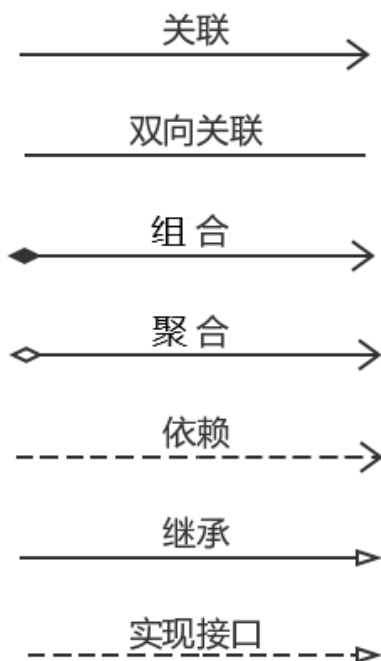


UML类图中的基本图示法



UML 图

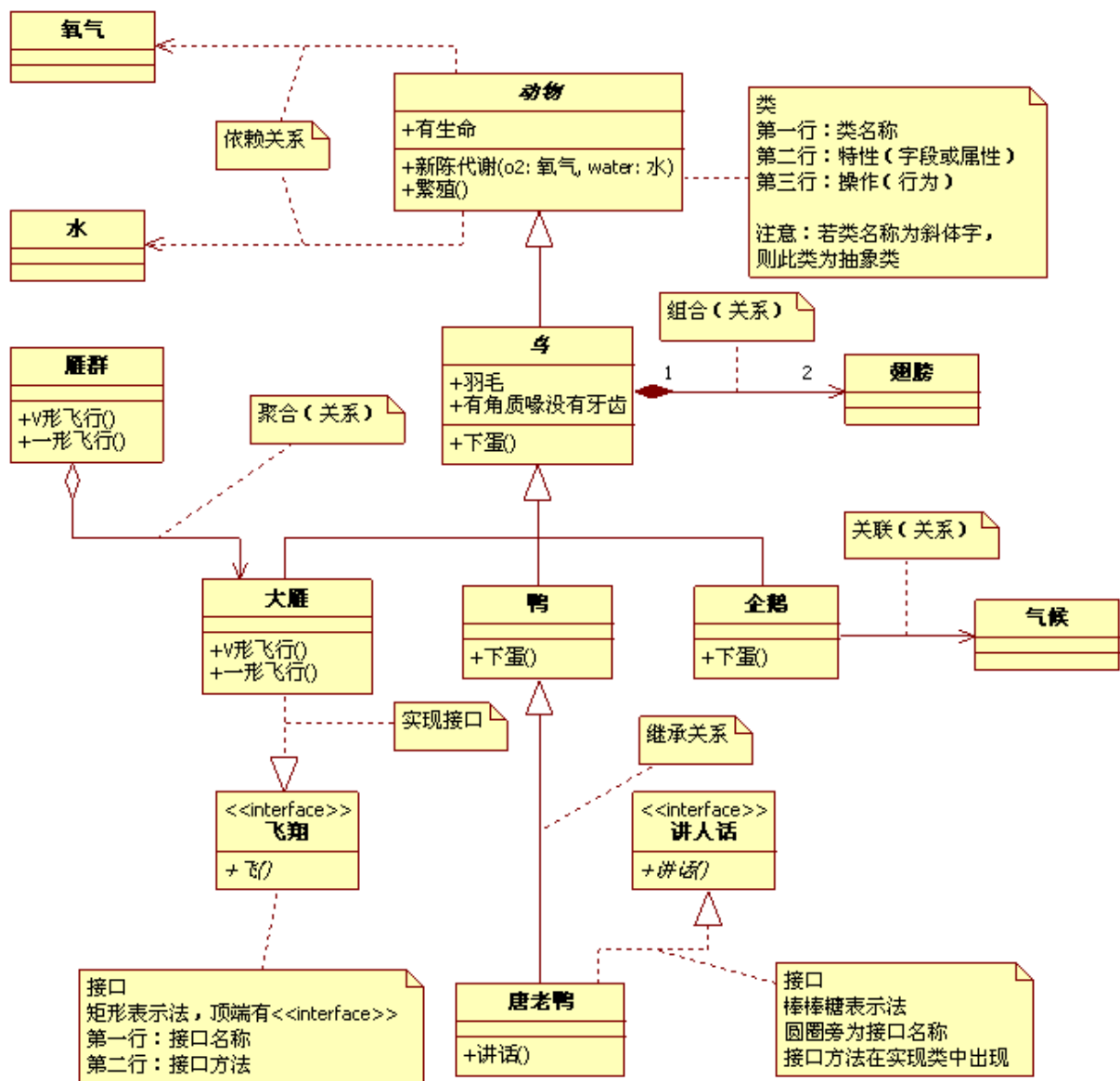
画UML图与写文章差不多，都是把自己的思想描述给别人看，关键在于思路和条理，UML图分类：

