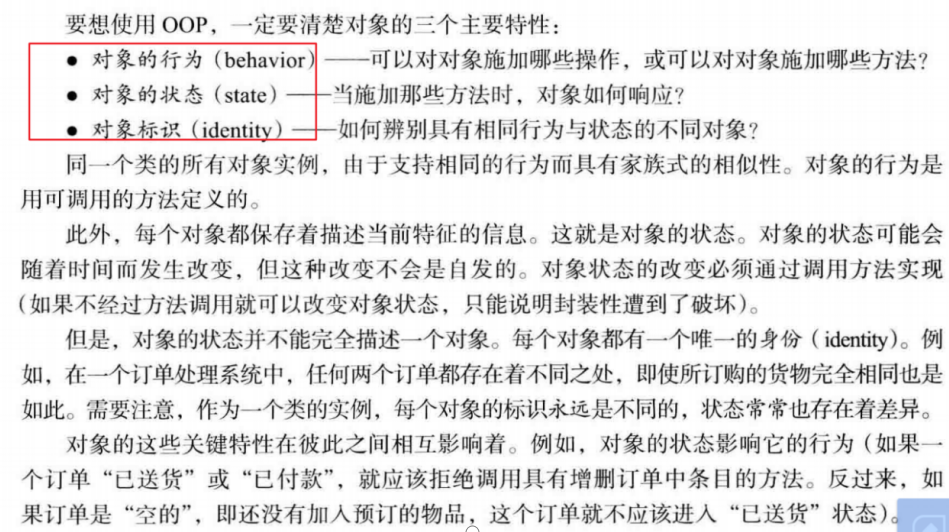
# [UML类图](https://www.cnblogs.com/huaxingtianxia/p/6590351.html)

