

面向对象方法与C++程序设计

第4章

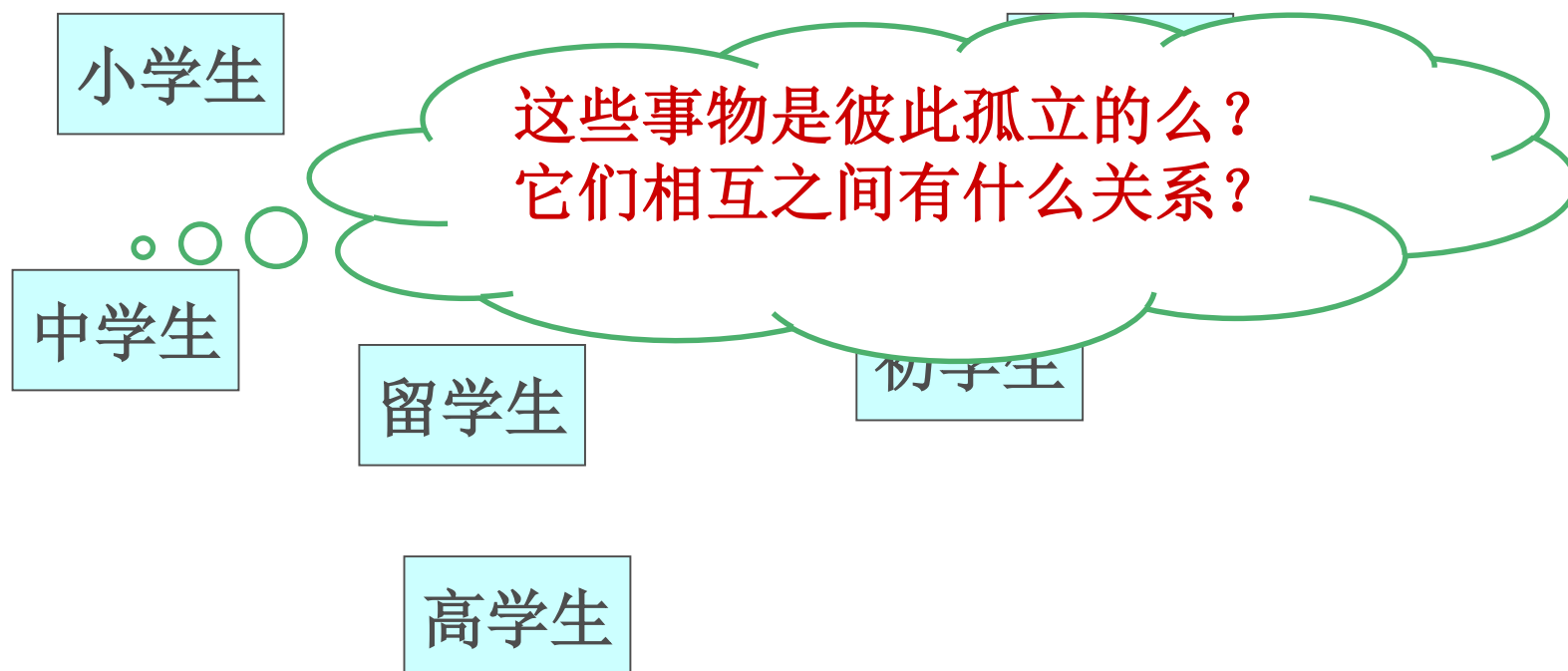
继承

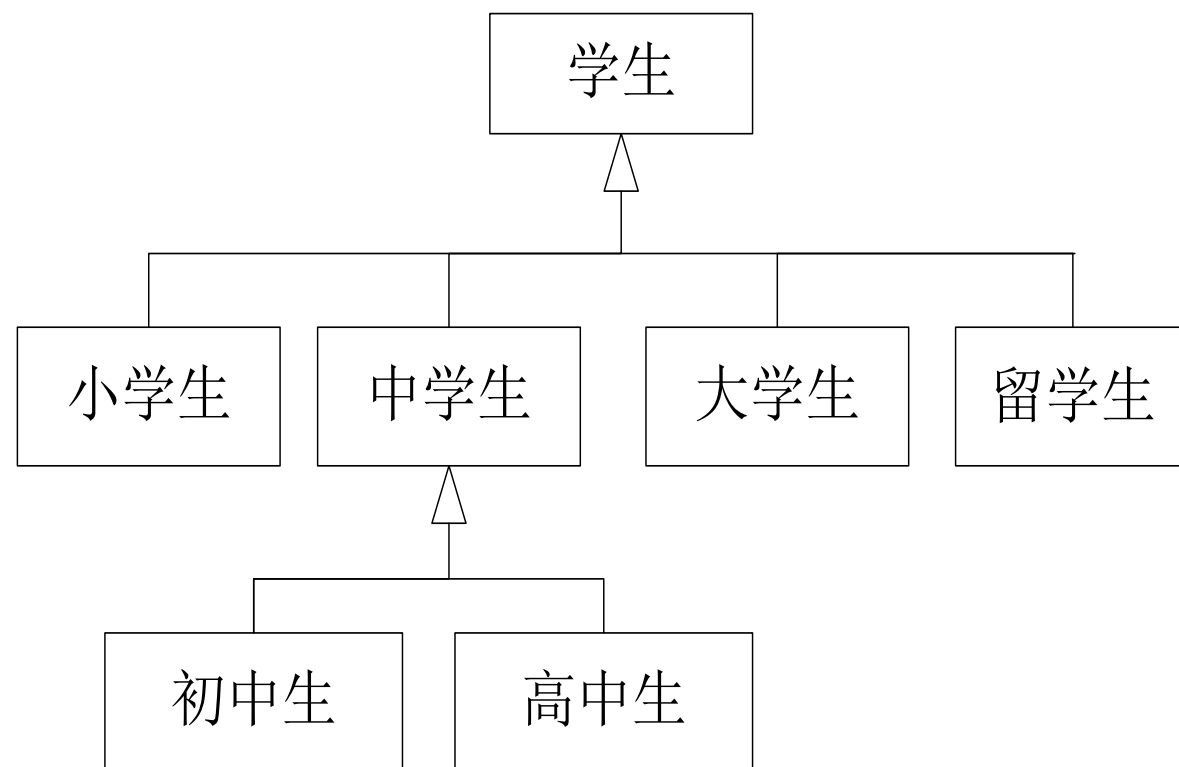
大连理工大学
主讲人-赵小薇





- 通过观察客观世界可以发现，世间的万事万物都不是孤立的,代码的设计过程往往能够反映出人们认识世界和描述世界的方式。





继承的概念



- 继承 (inheritance) 是一种联结类与类的层次模型。有了继承，类与类之间不再是彼此孤立的，一些特殊的类可以自动地拥有一些一般性的属性与行为，而这些属性与行为并不是重新定义的，而是通过继承的关系得来的。