用例图 (use case)

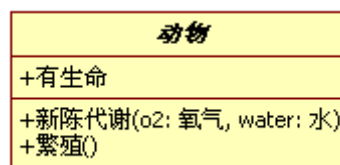
- 静态结构图：类图、对象图、包图、组件图、部署图
- 动态行为图：交互图（时序图与协作图）、状态图、活动图
- 类图是描述类与类之间的关系的，是UML图最核心的

UML 类图

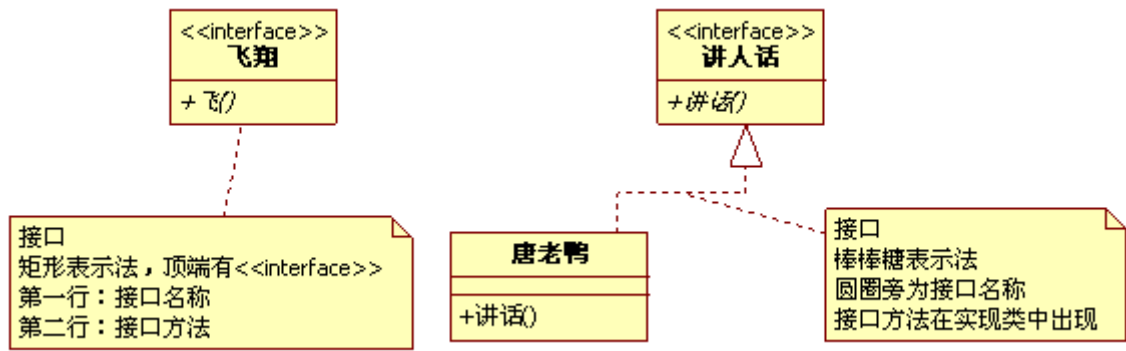
- 用于描述系统中的类(对象)本身的组成和类(对象)之间的各种静态关系。
- 类之间的关系：依赖、泛化（继承）、实现、关联、聚合与组合



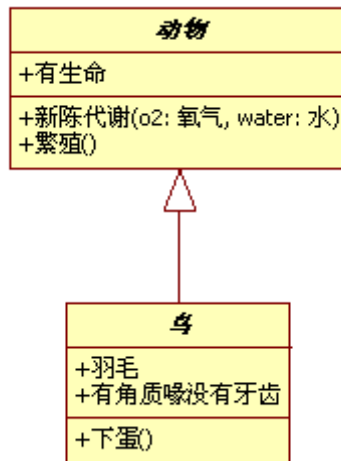
首先，看动物矩形框，它代表一个类（Class）。类图分三层，第一层显示类的名称，如果是抽象类，则就用斜体显示。第二层是类的特性，通常就是字段和属性。第三层是类的操作，通常是方法或行为。前面的符号，+ 表示public，- 表示private，# 表示protected。



