

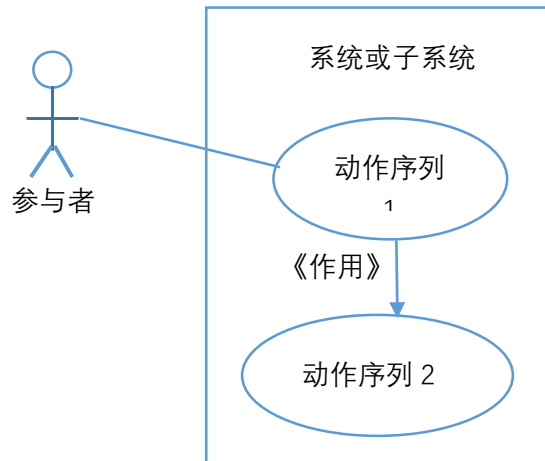
UML 建模相关复习点

用例图或者叫做用况图：(use-case diagram)

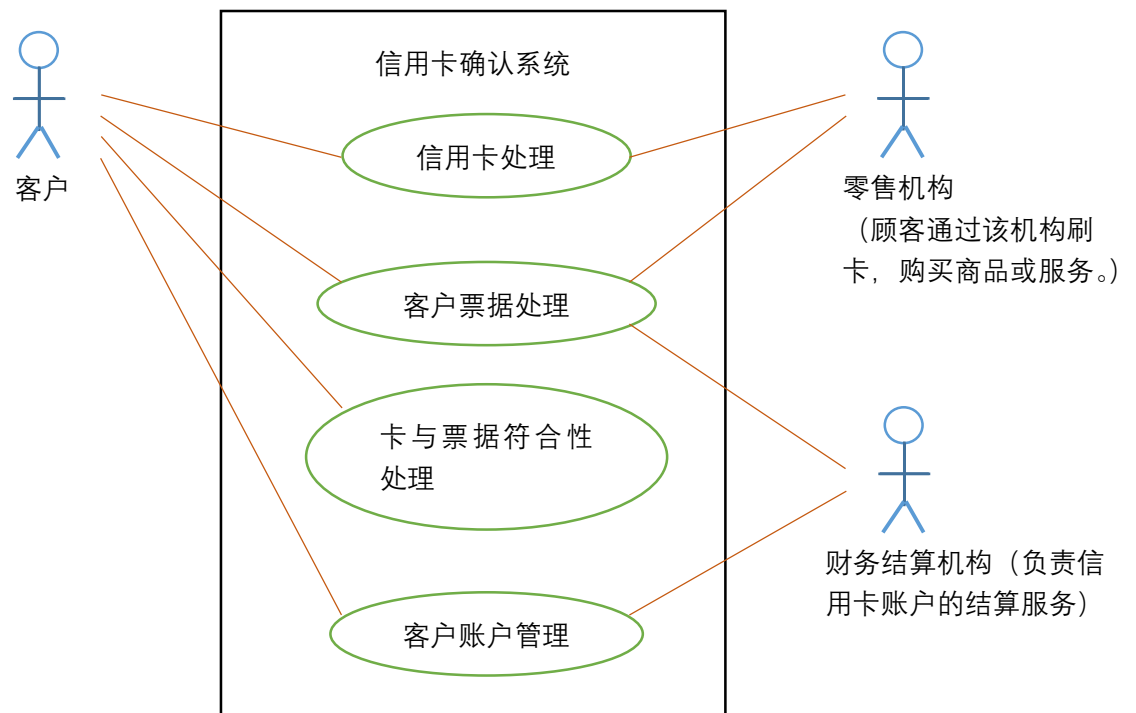
概念：用例图描述各个参与者在各个用例中的参与情况，描述系统为用户所感知的外部视图。

Use-case：是对一组动作序列的描述，系统执行这些动作应产生对特定参与者有值的、可观察的结果。

基本画法：



比如，一个简化的信用卡确认系统：(王立福第三版书：155 页)

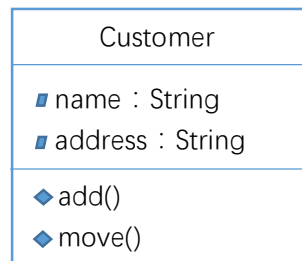


类图：(class diagram)

类是具有相同属性、操作和关系的对象集合的总称。通常画成矩形。包括三个部分：名称、属性、操作。

类可以有无数个或 0 个属性，属性可以只写名字。

操作是指对一个类中所有对象要做的事情的抽象，可以看成是类中调用的方法，类可以没有操作。操作可以是抽象操作（即没有实现的操作）。

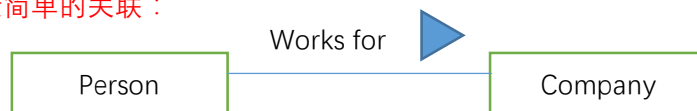


UML 中对抽象客观事物的 8 个术语：类、接口、协作、用况、主动类、构件、制品、节点。这些统称为“类目”。

表达类目之间关系的术语：

1. 关联

最简单的关联：



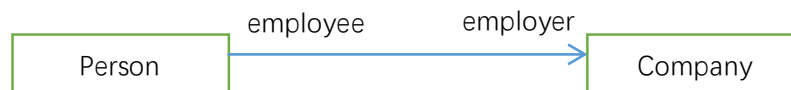
即：通过一条实线连接两个类目，在线段上方命名该关系。如果关联有方向性，用实心三角形指示关联的方向。

关联的导航：(是指通过给定的类目，可以找到与之关联的另一个类目)



单向导航：通过一个指示方向的单向箭头来修饰。一般导航是双向的。

关联的角色：



角色是关联一端的类目对另一端类目的呈现。如果一个类目参与关联时具有特点的角色，那么就要显式的命名该类目在关联中的角色。

关联的可见性：(一般不会考)