## 1.类描述-OOP的三个特征



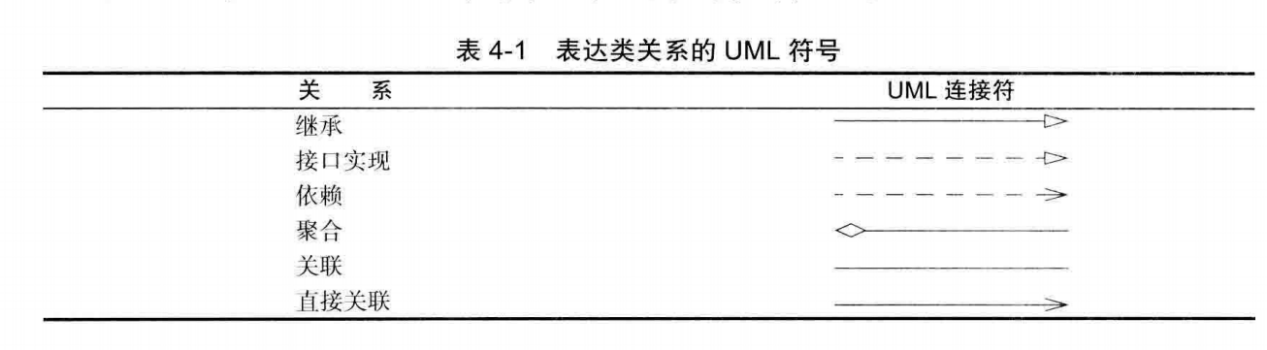
## 2.UML中类之间的关系

本节向大家学习一下UML箭头、线条代表的意义，UML中关系主要有依赖，聚合，合成，泛化和实现等，下面就让我们来看一下这些关系如何用UML箭头和线条来实现。

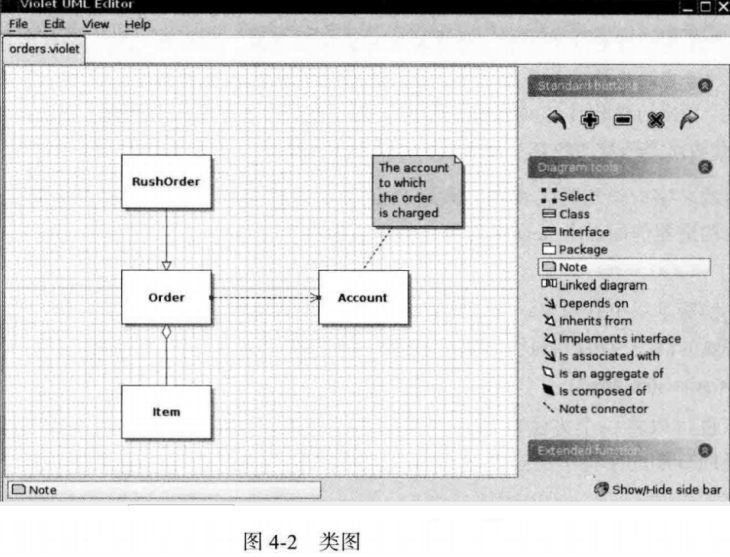
**关系**

后面的例子将针对某个具体目的来独立地展示各种关系。虽然语法无误，但这些例子可进一步精炼，在它们的有效范围内包括更多的语义。

### UML符号和含义示意图







Violet UML Editor

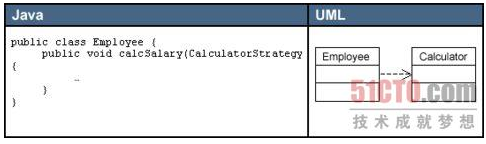
### ****依赖（Dependency）****

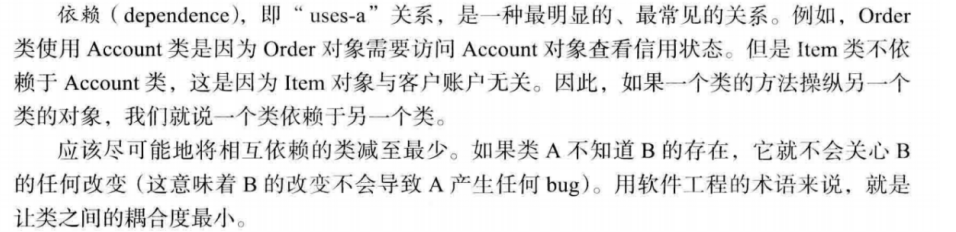
实体之间一个“使用”关系暗示一个实体的规范发生变化后，可能影响依赖于它的其他实例（图D）。

更具体地说，它可转换为对不在实例作用域内的一个类或对象的任何类型的引用。其中包括一个局部变量，对通过方法调用而获得的一个对象的引用（如下例所示），或者对一个类的静态方法的引用（同时不存在那个类的一个实例）。

也可利用“依赖”来表示包和包之间的关系。由于包中含有类，所以你可根据那些包中的各个类之间的关系，表示出包和包的关系。

图D





依赖（Dependency）     依赖指的是类之间的调用关系，在UML中用带虚线的箭头表示。如果类A访问类B的属性或者方法，  或者类A负责实例化类B，那么可以说类A依赖类B。和关联关系不同，无须在类A中定义类B类型的属性。

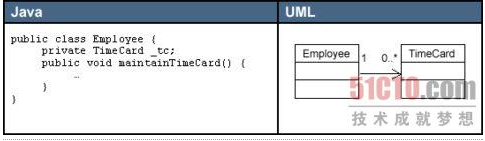
### ****关联（Association）****

实体之间的一个结构化关系表明对象是相互连接的。

UML箭头是可选的，它用于指定导航能力。如果没有箭头，暗示是一种双向的导航能力。

在[Java](http://lib.csdn.net/base/javase" \o "Java SE知识库" \t "https://www.cnblogs.com/huaxingtianxia/p/_blank)中，关联（图E）转换为一个实例作用域的变量，就像图E的“Java”区域所展示的代码那样。可为一个关联附加其他修饰符。多重性（Multiplicity）修饰符暗示着实例之间的关系。在示范代码中，Employee可以有0个或更多的TimeCard对象。但是，每个TimeCard只从属于单独一个Employee。

