

UML学习与总结

UML学习与总结

1.类、接口

类分三层：

接口类分两层：

2.继承关系

3.关联关系

4.聚合关系

5.合成『组合』

6.依赖关系

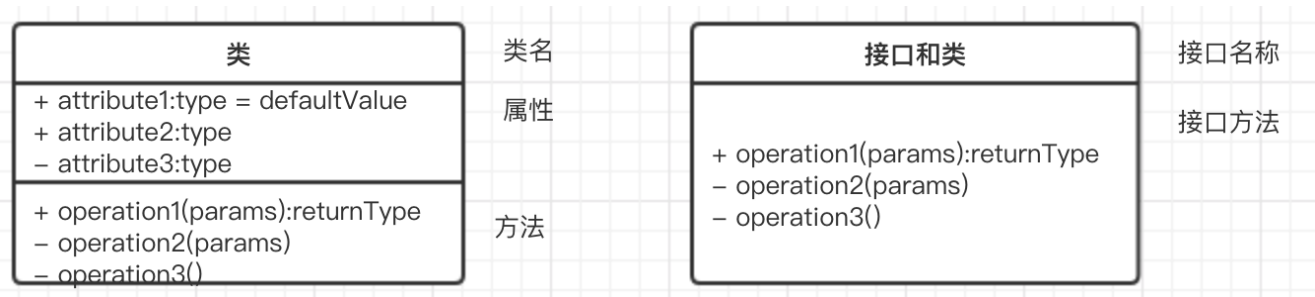
1.类、接口

类分三层：

- 第一层表示类 **名称**，如果是 **抽象类** 则用 **斜体表示**。 **+** 表示public, **-** 表示private, **#** 表示protected
- 第二层表示类 **属性**
- 第三层表示类 **方法**

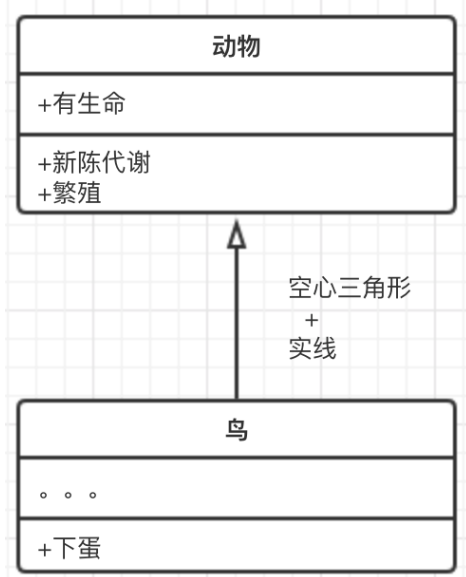
接口类分两层：

- 第一层表示 **接口名**（ **C++中为类名** ）
- 第二层表示 **接口方法**（ **C++中为虚函数名** ）



2.继承关系

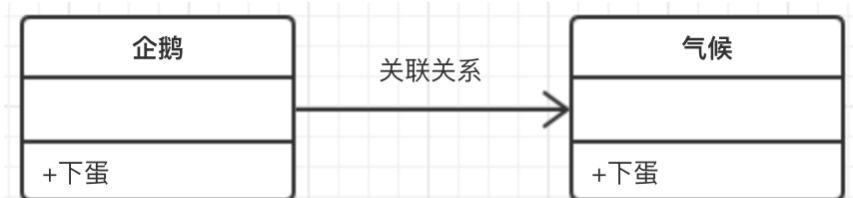
- 继承关系用 **空心三角形** + **实线** 来表示



```
1 class Bird : public Animal
2 {
3     //鸟继承动物类
4 }
```

3. 关联关系

- 关联关系（即一个类对象作为另一个类对象的成员变量）用 空线箭头 来表示

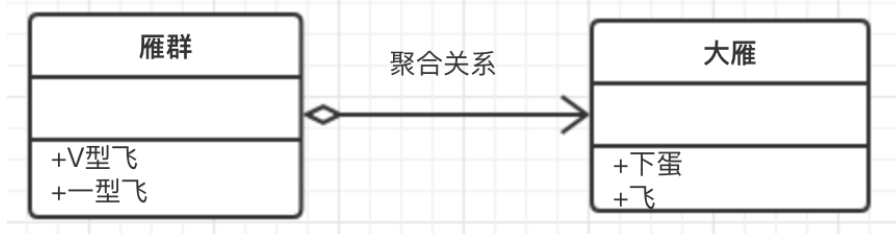


企鹅需要‘知道’气候的变化，需要‘了解’气候规律，当一个类‘知道’另一个类时，可以使用关联

```
1 class Penguin: public Bird
2 {
3     private:
4         Climate m_climate;    //在企鹅Penguin中，引用到气候Climate对象
5 }
```