- + 公共可见
- 对该关联之外的任何对象而言，不可见
- # 该端的对象只有另一端的“子孙”可访问
- ~ 在同一包中声明的类可访问的

关联的多重性：



一般表示法：下限..上限。比如 1..2，表示最少一个，最多 2 个。

如果只有一个整数值，就是表示上限值。*表示无穷的非负正整数范围。即表达无限多个的意思。

上例表明：Person 最多一个公司，Company 可以无限多个 Person。

聚合：(强调“部分-整体”的关系，表示一个类目是另一个类目的一部分)



聚合表示为用带有空心菱形的线段连接，空心菱形放在“整体类”这一端。

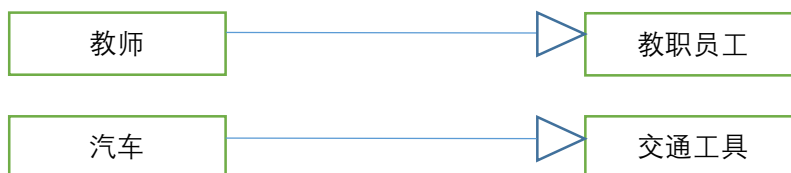
组合：(比聚合更强烈的“部分-整体”关系，指该部分和整体不可分割或具有相同生存周期)。



组合表示为带有实心菱形的线段连接，实心菱形放在“整体类”这一端。

理解组合：在一个时间段内，整体类的实例中至少包含部分类的实例，整体类负责创建和消除部分类的实例。或者整体类和部分类的实例具有相同的生存周期。

2. 泛化



泛化表示子类是超类或父类（一般性类目）的一个特殊性类目的一种关系。比如：教师是教职员的一个特殊子类。可以形容为“A is B”的概念。

泛化是从子类（特殊性类目）到父类（一般性类目）的带空心三角形的线段，空心三角形在父类端。

3. 细化

使用带空心三角形的虚线段。

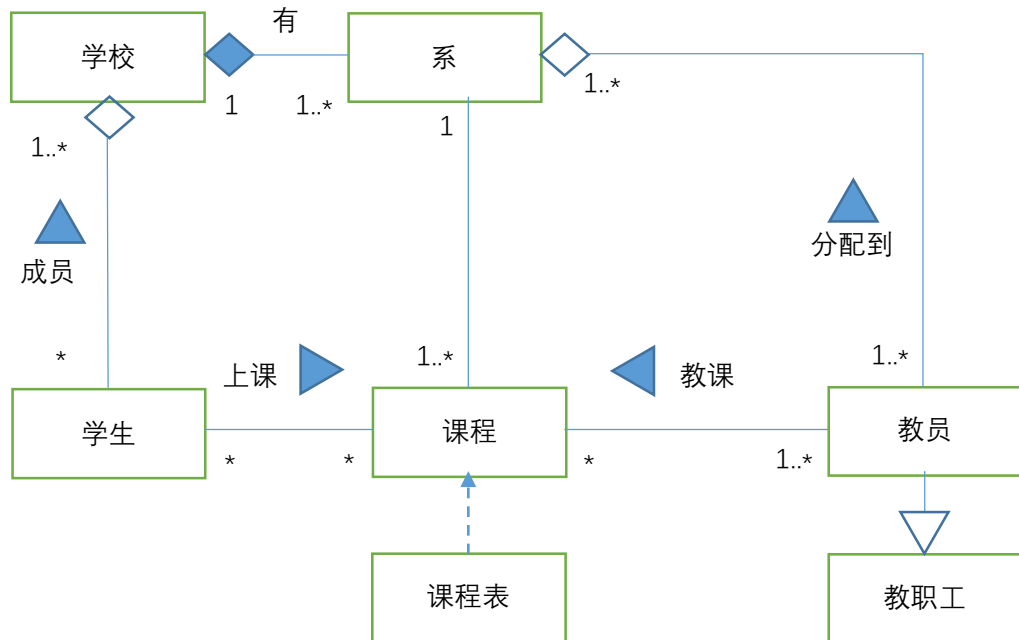
（一般不会考）

4. 依赖



依赖是一种使用关系，用于描述一个类目使用另外一个类目的信息和服务（比如操作）。
可以形容为“A use B”的概念。依赖表示为一条有向虚线段。

举例：一个学校有系和学生，每个系有自己的教员和课程。画出类图如下：



对象图：(object diagram)

对象图是类的实例，它表示系统执行在某一时间点的一个可能的快照(snapshot)。

(这个一般不会考)

状态图：(state diagram)

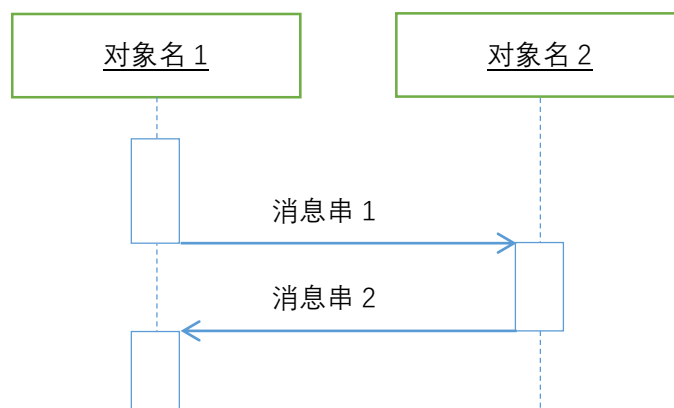
状态图通常是对类描述的补充，它说明该类的对象所有可能的状态 以及哪些事件将导致改变。

(这个一般不会考)

时序图：(sequence diagram)

