

# Ch. 7 类与类之间的关系

2018 / 12

# 面向对象

- 封装
- 继承
- 多态

# 为什么要多态？

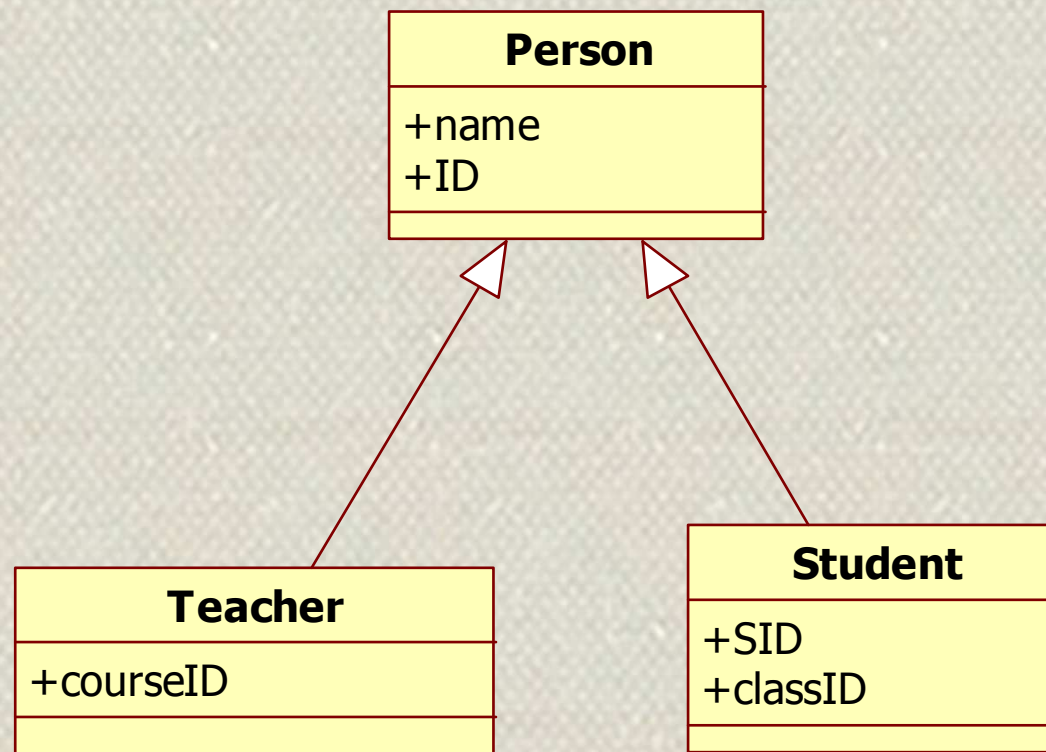
- 抽象类:Animal:人/青蛙/大象
- 接口:Movable:汽车/青蛙/棋子



