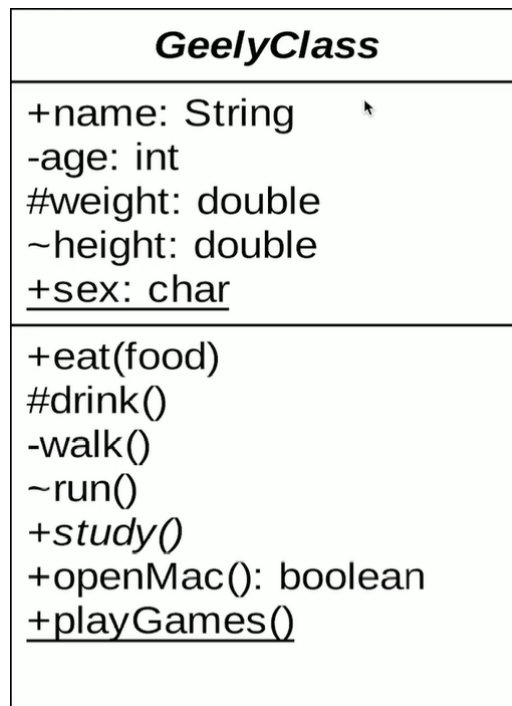


## 1.类图基础属性

---



- -表示private
- #表示protected
- ~表示default,也就是包权限
- \_下划线表示static
- 斜体表示抽象

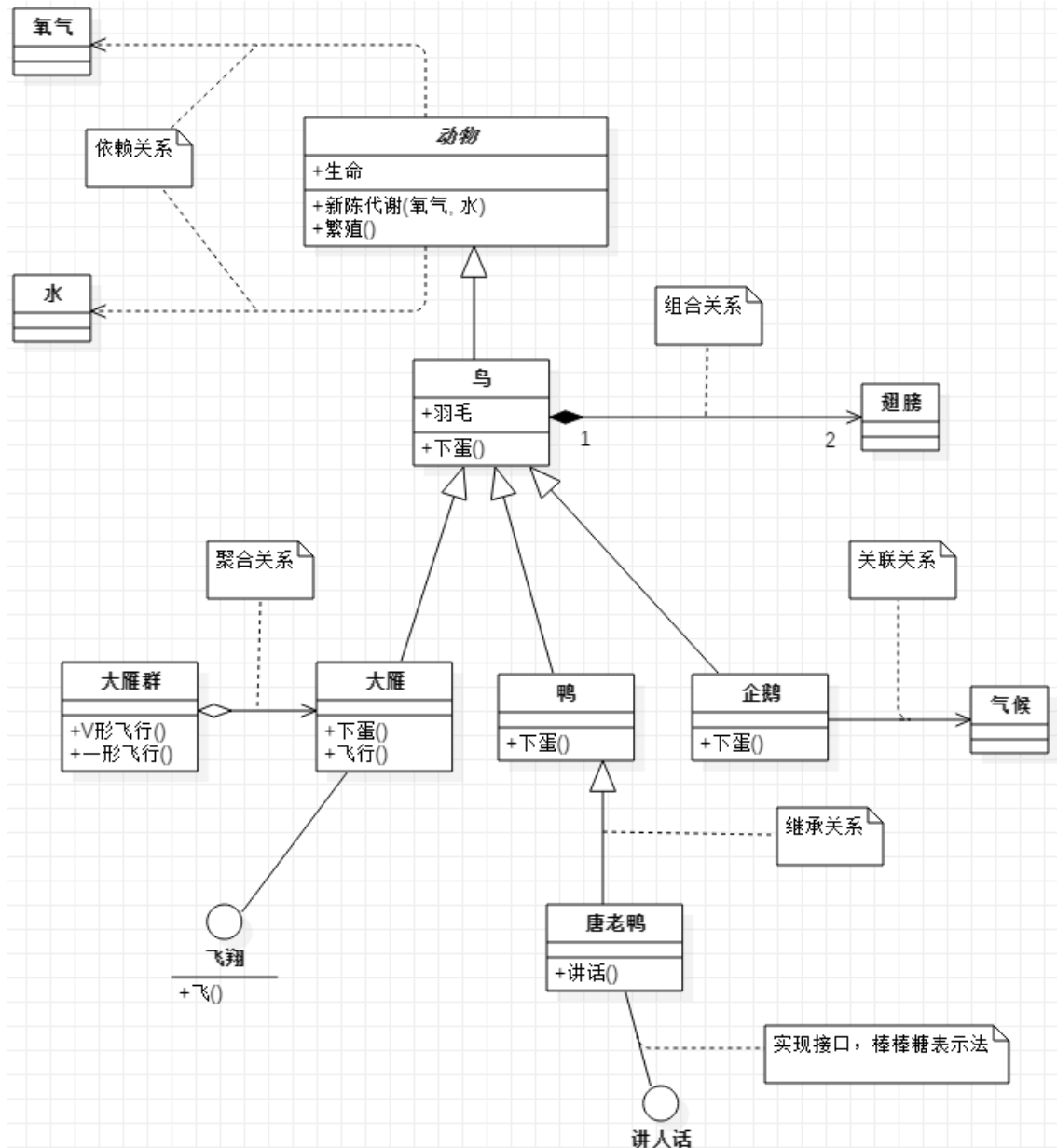
## 2.类与类之间关系

---

在UML类图中，常见的有以下几种关系：

- 泛化 (Generalization)
- 实现 (Realization)
- 关联 (Association)
- 聚合 (Aggregation)
- 组合(Composition)

- 依赖(Dependency)



## 2.1泛化

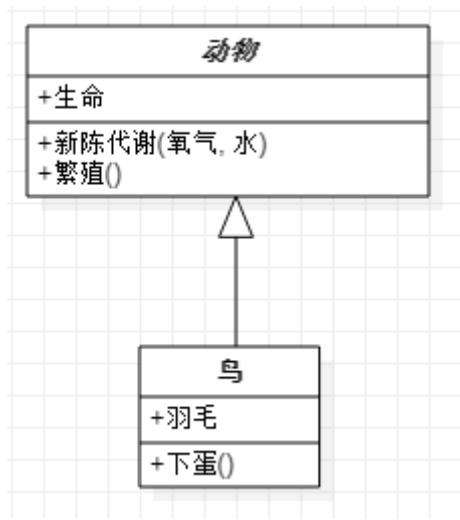
介绍：泛化(Generalization)表示类与类之间的继承关系，接口与接口之间的继承关系，或类对接口实现关系