时序图表示几个对象之间的动态协作关系。它主要用来分析对象之间发送消息的顺序。

消息：用于表达交互内容的术语。对象间的单向通信，从发送者到接受者的携带信息的控制流。

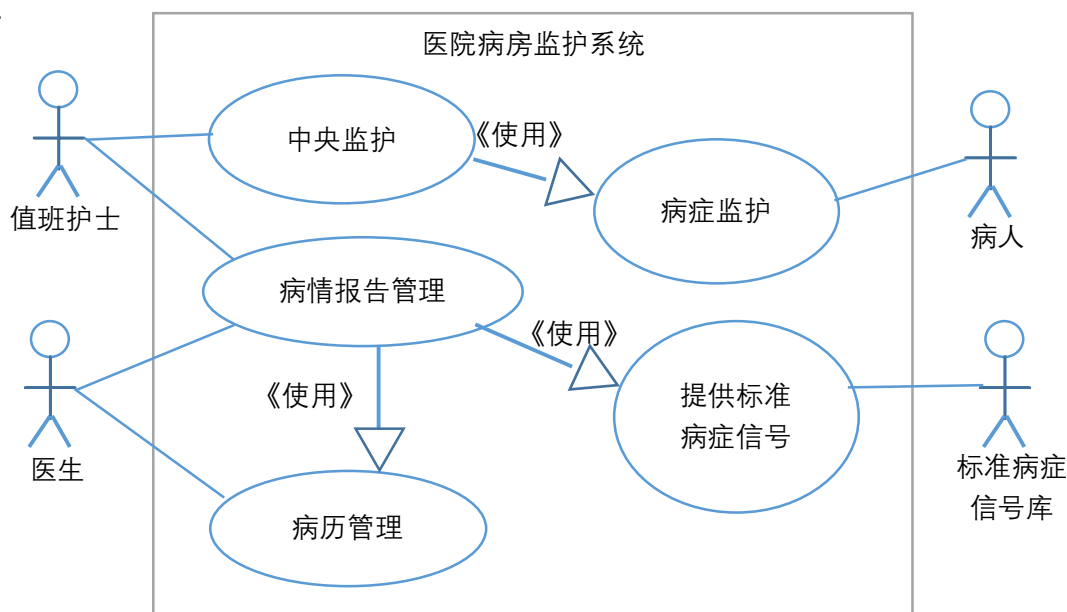


例题：医院病房监护系统

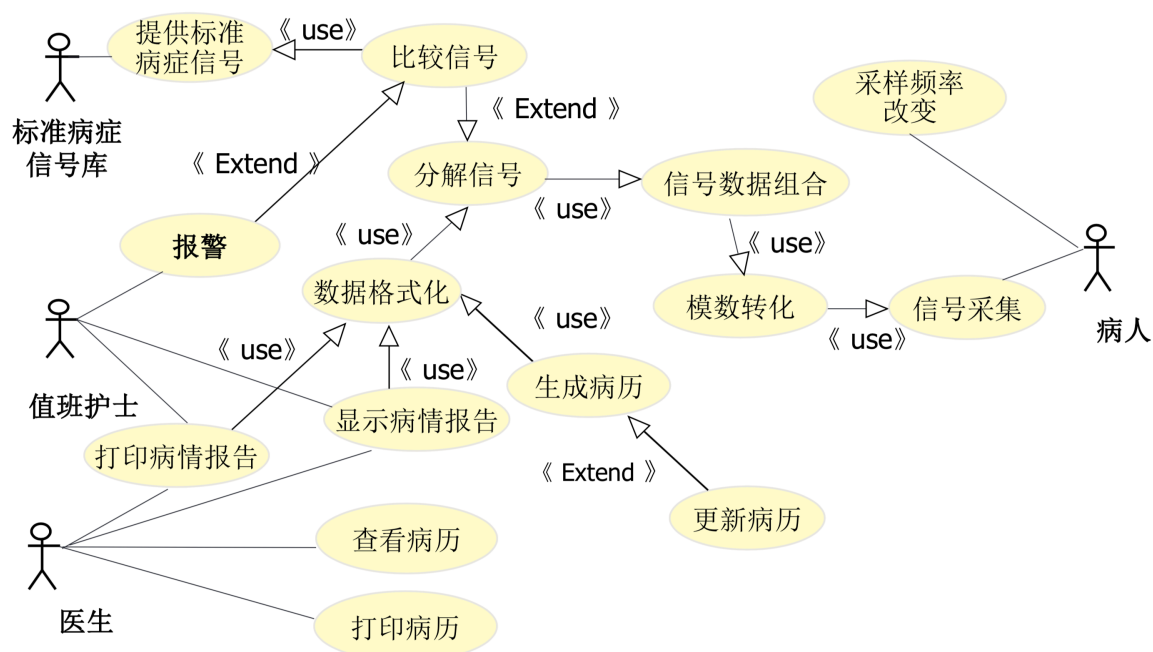
现有一医院病房监护系统，病症监视器安置在每个病房，将病人的病症信号实时传送到中央监视系统进行分析处理。在中心值班室里，值班护士使用中央监视系统对病员的情况进行监控，根据医生的要求随时打印病人的病情报告，定期更新病历，当病症出现异常时，系统会自动报警，并实时打印病人的病情报告，立即更新病历。