图E



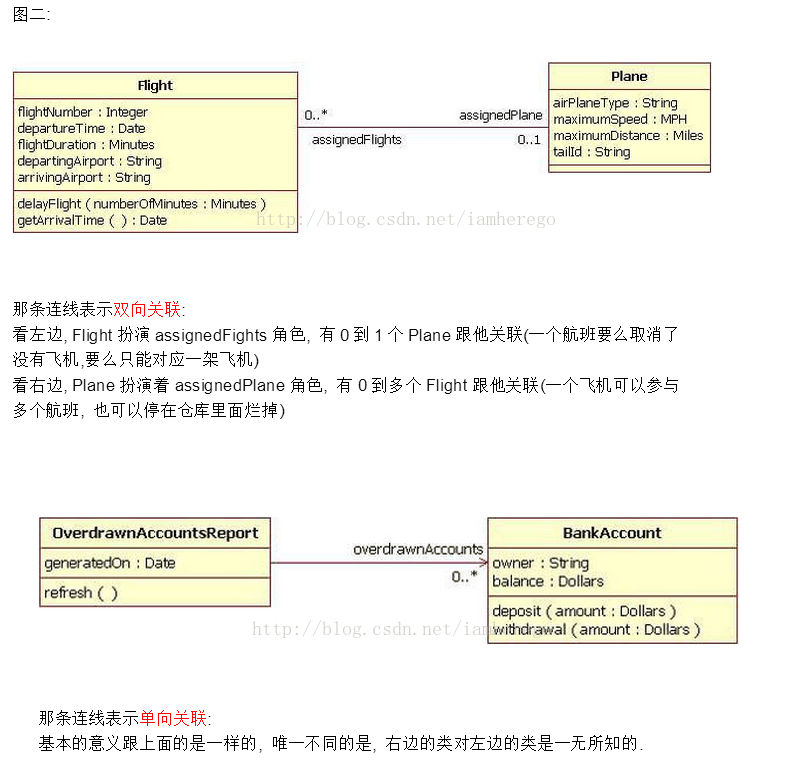
关联指的是类之间的特定对应关系，在UML中用带实线的箭头表示。按照类之间的数量对比，关联 可以分为以下三种：