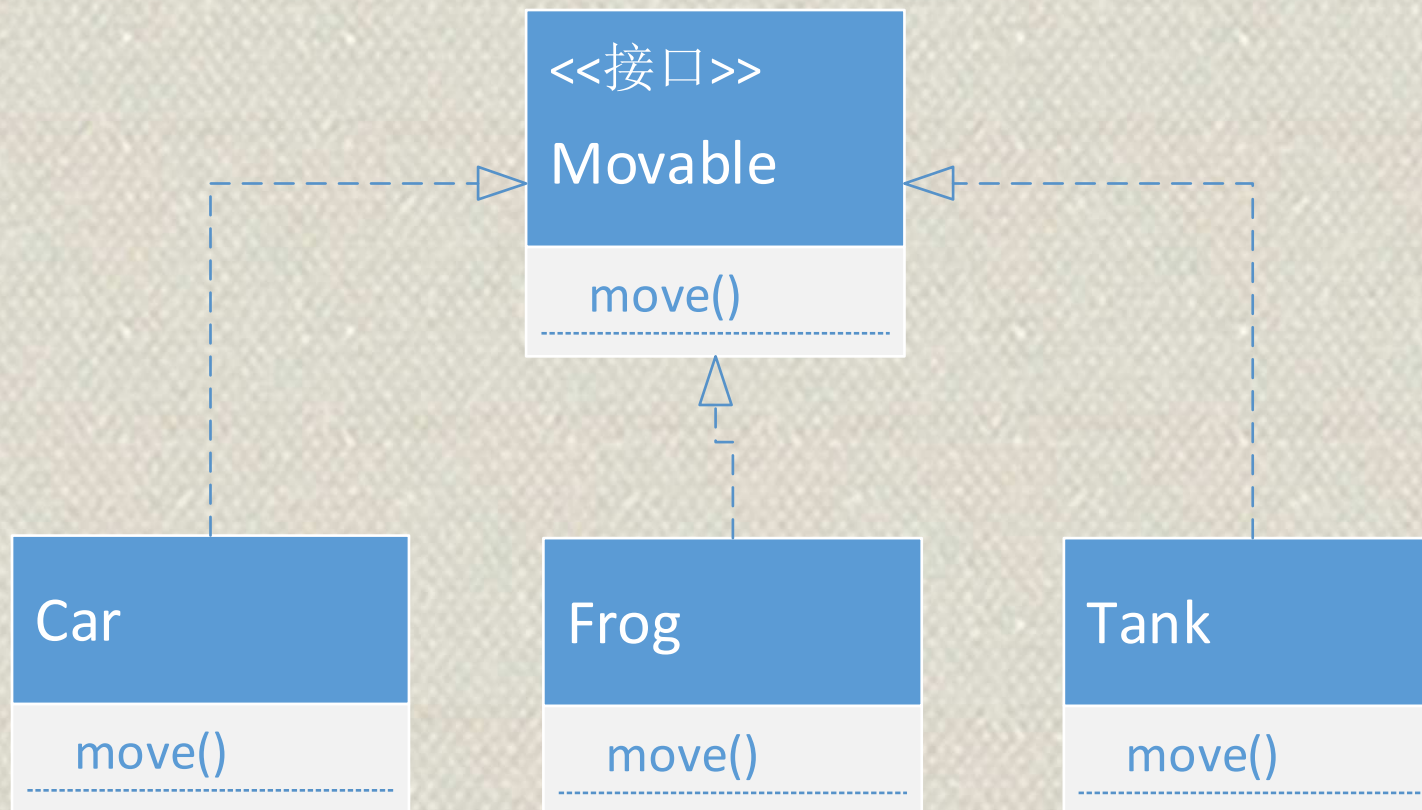
# UML类图

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

# 泛化



# 实现





# Is-a的概念

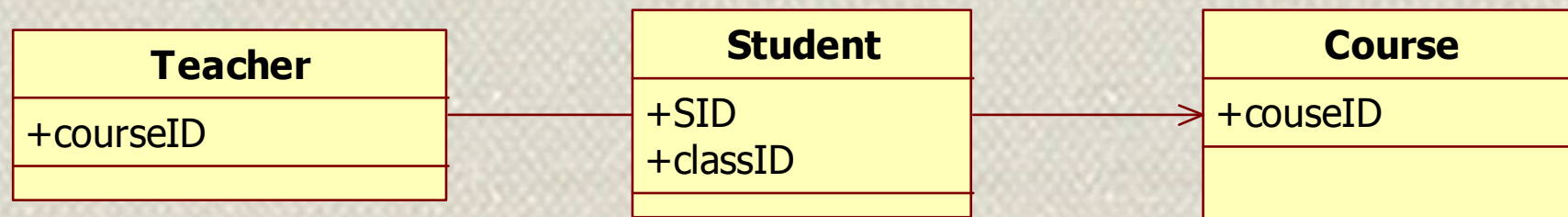
- Penguin is a waterfowl
- Parrot is a flyingBird
- Waterfowl is a bird
- flyingBird is a bird
- Bird is an animal

