# UML：统一建模语言

## 类图：

### 概述：面向对象建模的主要组成部分，展示了模型的静态结构

### 作用：表示类和类的关系，是设计阶段的产物

## 类图画法:

### 表示方法：

### + 表示public

### - 表示private

### # 表示protected

### 属性完整表示：可见性 名称 :类型 [ = 缺省值]

### 方法完整表示：可见性 名称(参数列表) [ :返回类型]

## 类与类之间关系的表示方法

### 关联关系：表示一类对象与另一类对象之间的联系。注意：关联关系和依赖关系的区别：依赖关系仅仅是变量和参数上依赖于另一个类，但是关联关系往往拥有另一个类类型的变量

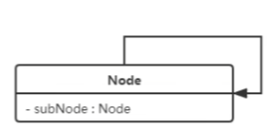
### 单项关联



### 双向关联



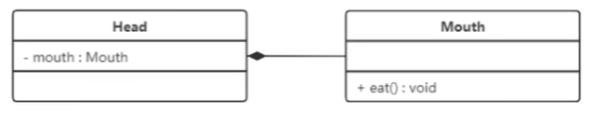
### 自关联



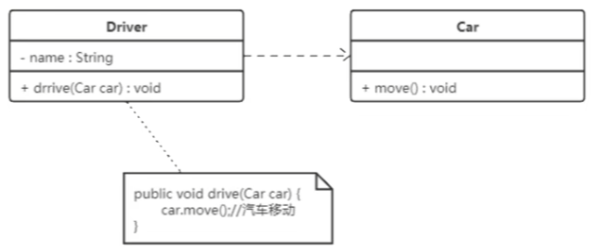
### 聚合关系:成员对象是整体对象的一部分，但成员对象可以脱离整体对象独立存在



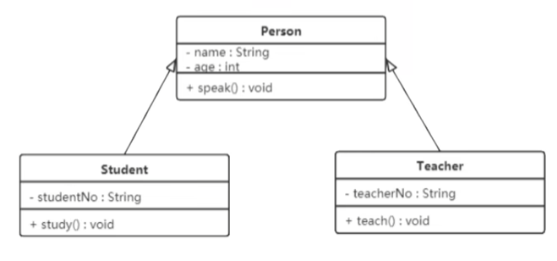
### 组合关系：一旦整体对象不存在，部分对象也将不存在，部分对象不能脱离整体对象而存在。



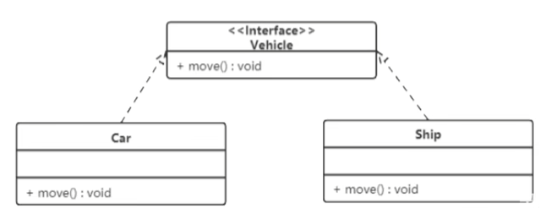
### 依赖关系： 临时性关联，某个类的方法通过局部变量，方法的参数，或者对静态方法的调用来访问另一个类



### 继承关系：父类与子类的关系，也就是泛化关系



### 实现关系：接口和实现类的关系



# 软件设计原则：维护性，复用性，可扩展性，灵活性

## 开闭原则：对扩展开放对修改关闭。在程序需要扩展的时候不能去修改原有代码。[见开闭原则](1%20开闭原则)

## 里氏代换原则 ：任何基类可以出现的地方子类一定可以出现。子类可以扩展父类的功能，但是不能改变父类原有的功能。例如正方形不属于长方形的案例[见里氏代换原则](2%20里氏替换原则)。因为正方形无法复用长方形所有功能。改进后的架构见[里氏代换原则改进](3%20里氏替换原则改进)

## 依赖倒转原则：高层模块不应该依赖于底层模块（A->B，A为高层模块），抽象类不应该依赖实现类，而实现类应该依赖于抽象类。详情请见[机箱示例](4%20依赖倒转原则)，改进后的架构见[机箱示例改进](5%20依赖倒转原则改进)

## 接口隔离原则：客户端不应该被迫依赖于他不使用的方法（例如B类继承于A类，但是只用A类的一部分方法），一个类对另一个类的依赖应该建立在最小的接口上（应该把A类拆开成两个类，B类单独继承于需要的那个类），例如[防盗门案例](6%20接口隔离原则)，[改进防盗门案例](7%20接口隔离原则改进)

## 迪米特原则：只和你的直接朋友交谈，不跟陌生人说话。如果两个软件尸体无需直接通信，那么就不应当发生直接的相互调用，可以通过第三方转发该调用。（例如要做软件的公司不会直接找软件工程师，而是去找软件公司，由软件公司去找软件工程师）

## 合成复用原则