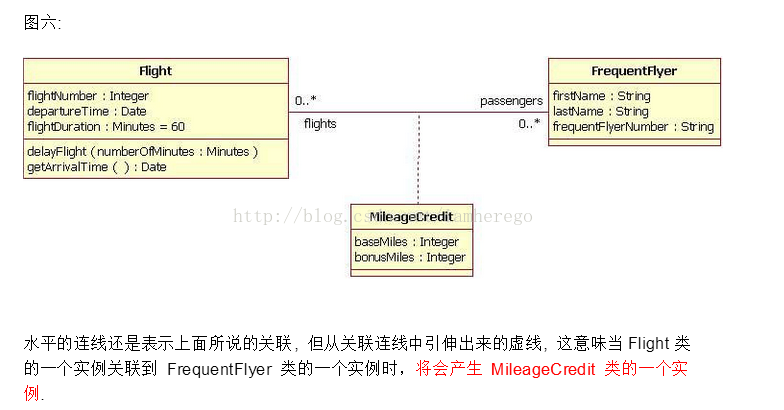
● 一对一关联

● 一对多关联

● 多对多关联

注意：关联还要以分为单向关联和双向关联







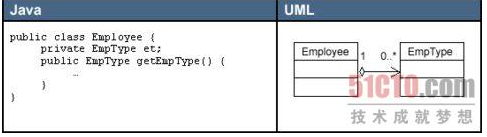
### ****聚合（Aggregation）****

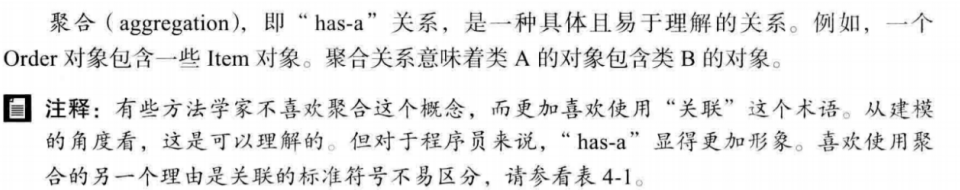
聚合（图F）是关联的一种形式，UML箭头代表两个类之间的整体/局部关系。

聚合暗示着整体在概念上处于比局部更高的一个级别，而关联暗示两个类在概念上位于相同的级别。

聚合也转换成Java中的一个实例作用域变量。  
关联和聚合的区别纯粹是概念上的，而且严格反映在语义上。聚合还暗示着实例图中不存在回路。换言之，只能是一种单向关系。

图F





聚集（Aggregation）     聚集指的是整体与部分之间的关系，在UML中用带实线的菱形箭头表示。 聚集关系还可以分为两种类型：

● 被聚集的子系统允许被拆卸和替换，这是普通聚集关系。     即聚合关系 (实线+空心菱形 表示)

● 被聚集的子系统不允许被拆卸和替换，这种聚集称为强聚集关系，或组成关系。即合成关系  (实线+实心菱形 表示)

注：强聚集（组成）可用带实线的实心菱形箭头表示。



### ****合成（Composition）****

组合（Composition）：用实心的菱形+实线箭头来表示  组合：部分和整体的关系，并且生命周期是相同的（“局部”会随着“整体”的创建而创建，销毁而销毁)。例如：人与手

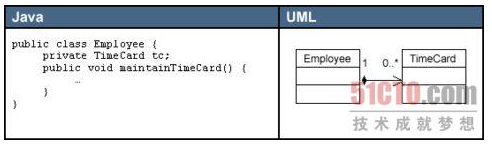
合成（图G）是聚合的一种特殊形式，UML箭头暗示“局部”在“整体”内部的生存期职责。

合成也是非共享的。所以，虽然局部不一定要随整体的销毁而被销毁，但整体要么负责保持局部的存活状态，要么负责将其销毁。

局部不可与其他整体共享。但是，整体可将所有权转交给另一个对象，后者随即将承担生存期职责。

Employee和TimeCard的关系或许更适合表示成“合成”，而不是表示成“关联”。

图G



聚集（Aggregation）     聚集指的是整体与部分之间的关系，在UML中用带实线的菱形箭头表示。 聚集关系还可以分为两种类型：

● 被聚集的子系统允许被拆卸和替换，这是普通聚集关系。     即聚合关系 (实线+空心菱形 表示)

● 被聚集的子系统不允许被拆卸和替换，这种聚集称为强聚集关系，或组成关系。即合成关系  (实线+实心菱形 表示)