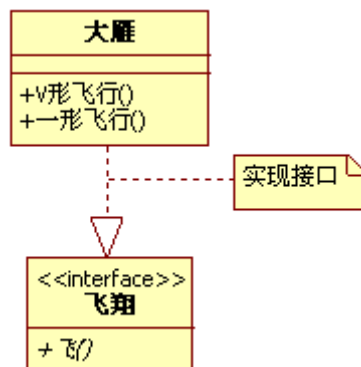
左下角的飞翔，它表示一个接口图，与类图的区别主要是顶端的<<interface>>显示。第一行是接口名称，第二行是接口方法。接口还有另一种表示方法，俗称棒棒糖表示法。唐老鸭是能讲人话的鸭子，实现了讲人话的接口。



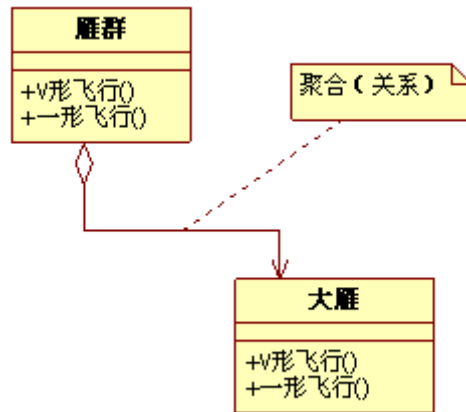
接下来的是类与类，类与接口之间的关系了。首先注意动物、鸟、鸭、唐老鸭之间的关系符号。它们都是**继承**的关系，继承关系用空心三角形 + 实线来表示的。



上述所列举的几种鸟中，大雁最能飞，让它**实现**了飞翔接口。实现接口用空心三角形 + 虚线来表示。

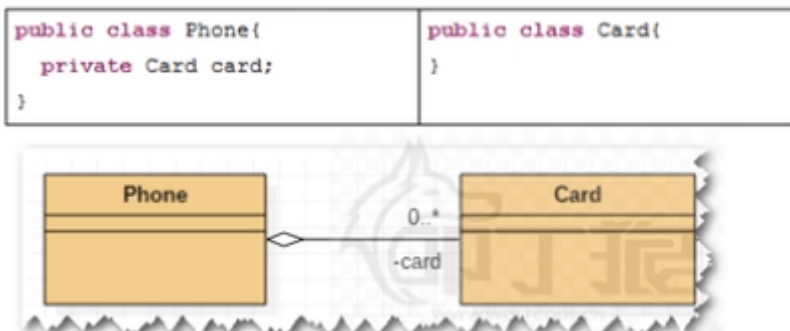


再来看大雁和雁群这两个类，大雁是群居动物，每只大雁都是属于一个雁群，一个雁群可以有多只大雁。所以它们之间就满足聚合（Aggregation）关系。聚合表示一种弱的拥有关系，体现的是A对象可以包含B对象，但B对象不是A对象的一部分。聚合关系用空心的菱形 + 实线箭头来表示。



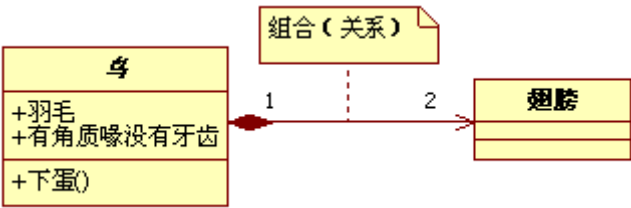
- 聚合关系是 **关联关系的一种特例**，他体现的是 **整体与部分**，是一种 **“弱拥有”** 的关系，即 **has-a** 的关系。聚合是 **整体** 和 **个体** 之间的关系。
- 例如，汽车类与引擎类、轮胎类，以及其它的零件类之间的关系便整体和个体的关系。
- 与关联关系一样，**聚合关系** 也是通过 **实例变量** 实现的。但是关联关系所涉及的两个类是处在同一层次上的，而在聚合关系中，两个类是处在不等层次上的，一个代表整体，另一个代表部分。
- 聚合关系表示整体和个体的关系，整体和个体可以相互独立存在，一定是有两个模块分别管理整体和个体。
- 在UML类图中，聚合通常使用 **空心菱形+实线箭头** 表示

