## UML建模语言

UML建模语言不是一种方法，它独立于过程。

为什么叫UML（自己）

U：对多种OO建模方法进行了统一，形成规范

M：用于建立软件开发中的各种模型

L：一种可视化的图式语言

## UML主要构成

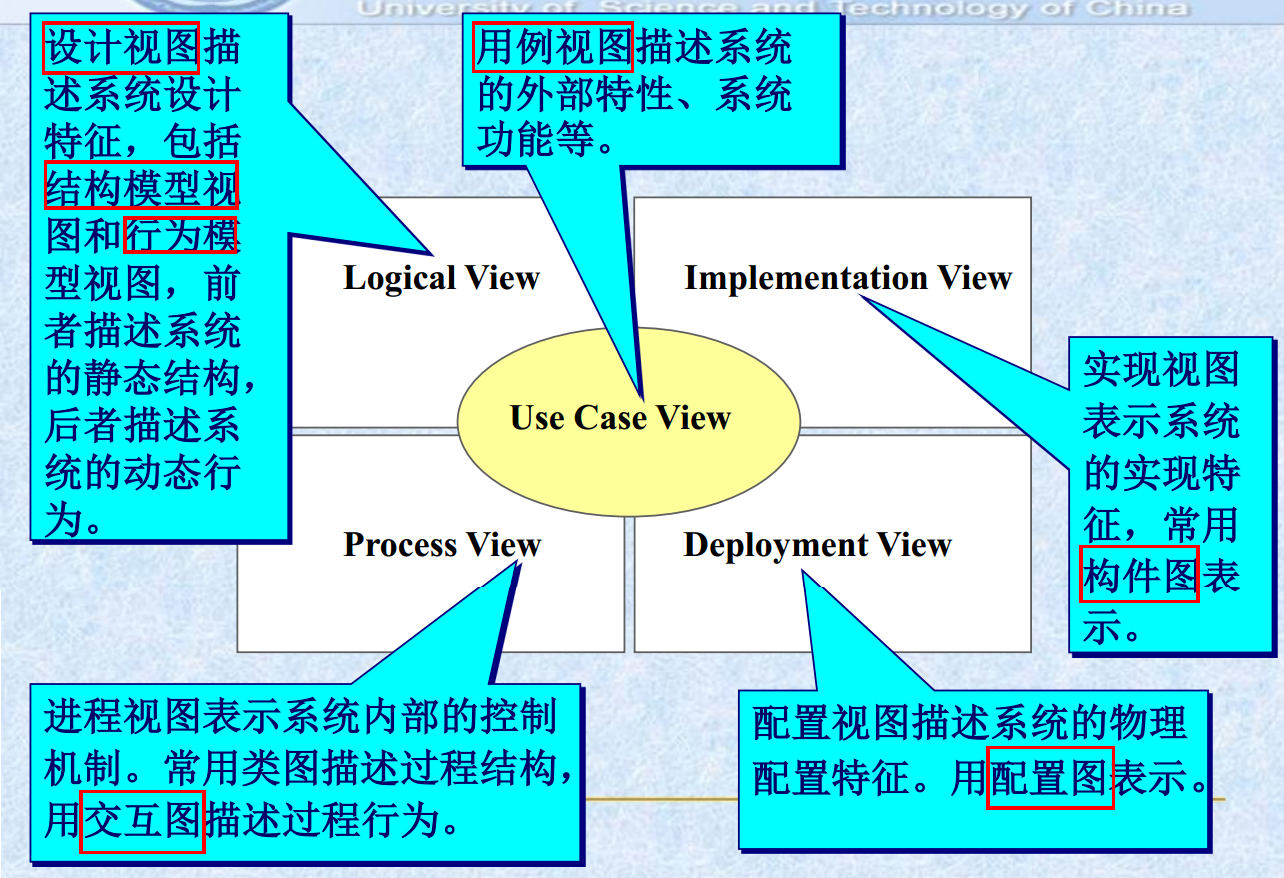
视图（逻辑的概念，不是物理的东西，不是图表，而是在某一层次上，对系统的抽象表示）