顶层用例图：

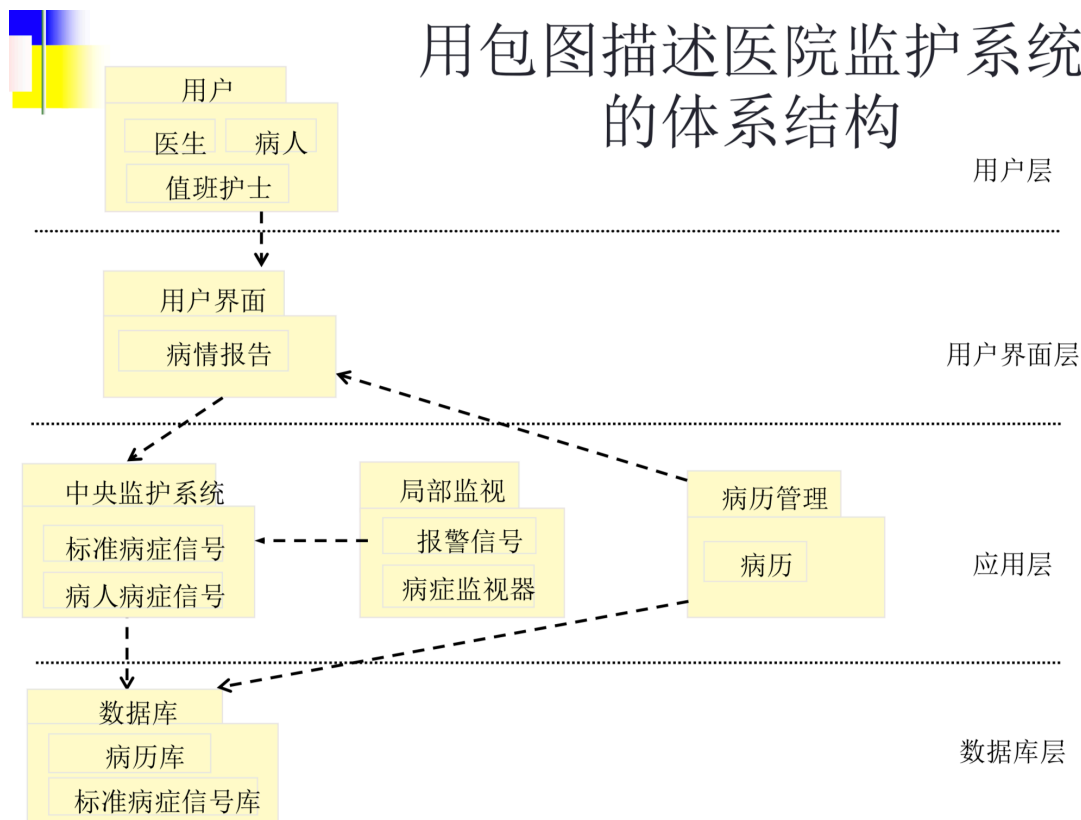
初步识别出系统的用例为:中央监护、病症监护、提供标准病症信号、病历管理、病情报告管理