# 泛化和实现的区别

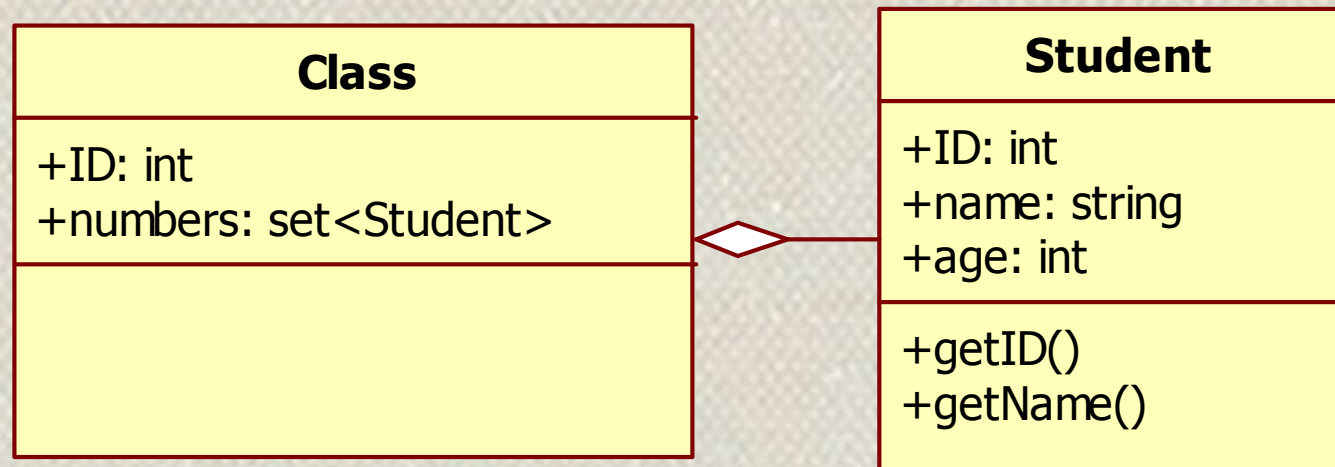
- 类与接口的区别
- 什么是接口interface，C++里没有接口的概念，和抽象类差不多
  - 接口中只能包含函数（方法）
  - 接口中的方法不能有任何实现
  - 接口不能被实例化



