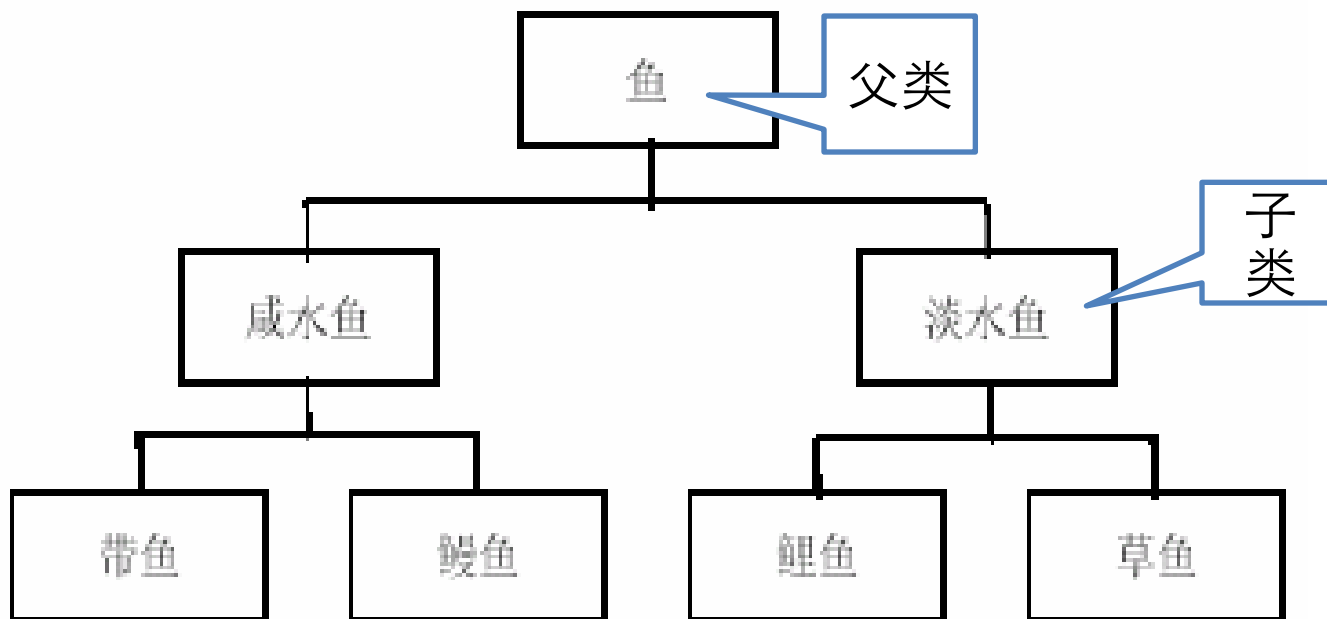


# 第八章 类的继承

---



- 继承：是使用已存在的类作为基础建立新类的技术。



- java只支持单继承，不允许多继承。



## □ 使用继承

### ➤ 编写父类

```
class 父类 {  
    //公共的属性和方法  
}
```

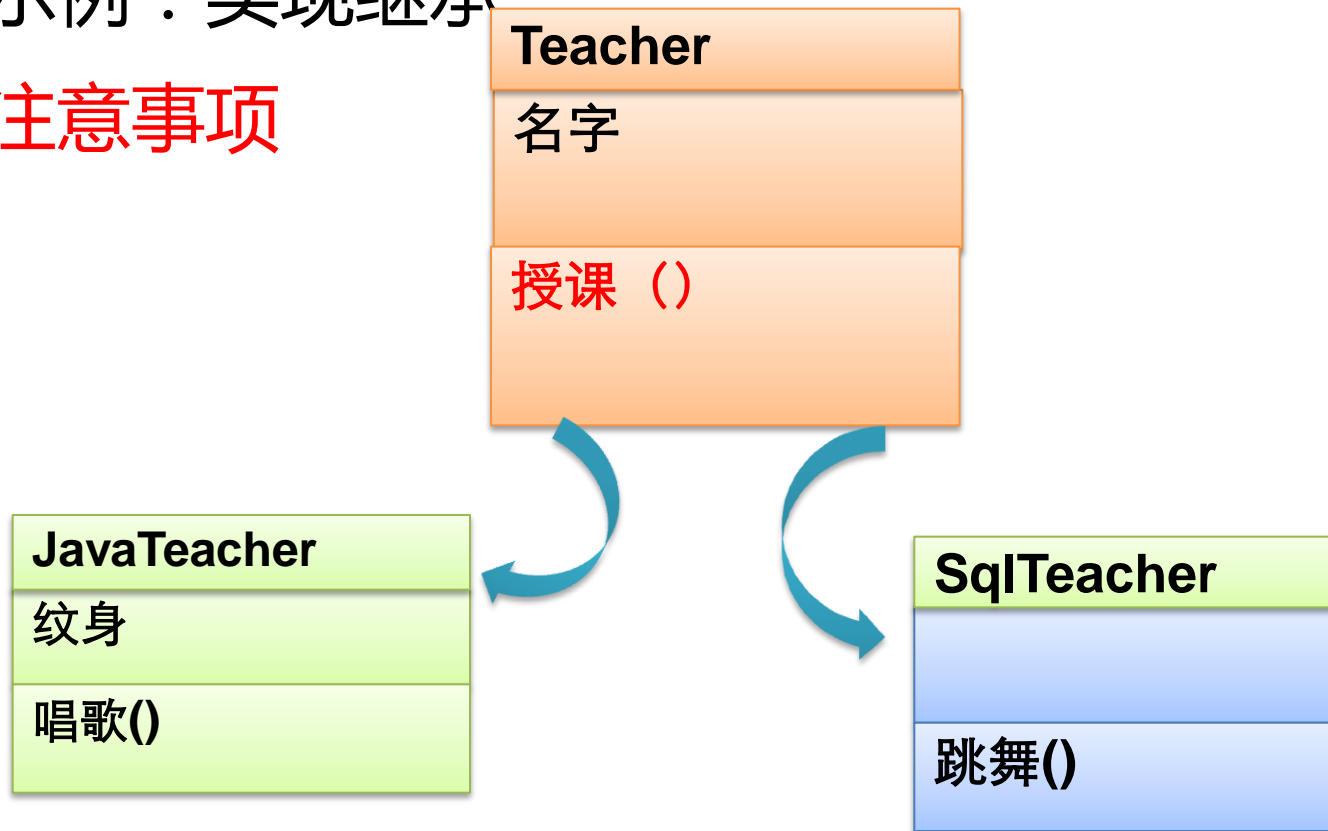
### ➤ 编写子类，继承父类

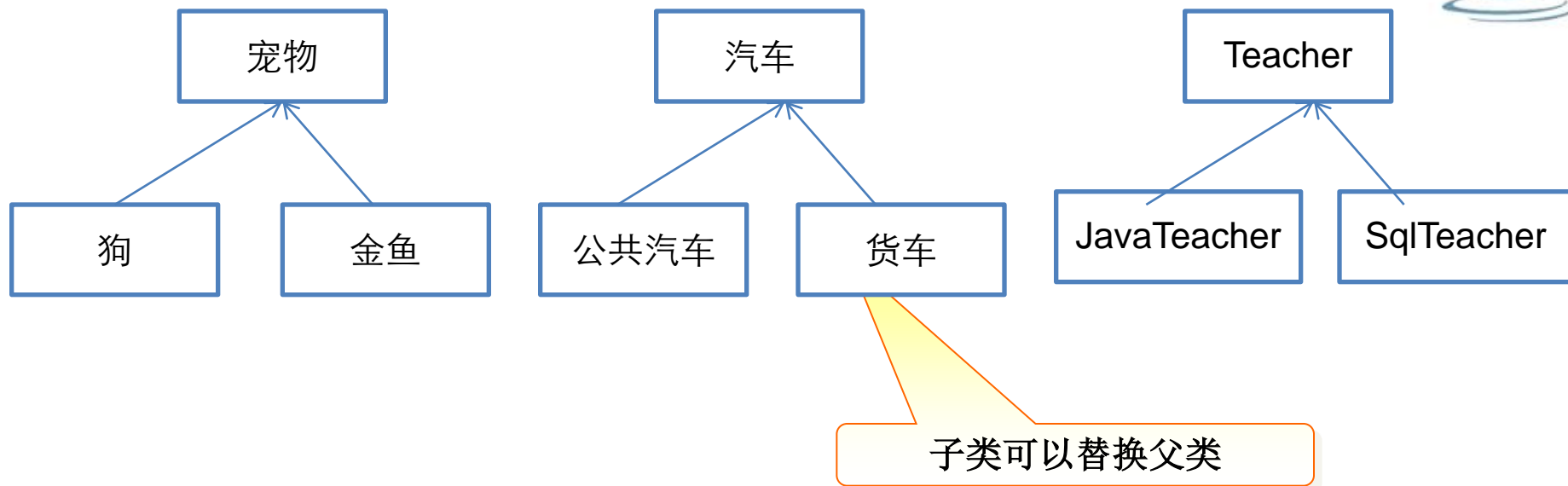
```
class 子类 extends 父类{  
    //子类特有的属性和方法  
}
```



❑ 示例：实现继承

❑ 注意事项





## ➤ 里氏替换原则（LSP）

- 在一个软件系统中，子类对象可以替换所有使用的父类对象，且程序行为没有变化



□ 子类中有和父类相同签名的方法，会如何？

```
class Teacher{  
    public void giveLesson(){  
    }  
}  
class JavaTeacher extends Teacher{  
    public void giveLesson(){  
    }  
}
```

子类**重写**父类方法



## □ 子类访问父类成员

使用`super`关键字

➤ 访问父类属性、方法



## □ 构造方法也会被重写吗？

不能被继承因此不能重写

## □ 如何在子类中调用父类的构造方法？

```
super();
```

```
super(参数表);
```



# Object类



## Object类

| 方法   | 用途                  |
|--|---------------------|
| <code>Object clone( )</code>   | 创建一个和被复制的对象完全一样的新对象 |
| <code>boolean equals(Object object)</code>                             | 判定对象是否相等            |
| <code>void finalize( )</code>  | 在一个不常用的对象被使用前调用     |
| <code>Class getClass( )</code>   | 获取运行时一个对象的类         |
| <code>int hashCode( )</code>   | 返回调用对象有关的散列值        |
| <code>void notify( )</code>  | 恢复一个等待调用对象线程的执行     |
| <code>void notifyAll( )</code>   | 恢复所有等待调用对象线程的执行     |
| <code>String toString( )</code>  | 返回描述对象的一个字符串        |
| <code>void wait( )</code><br><code>void wait(long milliseconds)</code> | 等待另一个线程的执行          |



## □ 修饰

➤ 类

➤ 方法

➤ **变量**