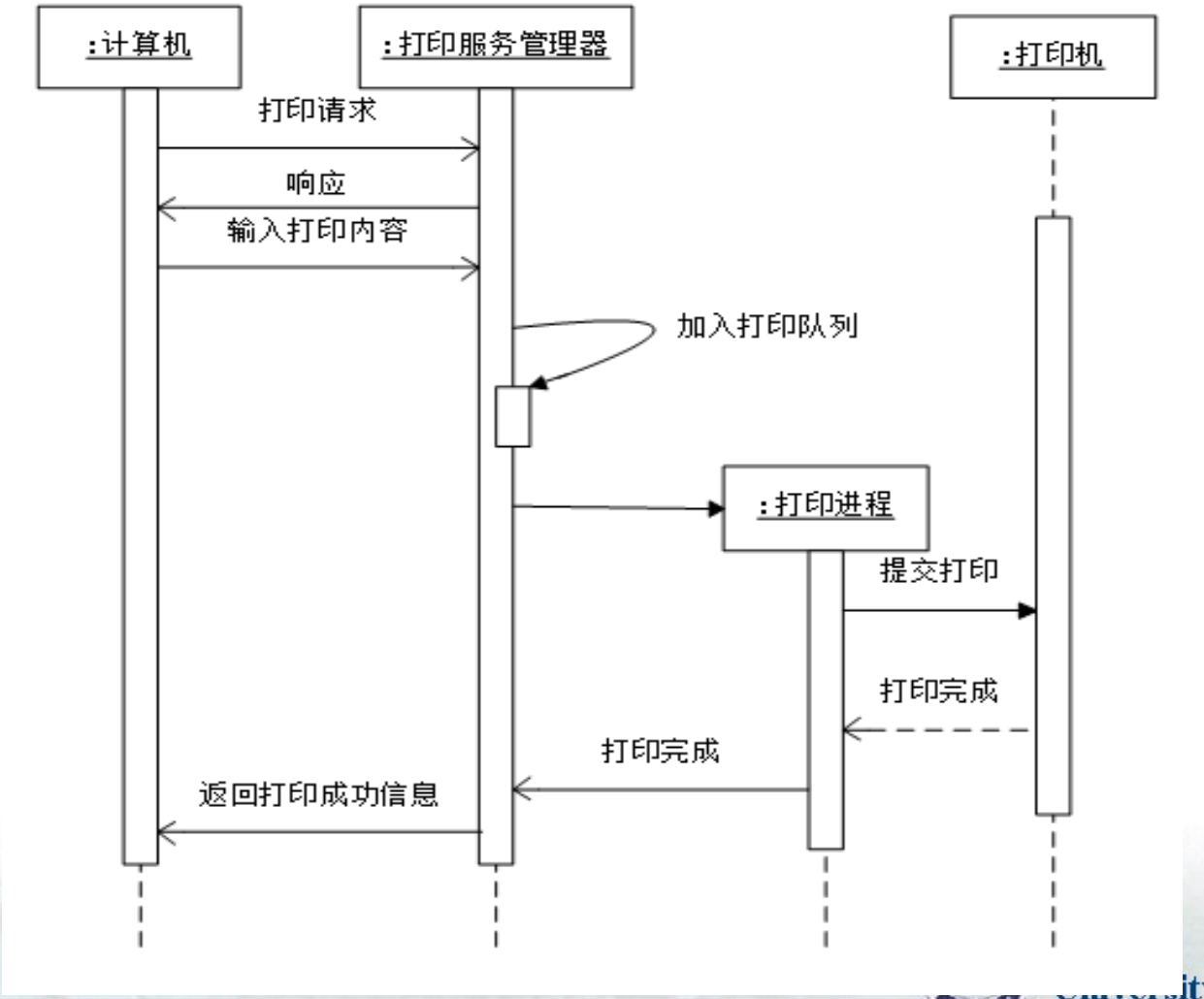
第三次作业

- P169-1-(3) 考虑使用网络打印机进行打印时出现的各种情况，绘制顺序图。



第三次作业

- 另一种:


:计算机
:打印服务管理器
:打印机
打印请求
响应
输入打印内容
加入打印队列
:打印进程
提交打印
打印完成
打印完成
返回打印成功信息

第三次作业

❑存在的问题

- 生命线表示的是对象，命名格式为：“对象名：类名”，或者用匿名类：“：类名”。很多同学直接用“类名”表示，也没有下划线。
- 注意分清楚哪些是同步消息哪些是异步消息
- 客户端不应该采用轮询的方式发送请求，最好用请求队列
- 没有把系统中涉及到的交互对象考虑清楚，画得太简单。

Q&A

Peking University