





## 本单元知识大纲

---

- “对象”定义
- 面向对象的软件开发模型
- 用例图建模基础

# 1. 什么是对象？

在现实世界中有意义的、与所要解决的问题有关系的任何事物都可以作为**对象**，包括具体的物理实体的抽象、人为的概念、任何有明确边界和意义的东西。

如：一名职工、一本图书、贷款、生产计划、一场演出等。



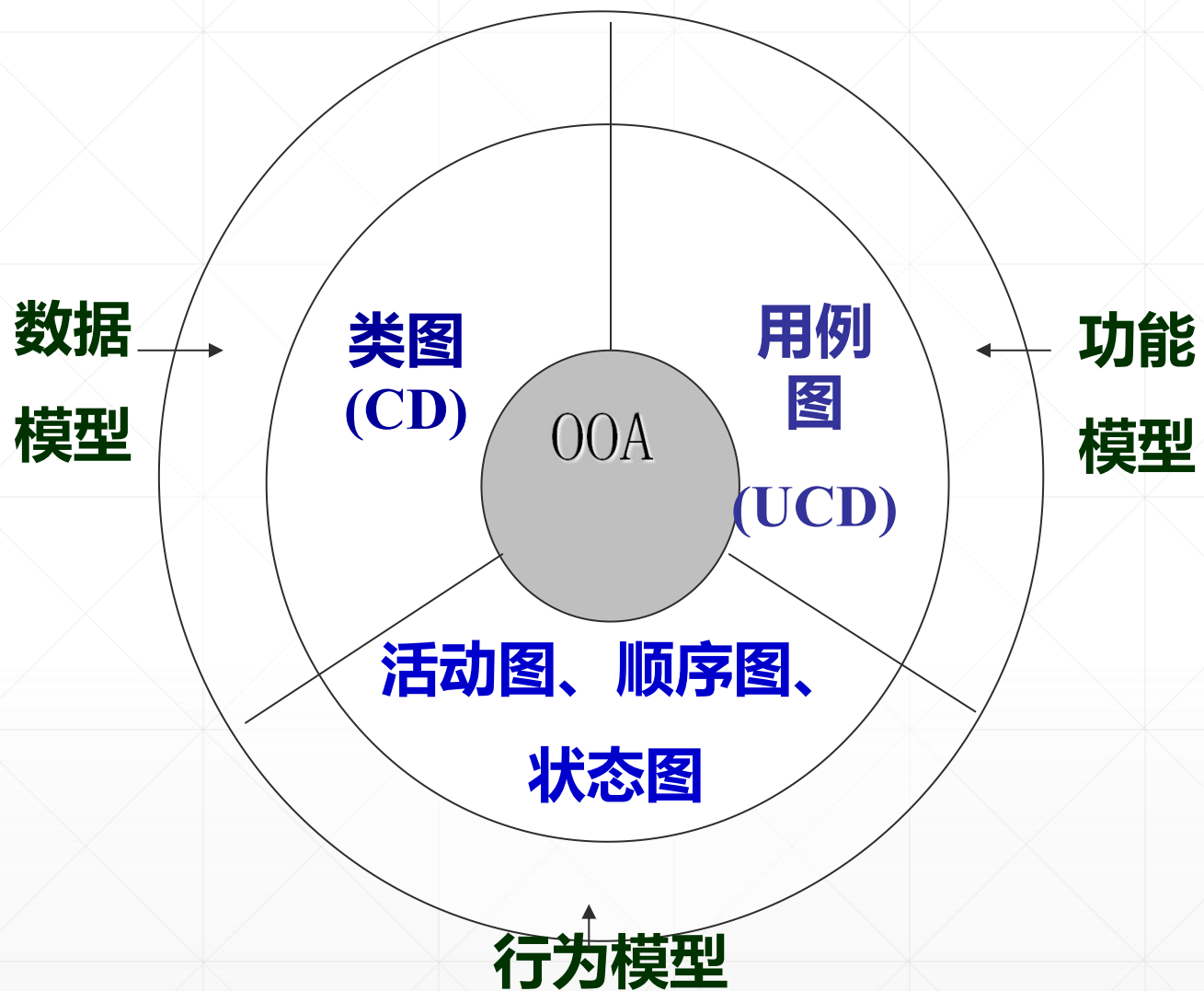
# 几种著名的面向对象方法

- Booch 方法 ( 1991 )
- Coad — Yourdon 方法 ( 1991 )
- Rumbaugh 方法 ( 简称 OMT ) ( object Modeling Technology , 1991 )
- Jacobson 方法 ( 简称 OOSE , 1992 )
- 由 Rumbaugh 、 Booch 、 Jacobson 提出的统一建模语言 ( 简称 UML ) ( Unify Modeling Language , 1994 )

一种可视化建模语言，能描述开发需要的各种视图，并以此为基础组建系统。

## 2. 面向对象的软件开发模型

- **数据模型（对象模型）：**  
描述系统数据结构的对象模型;
- **行为模型（动态模型）**  
描述系统控制结构
- **功能模型**  
描述系统功能。



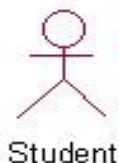
一个典型的软件系统使用**数据结构（对象模型）**，**执行操作（动态模型）**，并且完成**数据值的变化（功能模型）**。

### 3. 功能模型——用例图

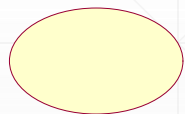
---

# ( 1 ) 用例图的基本图形符号

- 用例建模用于描述系统需求，把系统当作黑盒，从用户的角度，描述系统的场景。主要图形元素有以下几个：



- 参与者：是指外部用户或外部实体在系统中扮演的角色。可以是**一个人**、**一个计算机子系统**、**硬件设备**或者**时间**等角色



- 用例：对**一组**动作序列的描述，系统通过执行**这一组动作序列**为参与者**产生**一个可观察的结果。用例名往往用动宾结构命名。

- 
- 执行关联：参与者（Actor）执行用例（Use Case）之间的关系

## (2) 用例建模的过程

- 建立用例模型的顺序是：

1. 确定谁会**直接使用**该系统。这些都是参与者(Actor)。
  2. 选取其中一个参与者。
  3. 定义该参与者**希望系统做什么**，参与者希望系统做的每件事成为一个用例。
  4. 对每件事来说，**何时**参与者会使用系统，通常会**发生什么**，这就是用例的基本过程。
  5. 描述该用例的**基本过程**。
  6. 考虑一些可变情况，把他们创建为**扩展用例**。
  7. 复审不同用例的描述，找出其中的相同点，**抽出相同点**作为共同的用例。
  8. 重复步骤2~7找出每一个用例。
-





# 感谢观看！

---

授课教师：吴晓华    电子邮箱：[wxhcshua@126.com](mailto:wxhcshua@126.com)