软件工程：第十三章 UML简介

导学目标

- 了解UML的基础知识
- 了解UML的视图和构成
- 掌握常见的有关面向对象的考题

第一节 UML概述

UML是一个通用的**可视化建模语言**，是用于对软件进行描述、可视化处理、构造和建立软件系统支配的文档

制品：软件开发过程中的各种产物，如模型、源代码、测试用例等

UML适用于各种软件开发方法、软件生命周期的各个阶段、各种应用领域以及各种开发工具

UML目的是为了支持面向对象开发过程而设计的

1、什么是UML

- UML是一种语言，定义了一系列的图形符号来描述软件系统
- 图形符号有清晰的语义和严格的语法
- 图形符号及背后的语义语法组成了一个标准
- UML描述了一个系统的静态结构（属性）和动态行为（服务、方法）
- 通过静态属性定义系统中的对象的属性和操作以及这些对象之间的相互关系
- 动态行为定义了对象的时间特性和对象为完成目标而相互进行通信的机制
- UML标准没有定义一个标准的开发过程，它是为了支持面向对象开发过程而设计的
- UML不是一门程序设计语言，但可以利用代码生成器工具将UML模型转换为多种程序设计语言代码

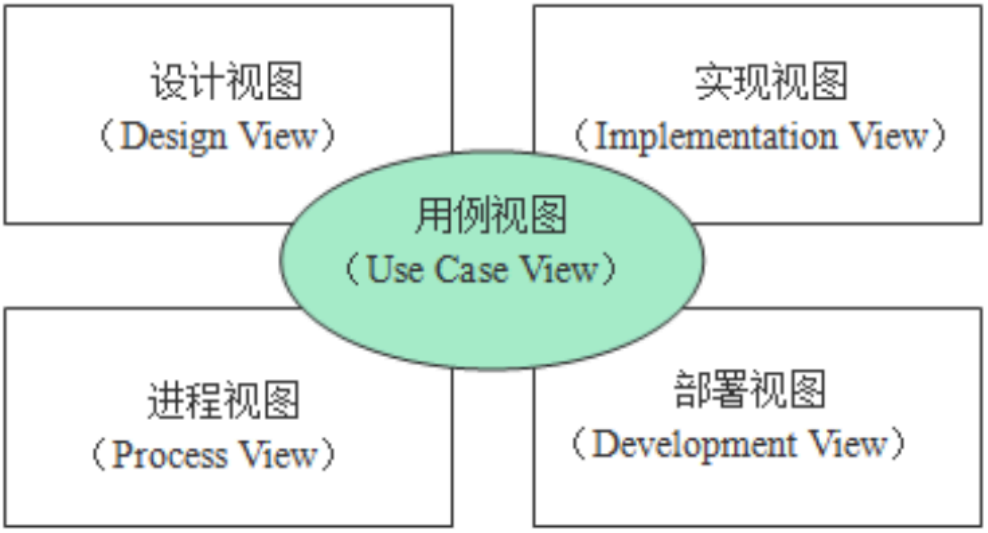
2、UML的特点

- 统一的标准(被OMG所认定的建模语言标准)
- 面向对象(UML是支持面向对象软件开发的建模语言)
- 可视化，表现能力强(图形符号)
- 独立开发过程，UML不依赖与特定的软件开发过程
- 概念明确，建模简洁，推行结构清晰，容易掌握和使用

第二节 UML视图

1、视图组成

- 设计视图：设计词汇、功能描述
- 实现视图：系统组装、配置管理
- 进程视图：性能、稳定性和吞吐率
- 部属视图：系统拓扑、分布、安装



注意：不同的视图突出特定的参与群体所关心的系统的不同方面，通过合并所有五个视图中得到的信息就可以形成系统的完整描述

2、用例视图（核心）

- 用于支持软件系统的需求分析、定义边界、关注功能
- 从系统参与者的角度去描述系统的外部行为和动态功能
- 用例视图的使用者是客户、开发人员及测试人员
- 用例图是核心，该视图定义了系统的需求，因此约束了描述系统设计和构造的某些方面的所有其他视图
- 通过用例图可以校验最终系统

3、设计视图

- 定义了系统的实现逻辑
- 设计视图的使用者主要是开发人员和设计者
- 它由程序组件的定义，主要是类、类所包含的数据、类的行为以及类之间交互的说明组成
- 它的图形模型包括：类图、对象图、状态图、顺序图、协作图及活动图

