# 概述

## UML简介

UML (Unified Modeling Language)为面向对象软件设计提供统一的、标准的、可视化的建模语言。适用于描述以用例为驱动，以体系结构为中心的软件设计的全过程。

UML的定义包括UML语义和UML表示法两个部分。

UML语义：UML对语义的描述使开发者能在语义上取得一致认识，消除了因人而异的表达方法所造成的影响。

UML表示法：UML表示法定义UML符号的表示法，为开发者或开发工具使用这图形符号和文本语法为系统建模提供了标准。

## UML的构成

事物(Things)：UML模型中最基本的构成元素，是具有代表性的成分的抽象

关系(Relationships)：关系把事物紧密联系在一起

图(Diagrams )：图是事物和关系的可视化表示

### UML事物

UML包含4种事物：构件事物、行为事物、分组事物、注释事物。

1. **构件事物： UML模型的静态部分，描述概念或物理元素**

它包括以下几种：

* 类：具有相同属性相同操作 相同关系相同语义的对象的描述
* 接口：描述元素的外部可见行为，即服务集合的定义说明
* 协作：描述了一组事物间的相互作用的集合
* 用例：代表一个系统或系统的一部分行为，是一组动作序列的集合
* 构件：系统中物理存在，可替换的部件
* 节点：运行时存在的物理元素

另外，参与者、信号应用、文档库、页表等都是上述基本事物的变体

1. **行为事物：UML模型图的动态部分，描述跨越空间和时间的行为**

* 交互：实现某功能的一组构件事物之间的消息的集合，涉及消息、动作序列、链接
* 状态机：描述事物或交互在生命周期内响应事件所经历的状态序列

1. **分组事物： UML模型图的组织部分，描述事物的组织结构**

* 包： 把元素组织成组的机制

1. **注释事物： UML模型的解释部分，用来对模型中的元素进行说明，解释**

* 注解 ：对元素进行约束或解释的简单符号

### UML关系

1. **依赖**

依赖(dependency)是两个事物之间的语义关系，其中一个事物(独立事物)发生变化，会影响到另一个事物(依赖事物)的语义

1. **关联**

关联(association)是一种结构关系，它指明一个事物的对象与另一个事物的对象间的联系。

1. **泛化**

泛化(generalization)是一种特殊/一般的关系。也可以看作是常说的继承关系

1. **实现**

实现(realization)是类元之间的语义关系，其中的一个类元指定了由另一个类元保证执行的契约

### UML图

1. **结构型（Structure Diagram）**

* 类图（Class Diagram）
* 对象图（Object Diagram）
* 构件图（Component Diagram）
* 部署图（Deployment Diagram）
* 包图（Package Diagram）

1. **行为型（Behavior Diagram）**
   * 活动图（Activity Diagram）
   * 状态机图（State Machine Diagram）
   * 顺序图（Sequence Diagram）
   * 通信图（Communication Diagram）
   * 用例图（Use Case Diagram）
   * 时序图（Timing Diagram）

## UML工具

* RationalRose
* Microsoft Office Visio
* PowerDesigner
* Star UML
* Argo UML

# 类图

## 类图的概念

类图以反映类的结构(属性、操作)以及类之间的关系为主要目的，描述了软件系统的结构，是一种静态建模方法

类图中的“类”与面向对象语言中的“类”的概念是对应的，是对现实世界中的事物的抽象

## 类图的组成

类图由三部分组成：类名、属性和操作（方法），其中类名是必须有的。

类如果有属性，则每一个属性都必须有一个名字，另外还可以有其它的描述信息，如可见性、数据类型、缺省值等

类如果有操作，则每一个操作也都有一个名字，其它可选的信息包括可见性、参数的名字、参数类型、参数缺省值和操作的返回值的类型等



# 对象图