细化的用例图：

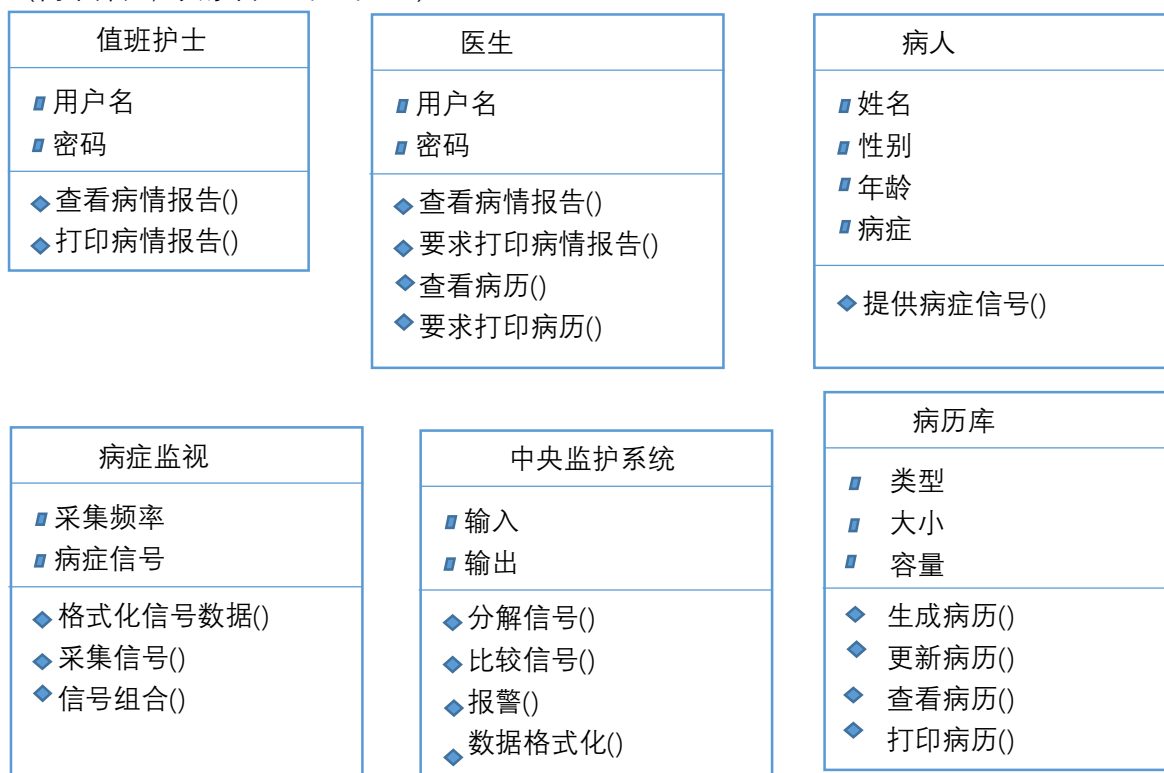


包图（考的可能性不大）：

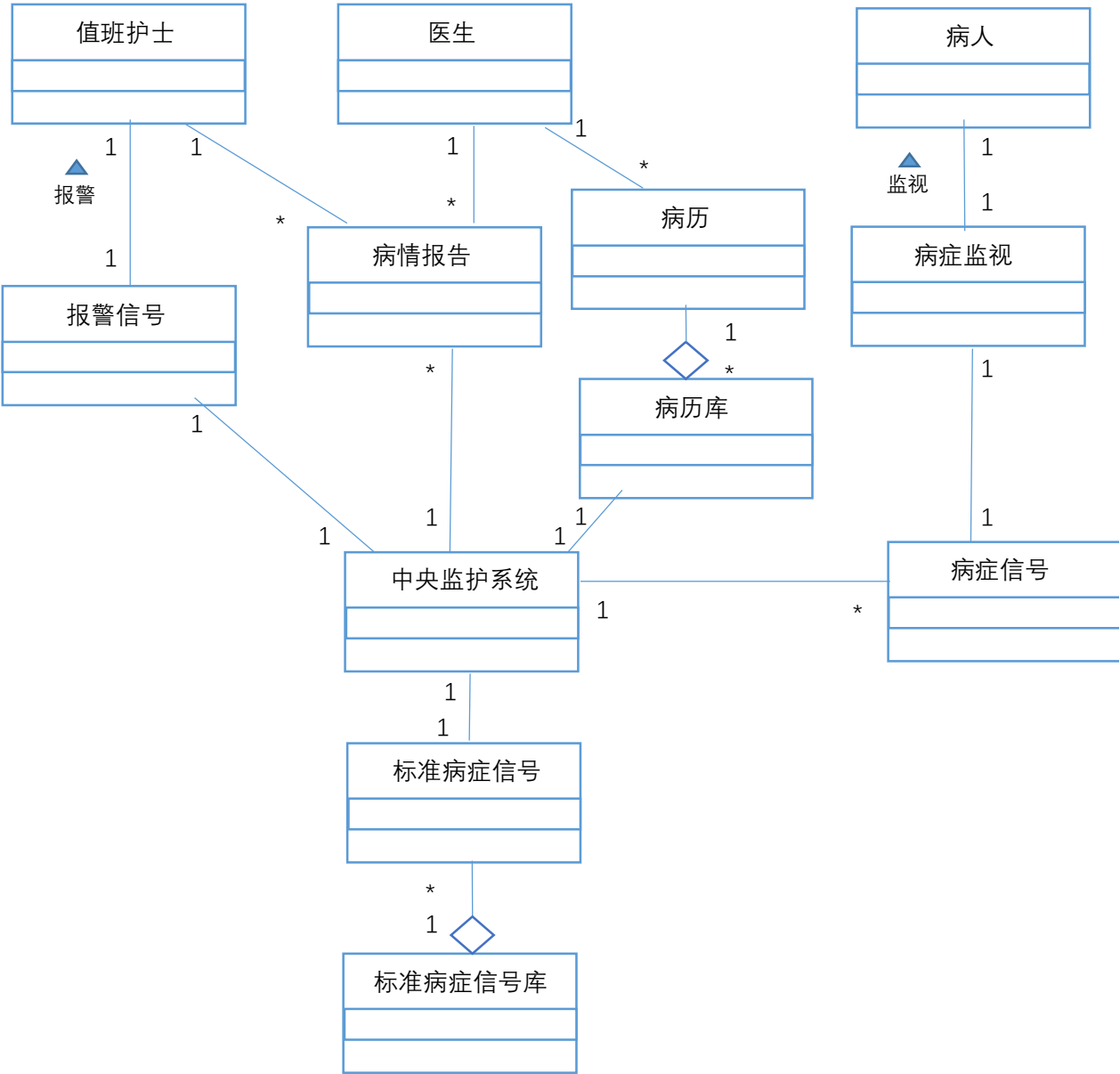


识别系统的类：

可以包括以下类：值班护士 医生 病人 病症监视 中央监护系统 报警信号 标准病症信号库 病历 病历库 病人病症信号 病情报告 标准病症信号
(例举部分，其余自己画一下吧。)



类图：标明类之间的关系：



时钟顺序图：(见最后例图)

结构化设计建模相关复习点

结构化分析方法(SA, Structure Analysis)是面向数据流进行需求分析的方法, 适合于数据处理类型软件的需求分析。

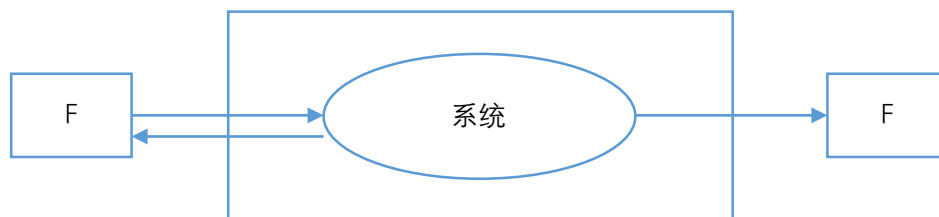
SA 常用的工具为: 数据流图 DFD, 数据字典 DD, 结构化英语, 判定树和判定表