注：强聚集（组成）可用带实线的实心菱形箭头表示。



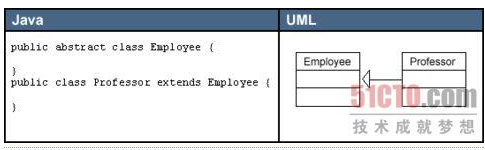
### ****泛化（Generalization）****

泛化（图H）表示**一个**更泛化的元素和一个更具体的元素之间的关系。

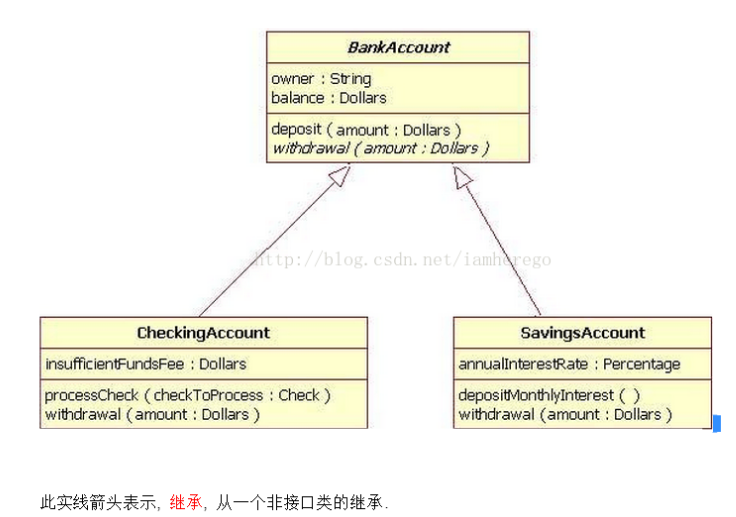
泛化是用于对继承进行建模的UML元素。

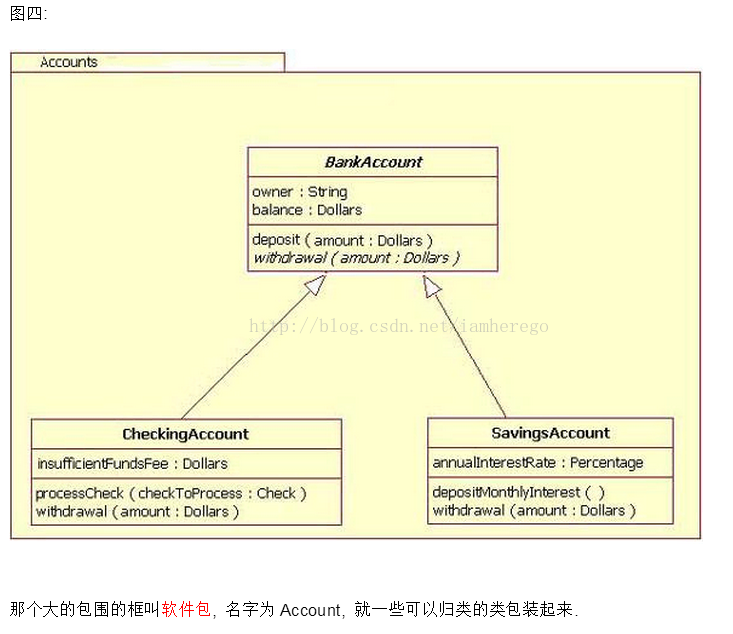
在Java中，用extends关键字来直接表示这种关系。

图H



泛化（Generalization）泛化指的是类之间的继承关系，在UML中用带实线的三角形箭头表示。

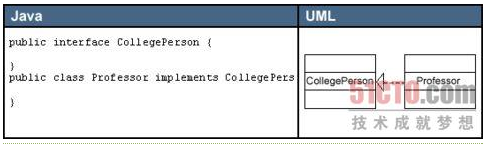




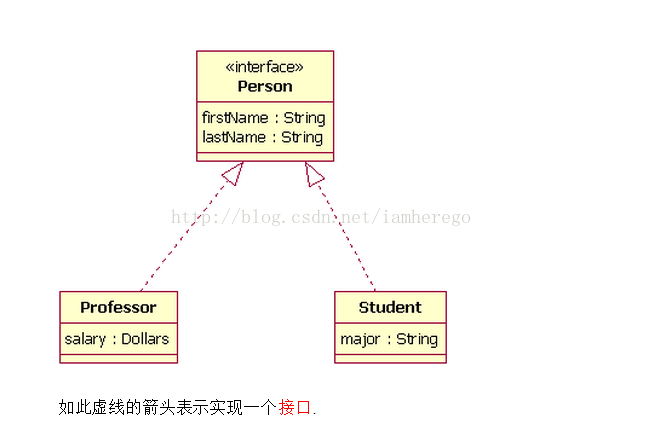
### ****实现（Realization）****

实例（图I）关系UML箭头指定两个实体之间的一个合同。换言之，一个实体定义一个合同，而另一个实体保证履行该合同。对Java应用程序进行建模时，实现关系可直接用implements关键字来表示。