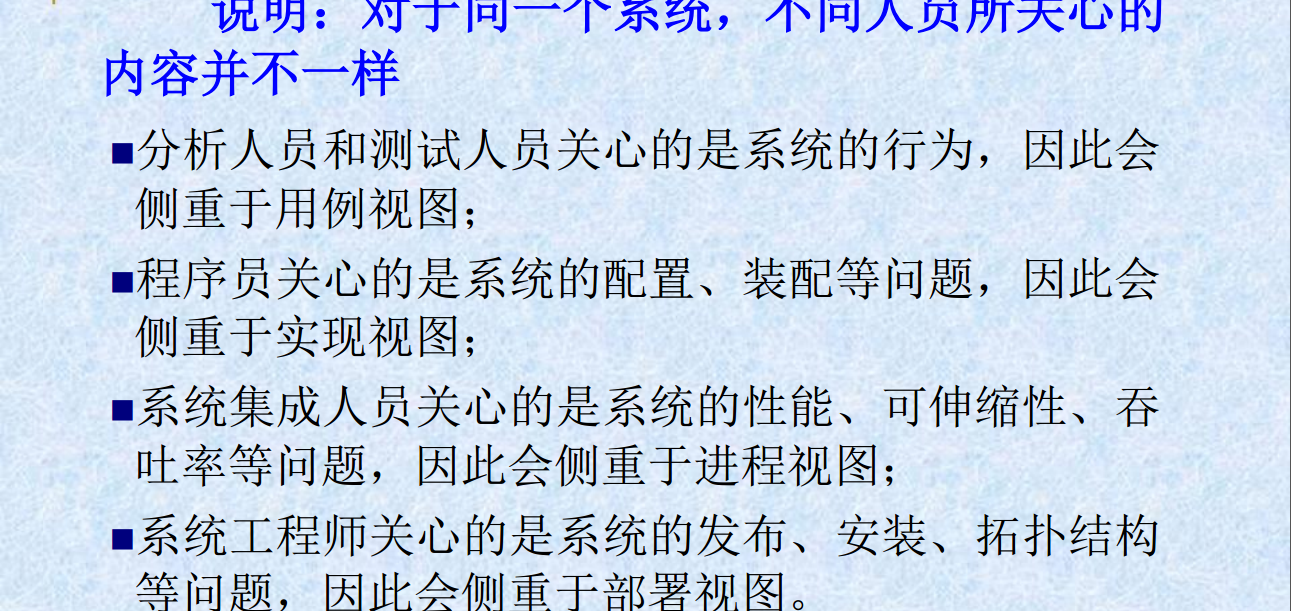
图

模型元素

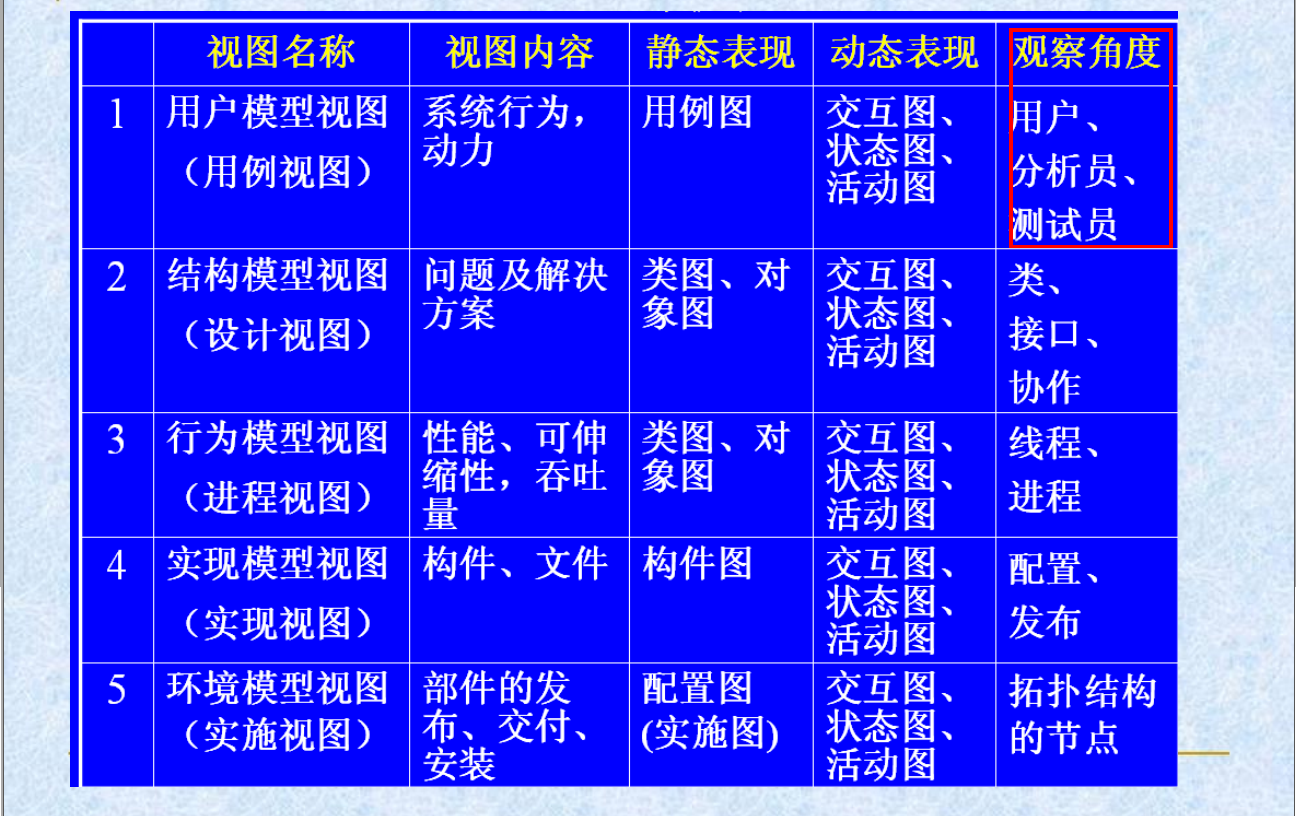
通用机制

## 视图

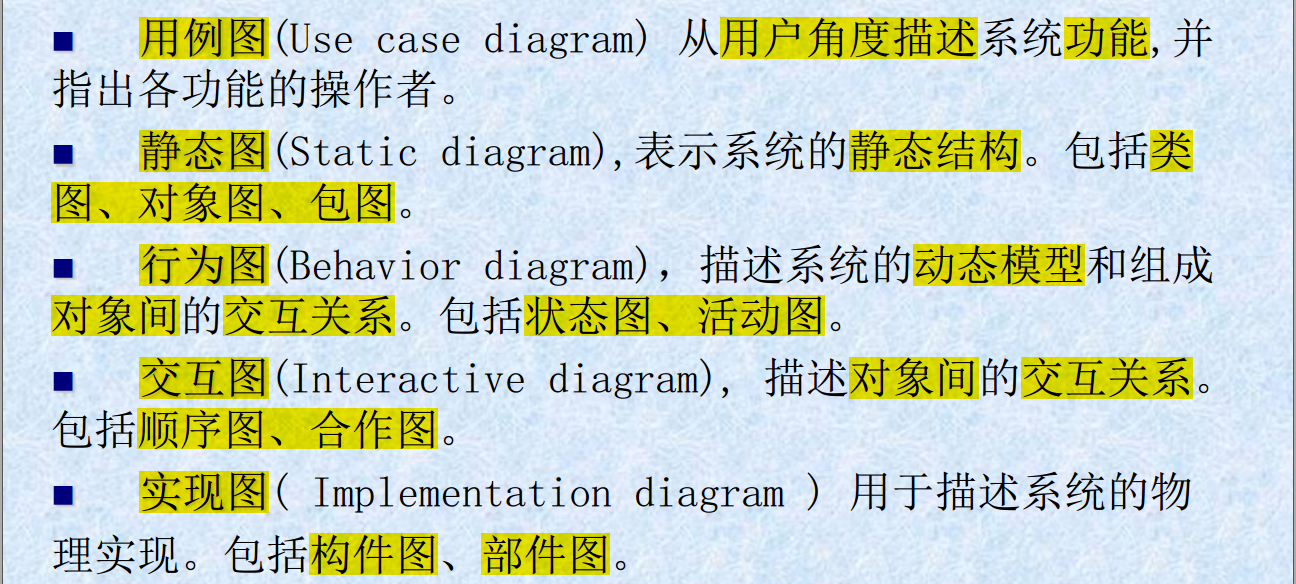


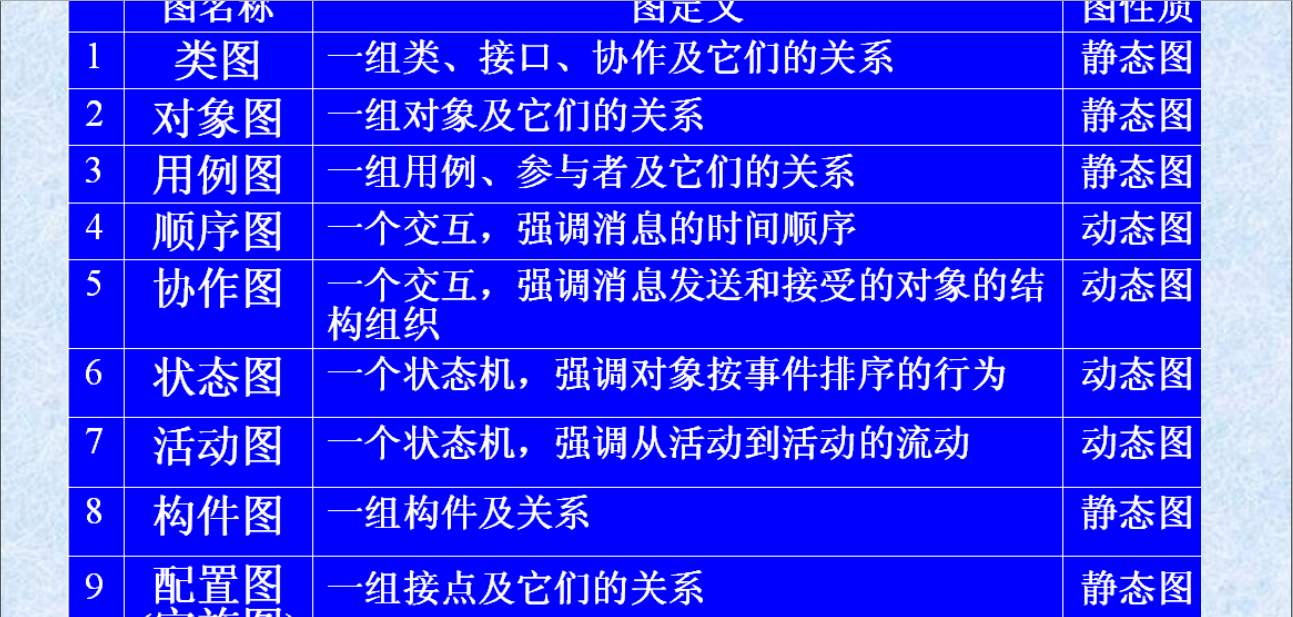


## UML中有哪些视图？静态表现和动态表现是什么？



## UML5种类型9种图





## UML9种图的画法（自己看，很重要）

## 用例间的三种关系

包含关系：包含其他用例，作为自身行为的一部分（被包含用例对执行者的间接性）

扩展关系：插入到基本用例中（触发的不确定性）

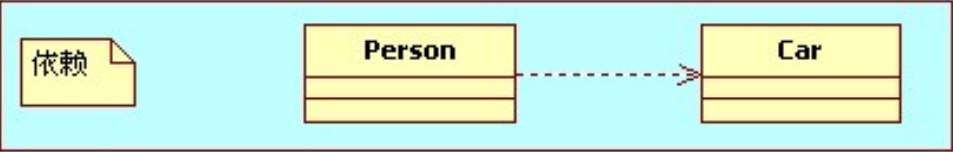
泛化关系：子用例