数据流图: (DFD: Data Flow Diagram) 也称 Bubble Chart 或 Data Flow Graph, 是描述数据处理的工具。

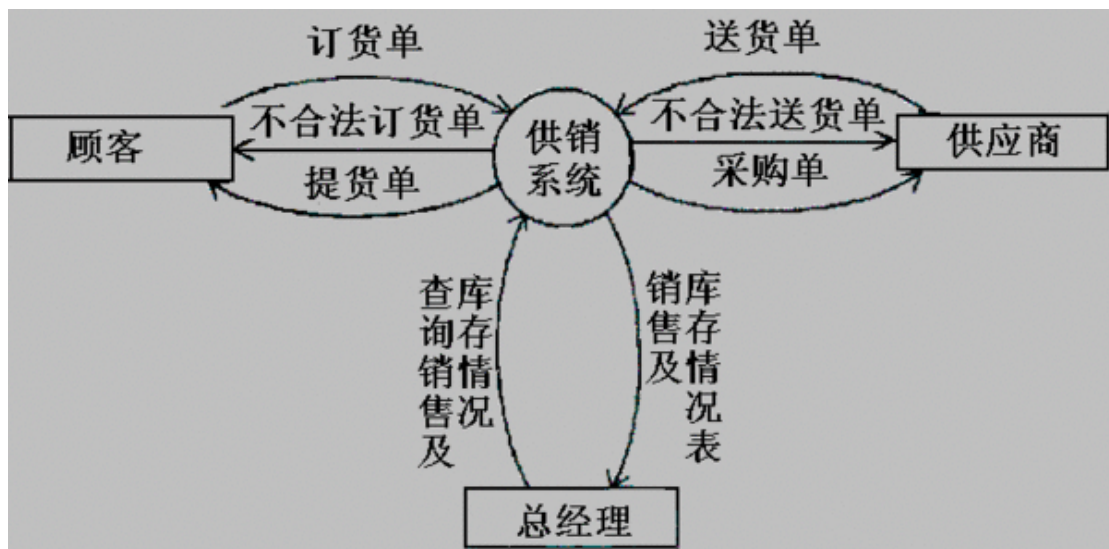
DFD 从数据传递和加工的角度, 以图形的方式刻画数据流从输入到输出的移动变换过程。

DFD 只反映系统必须完成的逻辑功能, 是一种功能模型。

顶层数据流图基本画法:



示例:



DD(Data Dictionary)就是用来描述数据流和加工的具体含义。

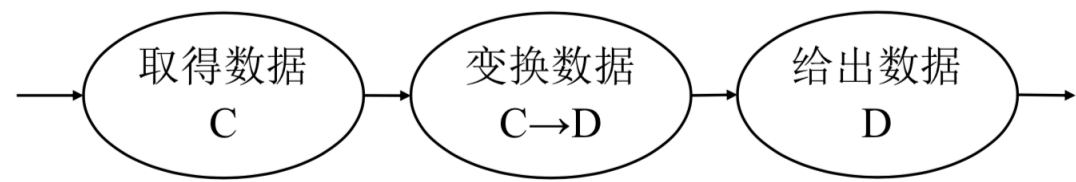
例:存折在 DD 中的定义:

存折=户名+所号+帐号+开户日+性质+(印密)+1{存取行}50

存储文件:存放的是什么数据

符号为: 存储文件名

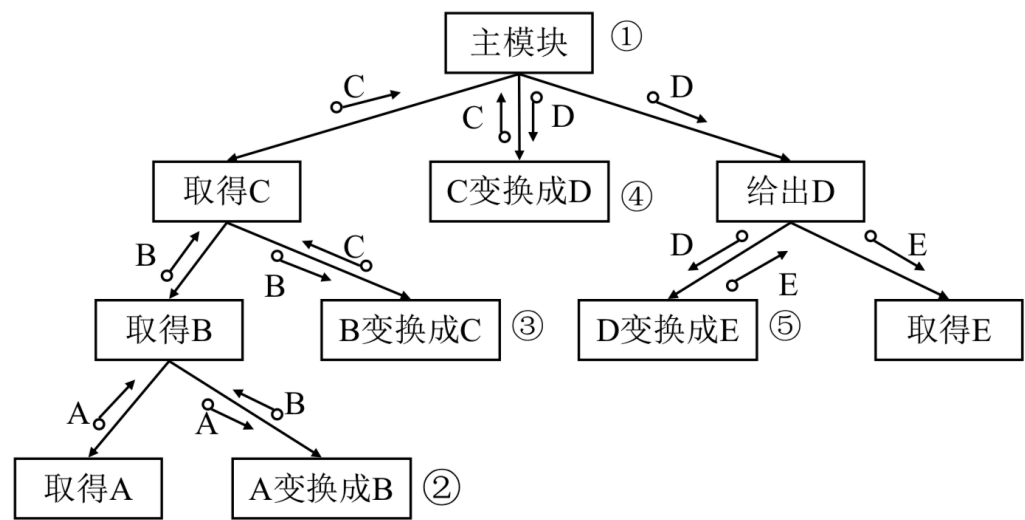
变换型数据流图：



由变换型数据流图得到系统结构图：

(注意：这个图里 C 变换成 D 过程的 C 和 D 箭头画反了，请注意)