在UML中，聚合通常是使用**空心菱形+实线箭头**来表示。



组合（Composition）是一种强的拥有关系，体现了严格的部分和整体的关系，部分和整体的生命周期一样。在这里鸟和其翅膀就是组合关系，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。组合关系用实心的菱形 + 实线箭头来表示。另外，你会注意到合成关系的连线两端还有一个数字1和数字2，这被称为基数。表明这一端的类可以

有几个实例，很显然，一个鸟应该有两只翅膀。如果一个类可能有无数个实例，则就是n来表示。关联关系、聚合关系也可以有基数的。

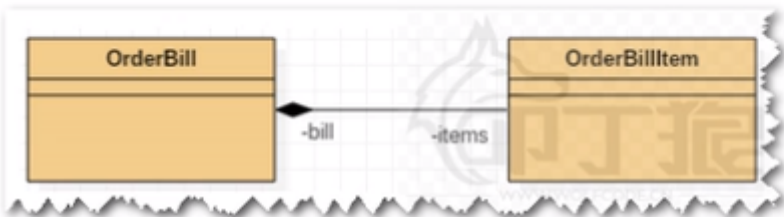


- 组合关系是 **关联关系的一种特例**，他体现的是一种 **contains-a** (包含)的关系，这种关系比聚合更强，也称为 **强聚合**。
- 它要求普通的聚合关系中代表整体的对象负责代表部分对象的生命周期，组合关系是不能共享的。代表整体的对象需要负责保持部分对象和存活，在一些情况下将负责代表部分的对象湮灭掉。代表整体的对象可以将代表部分的对象传递给另一个对象，由后者负责此对象的生命周期。换言之，代表部分的对象在每一个时刻只能与一个对象发生组合关系，由后者排他地负责生命周期。部分和整体的生命周期一样。
- **整体和个体不能独立存在，一定是在一个模块中同时管理整体和个体，生命周期必须相同(级联)。**
- 在UML类图中，组合通常使用 **实心菱形+实线箭头** 表示

在UML中，组合通常是使用实心菱形+实线箭头表示。

```
public class OrderBill {
    private List<OrderBillItem> items;
}

public class OrderBillItem {
    private OrderBill bill;
}
```



引用

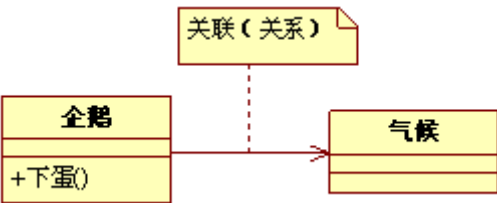
id	number	bill_id
10	100	1
20	100	2

id	sn
1	10001
2	10002

https://blog.csdn.net/m0_37989980

企鹅和气候两个类，企鹅是很特别的鸟，会游不会飞。更重要的是，它与气候有很大的关联。我们不去讨论为什么北极没有企鹅，为什么它们要每年长途跋涉。总之，企鹅需要知道

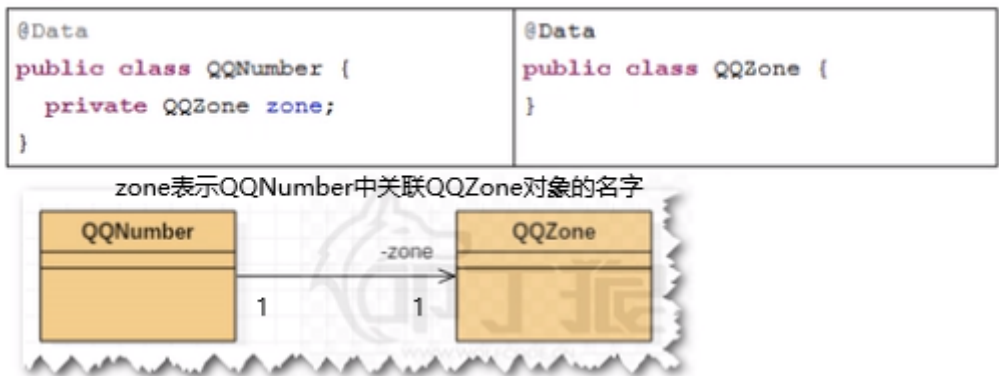
气候的变化，需要了解气候规律。当一个类知道另一个类时，可以用**关联**（association）。关联关系用实线箭头来表示。



