# Python面向对象 教案

周次：3

课次：2

章节内容：继承与多态

课时：2

## 单元教学目标

- 知识目标：能够区分类与子类的关系，并理解继承的语法结构及超类的调用方式   
- 技能目标：能够编写继承关系的类，并正确实现方法重写，完成多态的简单示例代码   
- 素养目标：能够遵循代码规范，编写结构清晰的代码，并在团队协作中有效沟通继承与多态的应用场景

## 教学重点

• 继承的结构与父类子类关系：掌握类的继承语法，理解父类与子类的层级关系，明确子类如何继承父类的属性和方法   
• 方法重写与类型检查：理解子类如何覆盖父类方法，掌握`super()`函数的使用场景，掌握类型检查（如`isinstance()`和`issubclass()`）的实践应用   
• 多态的核心概念：明确多态的定义（运行时绑定），理解方法重载（Python不支持，但可通过参数数量/类型实现类似效果）与类型检查的关联性

## 教学难点

• 学生可能难以理解继承的层次结构与方法覆盖机制，尤其是父类方法在子类中被覆盖或重写时的逻辑关系，容易混淆继承的深度与广度。例如，当子类重写父类方法时，如何判断调用的是哪个版本的函数，需要结合对象实例的类型进行动态绑定。   
  
• 多态的动态绑定概念可能抽象，学生容易陷入“方法是否在子类中定义”而非“如何运行时选择正确方法”的误区。例如，当子类重写了父类方法时，如何通过对象类型判断调用的是哪个版本的函数，需要结合`super()`函数和运行时类型检查（如`isinstance()`）进行说明。