## 类与类之间的关系

关联关系（对象之间的关系没有这个）



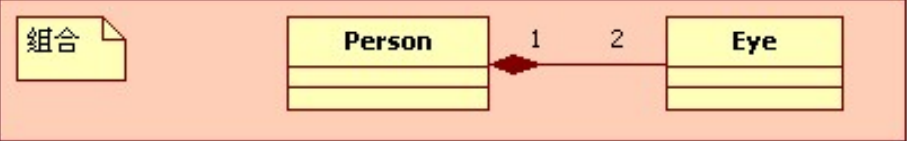
依赖关系



聚合关系（当A加入到B成为B的组成部分）



组合关系



以上四种关系的密程度

依赖<关联<聚合<组合

泛化（继承）

## 怎么画动态图

根据用例图，用例模板，前置条件，后置条件，事件流

## 类的类型（自己）

实体类

控制类

边界类

边界类是用户与边界类接口，控制类是边界类与实体类的媒介

## 对象图与类图使用相同的符号，只是在对象名下面加下划线

## 构件图

构件

遵从一组接口且提供其实现的物理的、可替换的部分

能够完成独立功能，是软件系统的组成部分

使软件复用成为真正意义上的可能

如：科大讯飞的语音组件

## 顺序图

对象之间消息的顺序，对象之间的交互

重点不是消息本身，而是消息产生的顺序

一些主要元素

对象