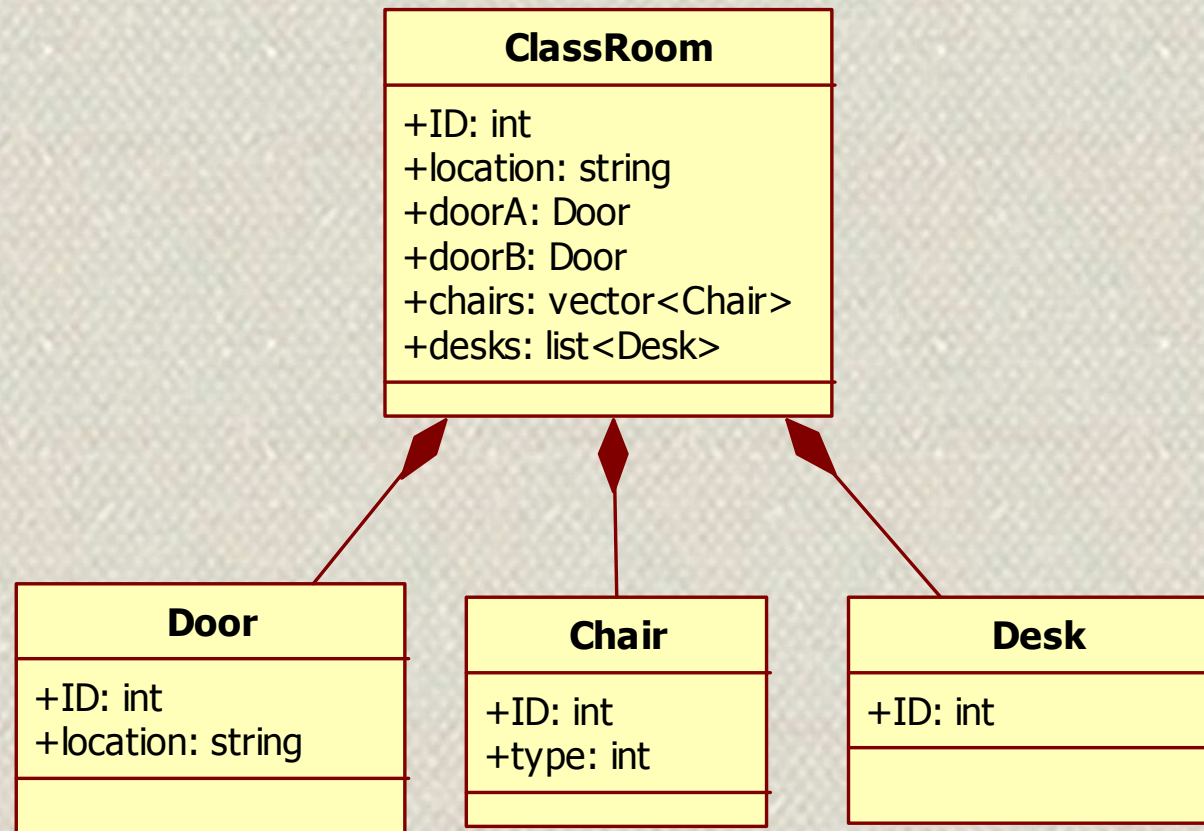
# 关联



# 聚合



# 组合





# Has-a的概念

- Bird has a beak
- Bird has a pair of wings

# 代码表达

```
class Classroom
{
    int roomID;
    Door front, back;
    BlackBoard mainboard
    list<Desk> desks;
};
```

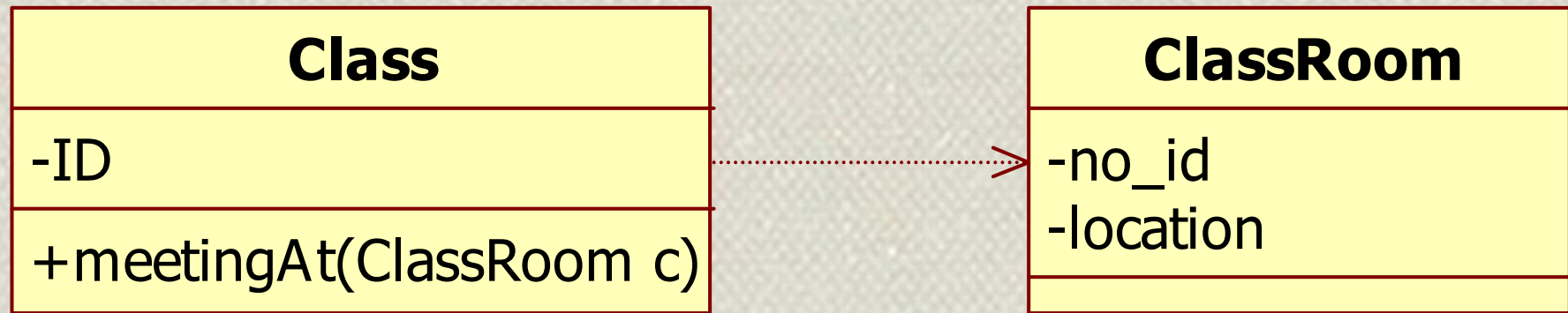


# 聚合与组合的区别

- 聚合，是成员构成集合
- 组合，是部件组成整体



# 依赖



# 依赖和关联的区别

- 依赖，有使用关系
- 关联，无使用关系

# Use-a的概念

- We use a book to read
- Students use a classroom to stay
- We don't use a teacher to instruct