### (1) 继承

介绍：继承表示是一个类（称为子类、子接口）继承另外的一个类（称为父类、父接口）的功能，并可以增加它自己的新功能的能力。

表示方法：继承使用空心三角形+实线表示。

示例：鸟类继承抽象类动物



## (2) 实现

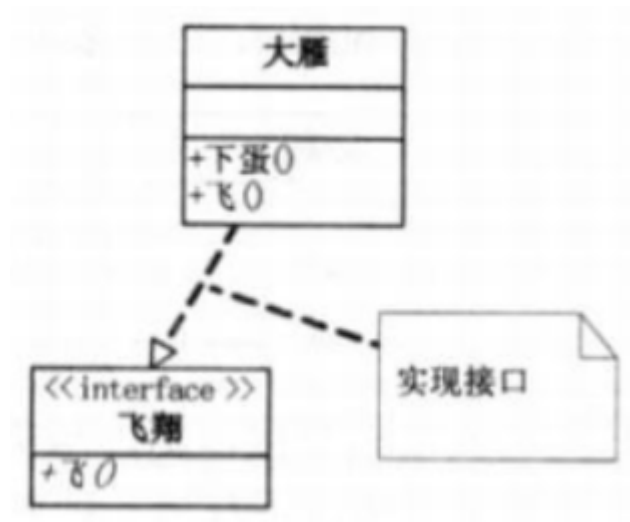
介绍：实现表示一个class类实现interface接口（可以是多个）的功能。

表示方法：

1) 矩形表示法

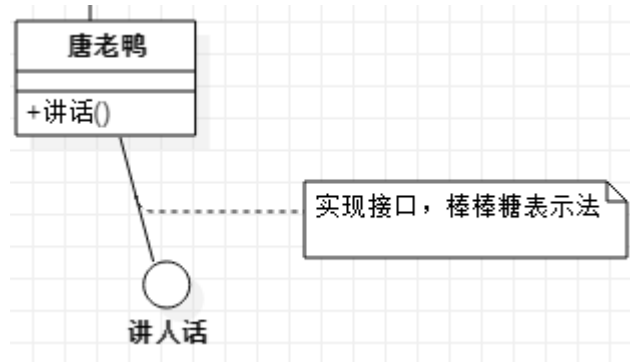
使用空心三角形+虚线表示

比如：大雁需要飞行，就要实现飞()接口



2) 棒棒糖表示法

使用实线表示

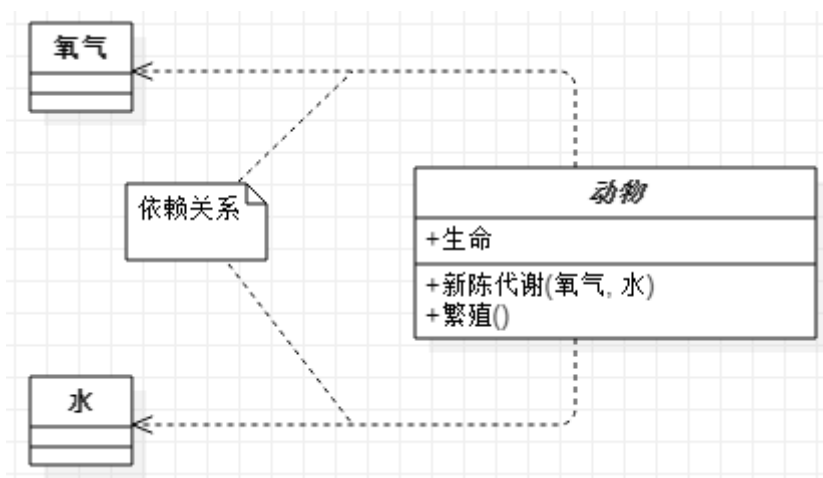


## 2.2依赖

介绍：对于两个相对独立的对象，当一个对象负责构造另一个对象的实例，或者依赖另一个对象的服务时，这两个对象之间主要体现为依赖关系。

表示方法：依赖关系用虚线箭头表示。

示例：动物依赖氧气和水。调用新陈代谢方法需要氧气类与水类的实例作为参数

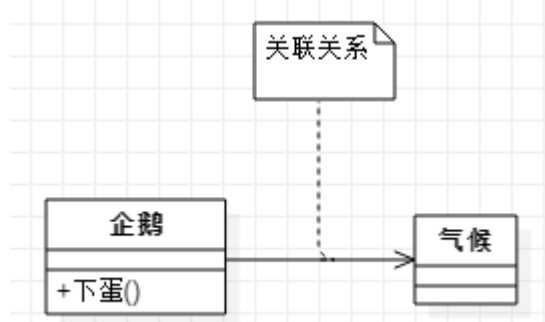