生命线：从对象图标向下延伸的一条虚线

激活期：对象执行一个动作的期间，长方形框

消息：生命线之间的消息箭头表示，按垂直线从上到下的顺序

消息

同步调用：实心箭头（消息发送，停止等待）

接收者是被动对象，即需要消息驱动才能执行动作的对象

一般必须有一个配对的返回消息

异步消息：棍形箭头（消息发送，继续自己的活动，不等待，降低系统耦合性）

返回消息：虚线箭头（只是针对异步消息，同步消息一般必须有一个返回消息不是此处）

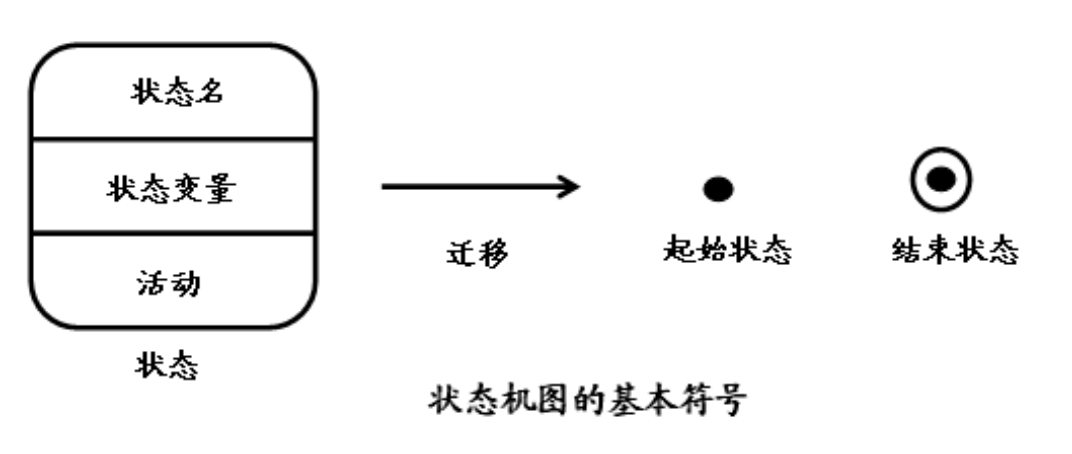
如果是过程调用返回，则返回消息是隐含的，所以返回消息可省略

如果非过程调用，如有消息返回，必须明确表示出来

## 协作图

顺序图清晰地展示了时间顺序，但不明确显示对象之间的关系；协作图清晰展示对象之间的关系（有哪些消息），消息顺序必须通过顺序号指明

## 状态图



状态变量：类的属性，临时变量

活动：

entry事件：进入状态的特定动作

exit事件：退出状态的特定动作

do事件：在该状态时执行的动作

所有对象都有状态（有可能辨析题）

## 活动图

活动流，描述完成一个操作所需要的活动。等同于流程图和数据流图DFD

实心圆表示活动的起点，

带边框的实心圆表示终点

圆角矩形表示执行的过程或活动

箭头表示活动之间的转换

粗线条表示可能会并行进行的过程的开始和结束

## 泳道图

活动图的变形

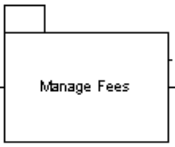
## 部署图（配置图）

描述了处理器，设备，软件构件运行时的体系结构

## 包图

设计良好的包高内聚，低耦合的，并且对其内容的访问具有良好的控制

一个“包图”可以是任何一种UML图组成，通常是UML用例图或类图



## 注UML这一章和设计工程这一章，很容易出大题，要重点看