从程序的角度理解继承



Student

```
char* name;  
char sex;  
int number;  
char* school;  
void print( );
```

CollegeStudent

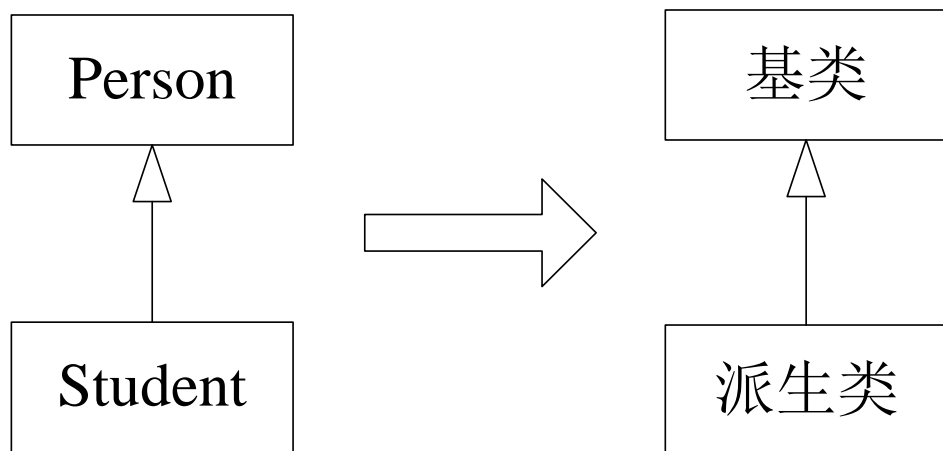
```
char* name;  
char sex;  
int number;  
char* school;  
char* major;  
void print( );
```



两个类有何异同?