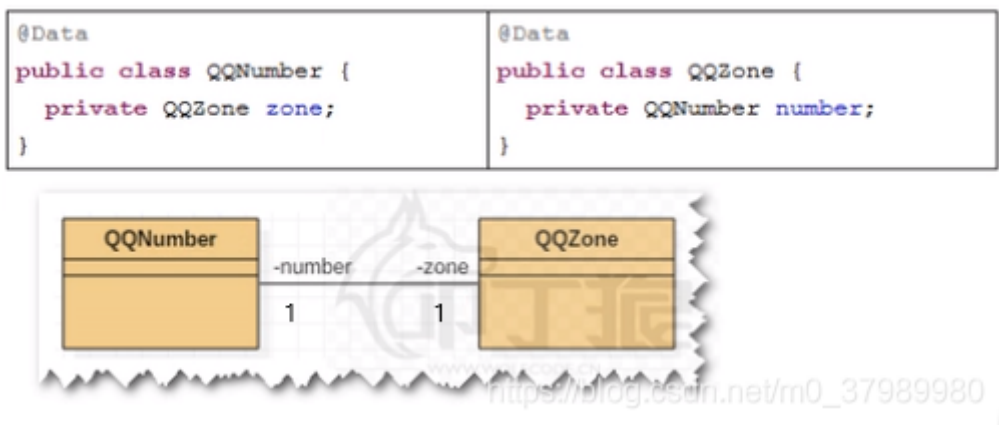
关联关系：指的是**类与类之间的联接**，它使一个类知道另一个类的属性和方法（实例变量体现）。A类依赖于B对象，并且把B作为A的一个成员变量，则A和B存在关联关系。关联可以是双向的，也可以是单向的。两个类之间是一个层次的，不存在部分跟整体之间的关系。

一对一：一个A对象属于一个B对象，一个B对象属于一个A对象。

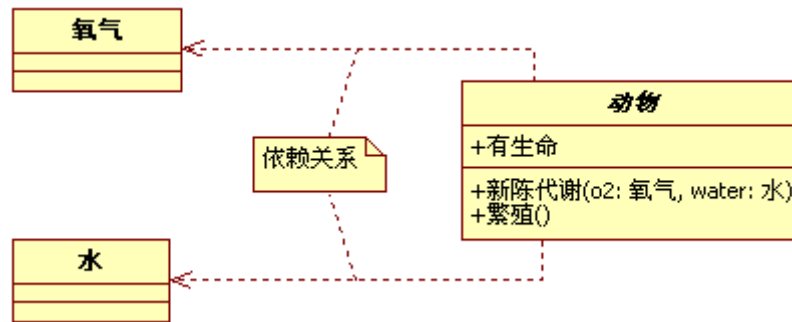
单向: Java类设计



双向: Java类设计



动物的几大特征，比如有新陈代谢，能繁殖。而动物要有生命力，需要氧气、水以及食物等。也就是说，动物依赖于氧气和水。他们之间是**依赖**关系（Dependency），用虚线箭头来表示。



依赖关系：指的是类与类之间的联接。依赖关系表示一个类依赖于另一个类的定义。一般而言，依赖关系在Java语言中体现为成员变量、局域变量、方法的形参、方法返回值，或者对静态方法的调用。

GeelyClass
+name: String -age: int #weight: double ~height: double +sex: char
+eat(food) #drink() -walk() ~run() +study() +openMac(): boolean +playGames()

-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