图I



实现（Realization） 实现指的是类与接口之间的关系，在UML中用带虚线的三角形箭头表示。



## 3.UML 类图使用说明

### 3.1 UML类图各符号含义

1. 类（Class）：使用三层矩形框表示。   第一层显示类的名称，如果是抽象类，则就用斜体显示。   第二层是字段和属性。  第三层是类的方法。   注意前面的符号，‘+’表示public，‘-’表示private，‘#’表示protected。

2. 接口：使用两层矩形框表示，与类图的区别主要是顶端有<<interface>>显示 。   第一行是接口名称。  第二行是接口方法。

3. 继承类（extends） ：用空心三角形+实线来表示。

4. 实现接口（implements） ：用空心三角形+虚线来表示

5. 关联（Association） ：用实线箭头来表示，例如：A公司和B公司进行合作关系。

6. 聚合（Aggregation） ：用空心的菱形+实线箭头来表示   聚合：表示一种弱的‘拥有’关系，体现的是A对象可以包含B对象，但B对象不是A对象的一部分，例如： 公司和员工

7.组合（Composition）：用实心的菱形+实线箭头来表示  组合：部分和整体的关系，并且生命周期是相同的。例如：人与手

8. 依赖（Dependency） ：用虚线箭头来表示，例如：动物与氧气

9. 基数 ：连线两端的数字表明这一端的类可以有几个实例，比如：一个鸟应该有两只翅膀。如果一个类 可能有无数个实例，则就用‘n’来表示。关联、聚合、组合是有基数的。

类图基本符号可拆分为虚线，箭头，实线，空心右三角，实心右三角，空心菱形和实心菱形。由这些基本的图形进行组合构成了类图的基本符号。这里要注意这几个符号的顺序，代表了类与类之间关系的耦合程 度。越向右耦合度越高(依赖<继承<聚合)。

其中

虚线+箭头是表示即依赖的关系,

实线+箭头表示关联的关系，

虚线+空心右三角表示implements,

实线+空心右三角表示的是泛化，即类的继承关系。

实线+空心菱形表示的是聚合的关系，

实线+实心菱形则表示 组合的关系。

### 3.2类图的时候要注意

类图的思想其实也还没有脱离面向对象的思想，以某个类为中心，有些线是射入的而有些线是射出的。

射入的线表示的是这个类被哪些类所调用, 而射出的线则表示该类调用了 哪些类，包括泛化，关联，依赖，聚合和组合四种关系。这类似于离散数学中有关图部分的描述。

### 3.3类之间的关系

UML把类之间的关系分为以下5种.

● 关联：类A与类B的实例之间存在特定的对应关系

● 依赖：类A访问类B提供的服务

● 聚集：类A为整体类，类B为局部类，类A的对象由类B的对象组合而成

● 泛化：类A继承类B

● 实现：类A实现了B接口   关联（Association）

关联指的是类之间的特定对应关系，在UML中用带实线的箭头表示。按照类之间的数量对比，关联  可以分为以下三种：

● 一对一关联

● 一对多关联

● 多对多关联

注意：关联还要以分为单向关联和双向关联

依赖（Dependency）     依赖指的是类之间的调用关系，在UML中用带虚线的箭头表示。如果类A访问类B的属性或者方法，  或者类A负责实例化类B，那么可以说类A依赖类B。和关联关系不同，无须在类A中定义类B类型的属性。

聚集（Aggregation）     聚集指的是整体与部分之间的关系，在UML中用带实线的菱形箭头表示。 聚集关系还可以分为两种类型：

● 被聚集的子系统允许被拆卸和替换，这是普通聚集关系。     即聚合关系 (实线+空心菱形 表示)

● 被聚集的子系统不允许被拆卸和替换，这种聚集称为强聚集关系，或组成关系。即合成关系  (实线+实心菱形 表示)

注：强聚集（组成）可用带实线的实心菱形箭头表示。

实现（Realization） 实现指的是类与接口之间的关系，在UML中用带虚线的三角形箭头表示。