

7统一建模语言UML (5*)

UML是通过图形化的表示机制进行**面向对象分析和设计**，并进行统一的、标准化的视图、图、模型元素和通用机制来刻画面向对象方法。

7.1UML的发展

UML的构成 (四部分)

- 1.视图 从某个角度观察到的系统叫视图
- 2.图
 - 视图由图组成
 - UML有9类图
 - 用例图
 - 类图 (对象图)
 - 包图
 - 状态图
 - 活动图
 - 顺序图
 - 协作图
 - 构建图
 - 部署图
- 3.模型元素 表示面向对象中的概念, 如类、对象、接口、消息和组件等
- 4.通用机制 通用机制用于描述系统的其他信息, 例如注释、通用系统的语义拓展

7.4UML的图和模型元素 (5*)

- 7.4.1 用例图
 - 由参与者、用例、他们之间的关系共同构成, 用于描述系统功能的图。
 - 从系统外部描述系统功能及其功能间关系
 - 主要用于子系统、包、类等功能行为描述
 - 用例图不负责描述功能实现的细节和性能的约束
 - 参与者 不仅仅指用户, 系统外部的、所有与系统交互的角色。
 - 用例 对一组动作序列的抽象描述, 系统执行动作序列, 产生响应结果。
 - 系统边界
 - 图形元素
 - 关系 描述用例图之间的关联
 - `<<use>>`: 使用关系, 表示参与者对用例的操作, 一般可省略
 - `<<include>>`: 包含关系, 表示一个用例行为包括另一个用例的行为。前者基本用例, 后者为拓展用例
 - `<<extend>>`: 扩展关系, 表示扩展用例对基本用例的特殊服务。
 - 泛化关系: 表示不同参与者或者不同用例间的继承关系
- 7.4.2 类图
 - 类图用于描述类的属性、方法和类间关系。属性和方法是类的内部结构, 关系是类间的关联。
 - 它们用于定义UML的静态模型。
 - 类的内部结构: 类名、属性、方法以及他们的可见性
- 7.4.3 包图
 - 对UML中的用例图、类图、UML关系等模型元素的封装, 用于描述具有相似功能的模型元素的组合。
 - 包之间的关系有两类:
 - 依赖关系: 一个包中引入另一个包的输出关系
 - 泛化关系: 定义包的继承关系, 体现 具有与类库相似的包的家族
- 7.4.4 状态图
- 7.4.5 活动图 描述用例或场景的活动顺序, 或者描述从一个活动到另一个活动的控制流
- 7.4.6 顺序图
 - 横坐标表示不同对象, 纵坐标表示时间, 说明某一时刻对象是否还存在
 - 消息:
 - 简单消息: 控制流, 描述对象间控制的转换
 - 同步消息: : 等待-继续“控制流
 - 异步消息: “不等待”控制流
 - 顺序图不仅仅描述系统交互或者操作过程, 对于复杂的系统交互或操作过程, 还可以增加结构化控制。
- 7.4.7 协作图 描述类和类间关系, 反映的是通过一组类的共同合作来完成的系统功能, 又称为合作图
- 7.4.8 构件图
- 7.4.9 配置图

7.5UML的关系 (4*)

- 7.5.1 关联关系
 - 描述类与类之间的关系构成。
 - 关联关系通常是双向的, 关联的多个类之间的彼此都能相互通信。
 - 1. 普通关联 最常见的关联关系, 有连接关系就能通用普通关联表示
 - 2. 限定关联 限定关联用于描述一对多、多对多的关联关系, 通过限定关联, 可以将多对多关系转化为多个一对多关系, 又可以将多个一对多关系转化为多个一对一的关系。
 - 限定词放在限定的类旁边, 用矩形或者(), 作为对类的约束。
 - 3. 关联类 虚线
 - 4. 递归关联 类间关系发生在单个类自身上, 类和它自身有关联关系。
 - 5. 聚合关联 又称聚集, 描述多个类之间是整体和部分的关联关系
 - 共享聚合 “部分”类的对象可以同时成为多个“整体”类的对象
 - 复合聚合 “部分”类的对象完全参加一个“整体”类的对象
- 7.5.2 泛化关系
 - 泛化关系用于描述一个类自动具有另一个类的属性和方法的机制
 - 又被成为继承关系, 带有空心的三角形的直线连接
 - 1. 普通泛化 一般意义上的继承
 - 2. 受限泛化 对泛化关系加约束, 强化泛化关系的语义信息
 - 交叠泛化
 - 不相交泛化
 - 完全泛化
 - 不完全泛化
- 7.5.3 依赖关系 虚线箭头表述
- 7.5.4 实现关系 描述同一模型间的不同细化过程, 体现的是类间的语义关联。

7.6UML的通用机制 (3*)

- 常见的通用机制包含: 修饰, 注释, 规格说明
1. 修饰 用于增加UML模型元素的语义信息
 2. 注释 用缺角矩形表示, 通过虚线与被注释的连接
 3. 规格说明
 - 规格说明是对UML图形的一个标准化描述, 既增加了事物的图形文字内容, 又使得UML提供了可视化视图以及与之有关的语法和描述。
 - 非标准方式
 4. 扩展机制
 - UML的修饰、注释、规格说明体现的就是UML的扩展机制, 主要是对事物外部形象的扩展
 - 为了既适应大型项目开发建模过程, 又适应描述具体的功能、组织和个人, UML还提供了语义信息更加丰富的扩展机制。
 - 构造型 用于UML已有的模型元素的基础上, 通过增加语义信息或说明来建立一种新的元素。
 - 分类
 - 标签型 通过增加“属性-值”来进一步描述问题中的事物
 - 约束型 在UML的基础上, 增加对事物或者事物间关系因该满足的规则或语义。