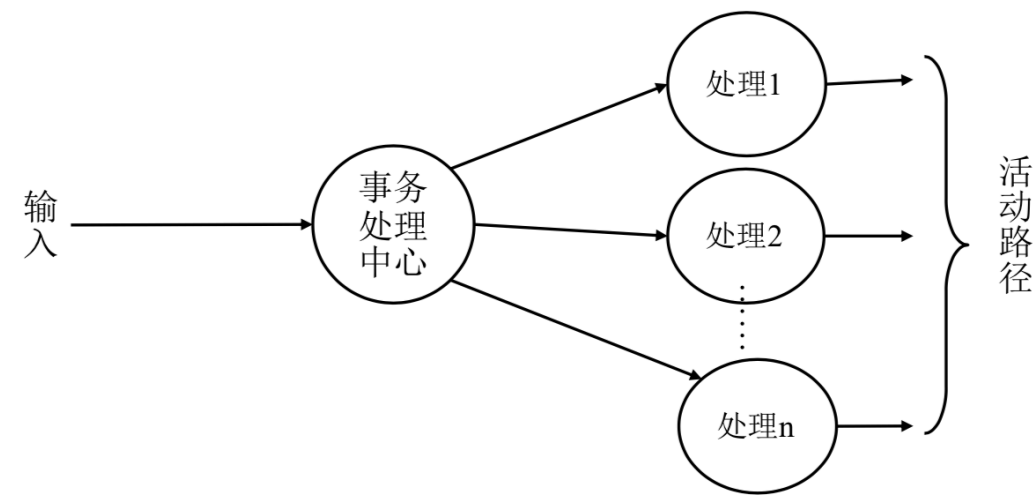
变换型的系统结构图



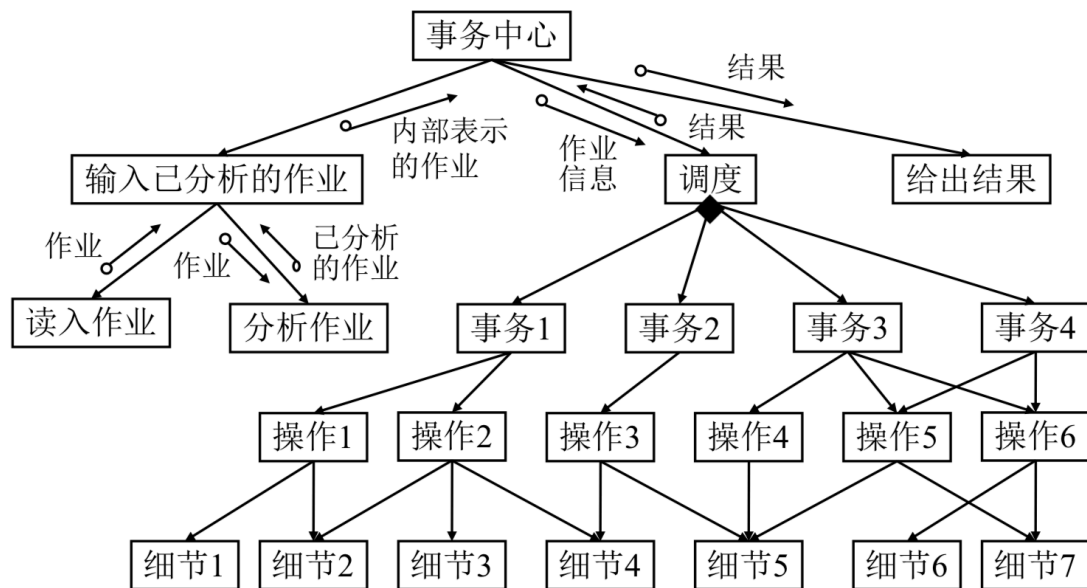
变换过程： $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$

事务型数据流图：

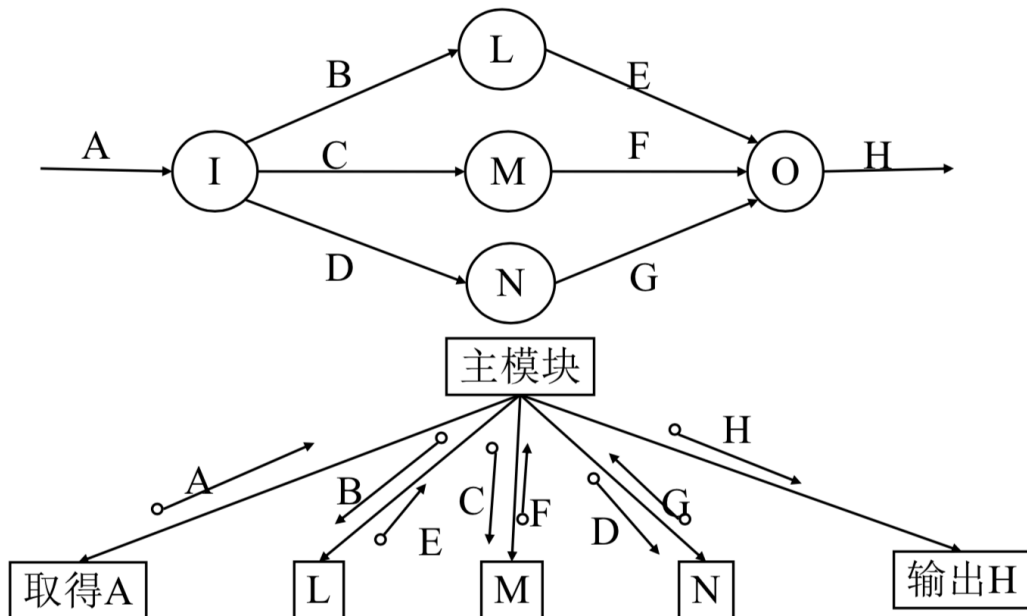
某个加工将它的输入分离成一串发散的数据流，形成许多活动路径，并根据输入的值选择其中一条路径。**事务处理中心：完成选择分派任务的部分。**