继承与派生



➤ 面向对象语言中的继承就是以已经存在的类为基础构建一个新的类。

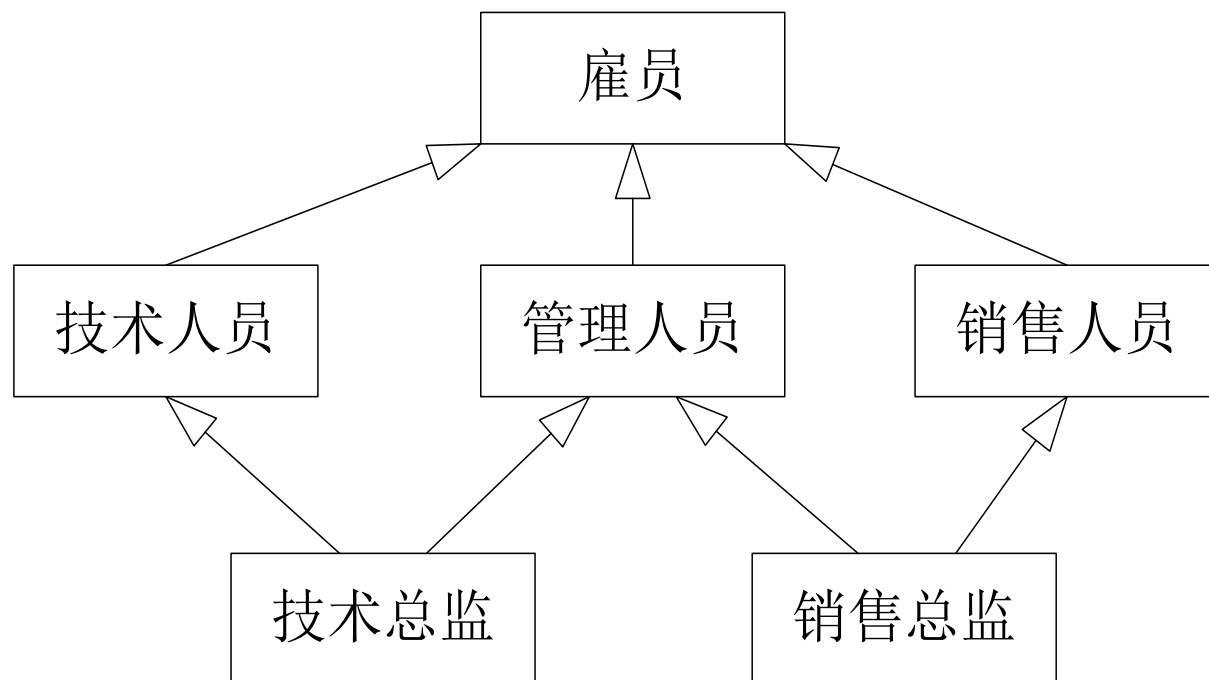
- 已存在的类称为基类 (base class) 或父类 (father class)
- 新建立的类称为派生类 (derived class) 或子类 (son class)



多重继承

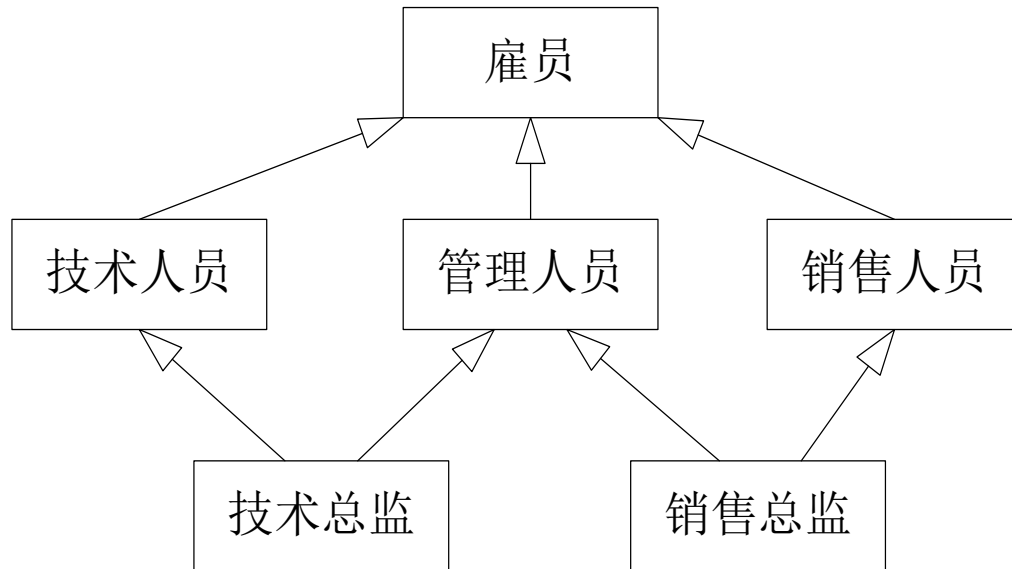


- 一个派生类有两个或者多个基类，这种继承称为多重继承或多继承。





- 在类族中，直接参与派生出其他类的基类被称为直接基类，基类的基类甚至更高层的基类，称作间接基类。
- C++中的继承具有传递性，具有非对称性，不能够循环继承。



派生类的定义



C++中单继承派生类的定义形式如下：

```
class 派生类名 : [继承方式] 基类名  
{  
    派生类成员声明;  
};
```

- “继承方式” 包括 **public** (公有继承) , **private** (私有继承) 和 **protected** (保护继承)
- 继承方式是可选的, 如果不显式声明继承方式, 那么默认是 **private** (私有继承) 。



举例



```
class CollegeStudent : public Student
{
private:
    char* major;
public:
    CollegeStudent ();
    ~CollegeStudent( );
    void print( );
};
```

// 专业

// 构造函数
// 析构函数
// 输出函数

派生类(子类)

基类(父类)

继承方式



派生类的定义



C++中多重继承派生类的定义形式如下:

```
class 派生类名: [继承方式] 基类名1, [继承方式] 基类名2,..., [继承方式] 基类名n  
{  
    派生类成员声明;  
};
```

注意下面的这种情况:

```
class Assistant : protected Student, Teacher  
{  
    .....  
};
```



Assistant类对Teacher类的继承方式是什么方式?