4、进程视图

- 描述系统的实现模块及它们之间的依赖关系
- 进程视图的使用者主要是开发人员
- 进程视图包括形成并发和同步机制的进程和线程

5、实现视图

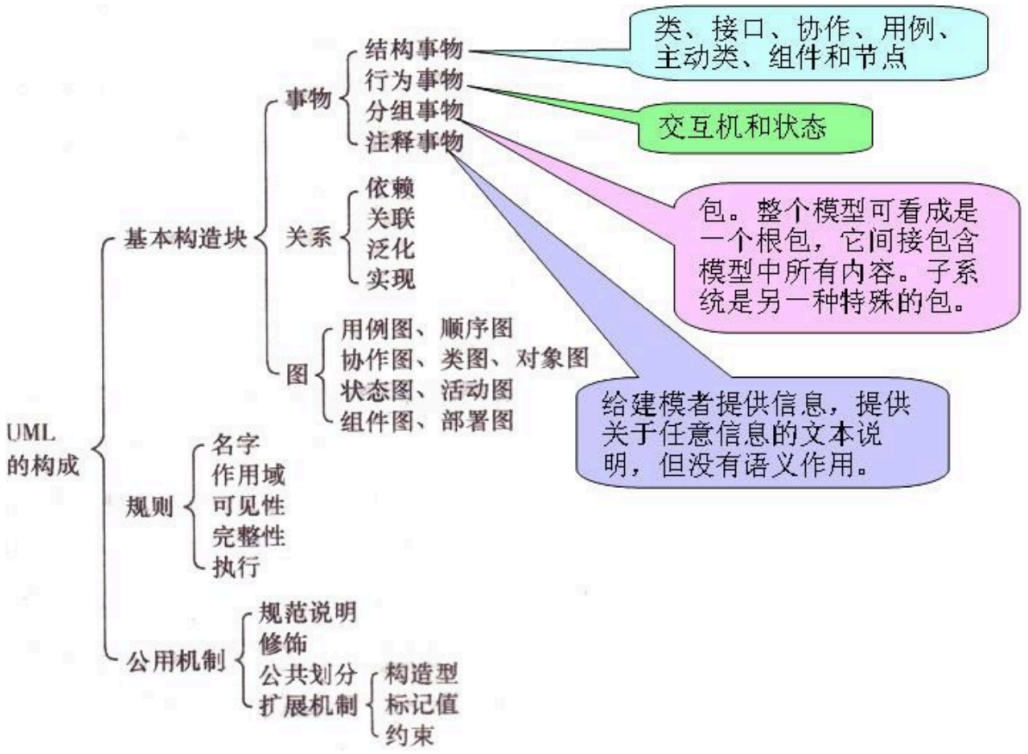
- 描述构造系统的物理组件，这些组件包括如何执行文件、代码库和数据库等内容
- 实现视图的使用者主要是开发人员和系统集成人员
- 该视图由动态图(状态图、协作图、活动图)和实现图(组件图和部署图)组成

6、部署视图

- 描述物理组件如何在系统运行的实际环境(如计算机网路)中分布
- 部署视图的使用者是开发人员、系统集成人员和测试人员
- 该视图由部署图表示

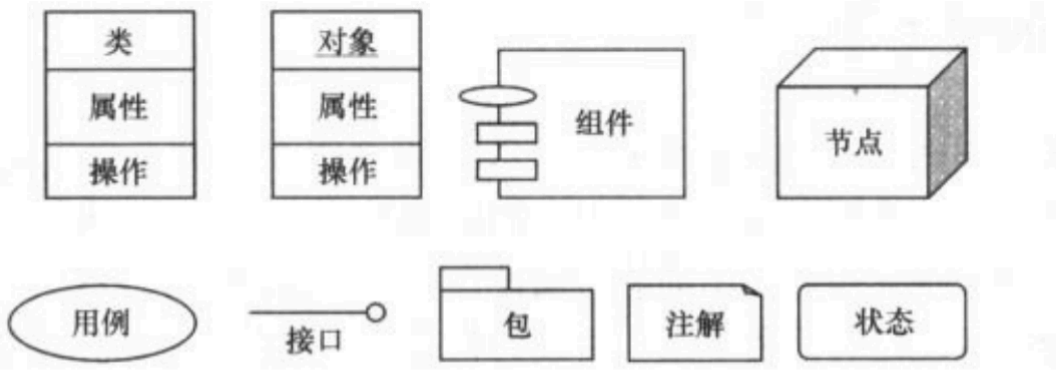
第三节 UML构成

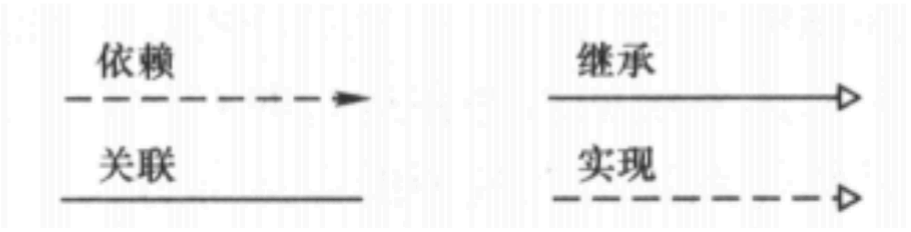
1、UML构成



2、UML的模型元素

- UML把在图中使用的概念统称为模型元素
- 图形符号隐含表示了模型元素的语法和语义
- 模型元素描述了系统结构(静态特征)和行为(动态特征)
- UML中定义了两类模型元素：
  - 用于表示模型中的某个**概念**（类、对象、接口）
  - 用于表示模型之间的相互**关系**（依赖、关联、泛化、实现）
- 示例图





3、UML模型规则

- 名字：任何一个UML成员都必须包含一个名字
- 作用域：UML成员所定义的内容起作用的上下文环境
- 可见性：UML成员被其他成员引用的方式
- 完整性：UML成员之间相互连接的合法性和一致性
- 运行属性：UML成员在运行时的特性

注意：一个完整的UML模型必须对上述内容给出完整的解释