事务型系统结构图



事务型问题导出的系统结构图



例题：在一简化的图书管理系统中

图书管理员负责：

- (1) 购入新书时，录入新书的信息（分类目录号、流水号、书名、作者、单价、购书日期）；

(2) 当读者借书时，将读者的借书单信息输入系统，检查读者学号是否有效，若无效，拒绝借书，否则登记图书分类目录号、读者学号和借书日期等，写入借书文件中；

(3) 当读者还书时，根据还书单信息，从借书文件中查看读者的借书记录，看是否超期，若超期则给出罚款单；

(4) 当一些书过期没有保留价值时，从图书目录中删除相关记录。

读者负责：

(1) 借书时，填写借书单（姓名、学号、图书分类目录号）；

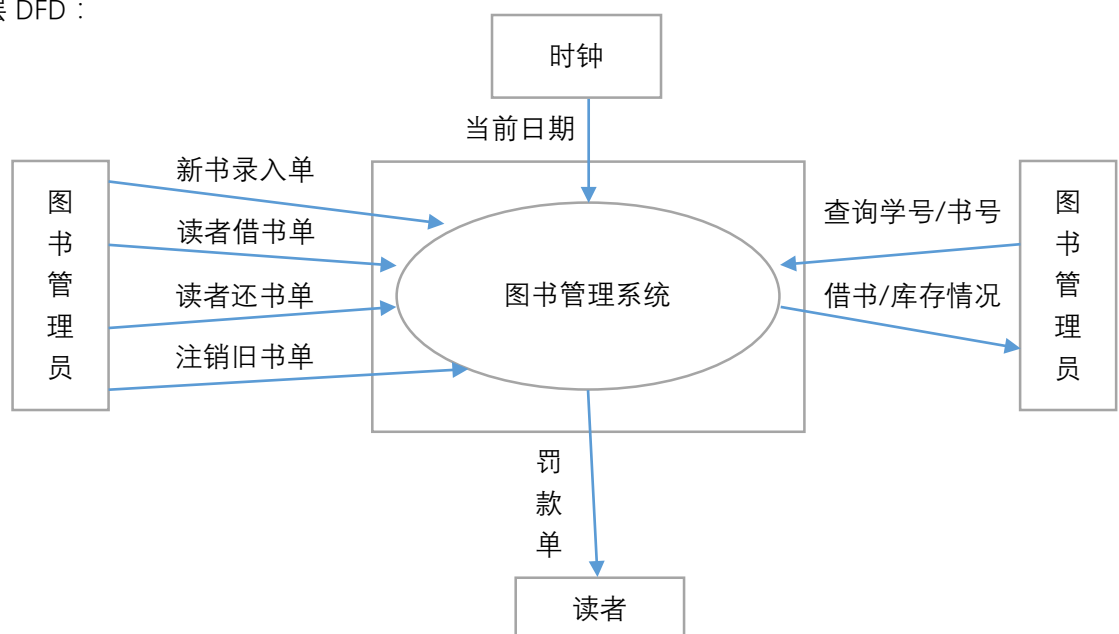
(2) 还书时，填写还书单（姓名、学号、图书分类目录号）；

图书馆领导负责：

(1) 根据读者学号，查询读者借书情况；

(2) 根据图书流水号，查询图书库存情况。

画出顶层 DFD：



给出顶层 DFD 的数据字典：

图书入库单=分类目录号+流水号+书名+作者+单价+购书日期

读者借书单=姓名+学号+图书分类目录号

读者换书单=姓名+学号+图书分类目录号

罚款单=姓名+金额

注销单=书名+分类目录号

读者=姓名+学号

图书库存=书名+分类目录号+库存数量

读者借书情况=姓名+书名+日期+分类目录号

给出可能用到的存储文件：

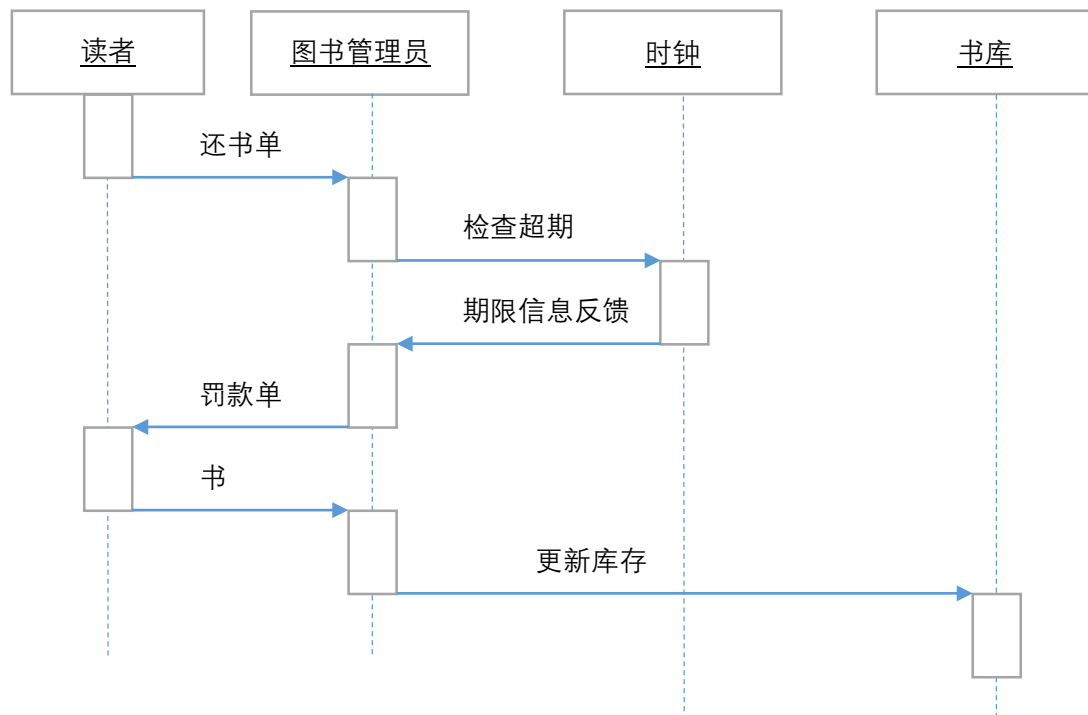
借书记录文件

图书记录文件

罚款记录文件

学号记录文件

还书过程顺序图（UML 时序图）：



顺序图的图例注意：

对象名需要画下划线。

生命线为虚线。

时序过程为方框。

箭头为简单消息（一般考试也只需要考到这个）。