

Python面向对象编程

Python面向对象编程

- ◆ 面向对象概述
- ◆ 面向对象的特点
- ◆ 综合案例

面向对象概述

◆ 类

技术解释：类是定义对象的模板或蓝图。它定义了对对象的属性（数据）和行为（方法或功能）。

举例：您可以把类想象成是一个蓝图或者是食谱。比如，一个蛋糕的食谱会告诉您需要哪些原料和制作方法，但食谱本身不是蛋糕。

面向对象概述

◆ 属性

技术解释：属性是类或对象的变量，它存储了关于对象的信息或状态。

举例：车的颜色、品牌、型号等信息就是车的属性。它们描述了车的特征。

面向对象概述

◆ 行为

技术解释：行为通常是类中定义的方法，它描述了对象可以执行的操作。

举例：汽车可以启动、加速和刹车。这些都是汽车的行为，描述了汽车可以做什么。

面向对象概述

◆ 对象

技术解释：对象是类的实例。您可以根据一个类创建多个对象，每个对象都有自己的属性（数据）和行为（方法）。

举例：如果类是蛋糕的食谱，那么对象就是根据食谱制作出来的实际的蛋糕。您可以根据同一个食谱制作出许多蛋糕，每个蛋糕都是一个对象。

面向对象的特点

◆ 封装

技术解释：封装是指将对象的状态（属性）和行为（方法）包装在一起，并限制外部直接访问对象的内部细节（例如，直接修改对象的某个属性）。通过封装，对象的内部实现可以自由修改，而不影响到外部代码。

举例：就像我们去银行通过柜员机拿钱，我们只需要输入账号密码，还有取款金额，就可以拿到钱，里面的细节我们不需要了解。

面向对象的特点

◆ 继承

技术解释：在面向对象编程（OOP）中，继承是一种机制，允许我们定义一个新的类（子类）基于一个已有的类（父类）。子类继承了父类的属性和方法，并可以添加自己的属性和方法，或重写父类的方法以实现不同的行为。

举例：扩展之前的例子，在BankAccount基础上，做出可以计算利息的SavingsAccount

面向对象的特点

◆ 多态

技术解释：多态是指不同类的对象对同一消息（方法调用）可以作出不同的响应。这意味着不同类的对象可以使用相同的接口（方法名），但具体实现（方法的内部代码）可以不同。

举例：同样是计算面积，圆形和矩形的计算方法有所不同

综合案例

◆ 神经网络的继承（看操作演示）