4. 聚合关系

- 聚合关系表示一个弱的‘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。用 空心菱形 + 实线箭头 来表示



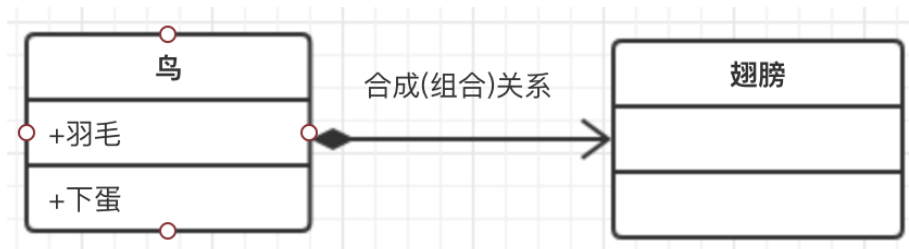
每只大雁都属于一个雁群，一个雁群可以有多个大雁。

```

1 class WideGooseAggregate
2 {
3     public:
4         list<WideGoose> m_listWideGoose; //在雁群WideGooseAggregate类中，有大雁对象的集合
5 }
  
```

5.合成『组合』

- 合成『组合』是一种强‘拥有’关系，体现了严格的部分与整体之间的关系，部分和整体的生命周期是一样的。用 **实心菱形 + 实线箭头** 来表示



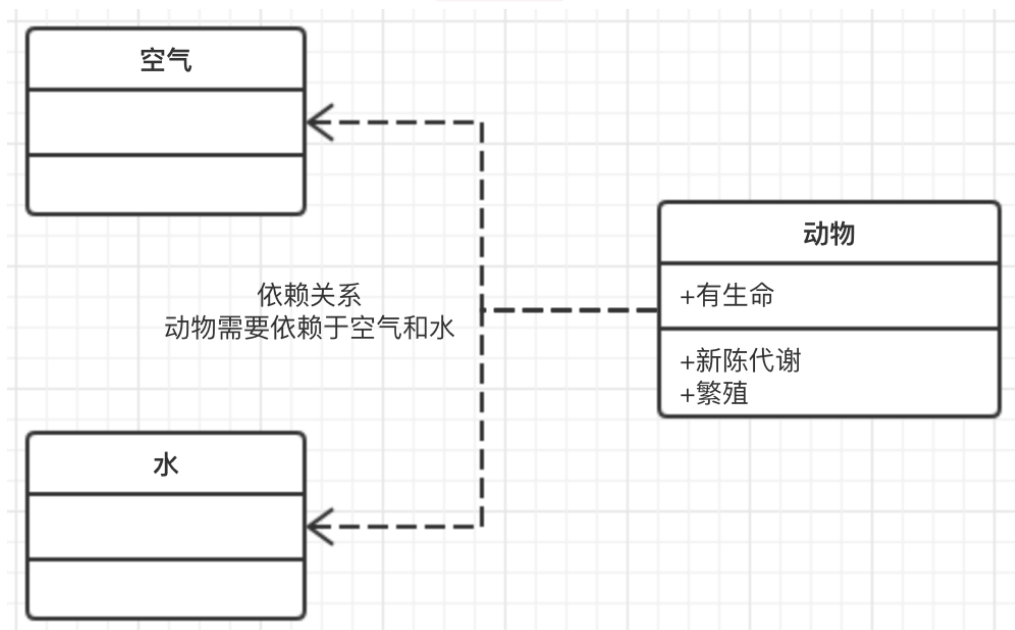
鸟和翅膀就是合成『组合』关系，因为他们是整体和部分的关系，并且翅膀和鸟的生命周期是相同的

```

1 class Bird
2 {
3     private:
4         Wing    m_wind; //在鸟类Bird中，初始化时即实例化了一个Wing翅膀
5 }
  
```

6.依赖关系

- 一个类中需要依靠其他的类。用 **虚线箭头** 来表示



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，他们之间就是依赖关系。

```
1 class Animal
2 {
3     Animal(Oxygen oxygen, Water water);
4 }
```