## 2.3关联

介绍：对于两个相对独立的对象，当一个对象的实例与另一个对象的一些特定实例存在固定的对应关系时，这两个对象之间为关联关系。

表示方法：关联关系用实线箭头表示。

示例：企鹅需要‘知道’气候的变化，需要‘了解’气候规律。当一个类‘知道’另一个类时，可以用关联。

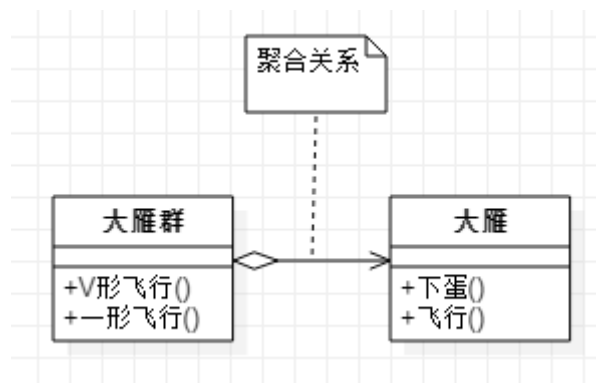


## 2.4聚合

介绍：表示一种弱的‘拥有’关系，即has-a的关系，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。两个对象具有各自的生命周期。

表示方法：聚合关系用空心的菱形+实线箭头表示。

示例：每一只大雁都属于一个大雁群，一个大雁群可以有多只大雁。当大雁死去后大雁群并不会消失，两个对象生命周期不同。



## 2.5组合

介绍：组合是一种强的‘拥有’关系，是一种contains-a的关系，体现了严格的部分和整体关系，部分和整体的生命周期一样。

表示方法：组合关系用实心的菱形+实线箭头表示，还可以使用连线两端的数字表示某一端有几个实例。

示例：鸟和翅膀就是组合关系，因为它们是部分和整体的关系，并且翅膀和鸟的生命周期是相同的。

