

Day13-Python 面向对象基础

一、学习目标

- 理解类与对象的概念
- 掌握类的属性与方法
- 掌握构造函数 `__init__` 的用法
- 理解继承与多态的基本原理
- 能独立定义类并创建实例对象

二、知识点讲解

1. 类与对象的定义

使用 `class` 定义类；用 `类名()` 创建对象实例。

```
class Animal:
    pass

a = Animal()
print("对象类型:", type(a))
print("是否是 Animal 的实例:", isinstance(a, Animal))
```

2. 类的属性与方法

属性保存对象状态；方法定义对象行为，第一个参数必须是 `self`。

```
class Dog:
    def __init__(self):
        self.name = "小黑"
        self.age = 2
```

```
def bark(self):
    print(f"{self.name}: 汪汪! ")

d = Dog()
print("名字:", d.name)
print("年龄:", d.age)
d.bark()
```

3. 构造函数 `__init__`

`__init__` 在对象创建时自动调用，用于初始化属性，可传参设置初始值。

```
class Student:
    def __init__(self, name, grade):
        self.name = name
        self.grade = grade

    def info(self):
        print(f"{self.name} 的年级是 {self.grade}")

s = Student("林", "高三")
s.info()
```

4. 继承与多态

子类可继承父类属性方法，并可重写方法实现多态。

```
class Person:
    def __init__(self, name):
        self.name = name

    def speak(self):
        print(f"{self.name} 在说话")

class Student(Person):
```

```
def speak(self):
    print(f"{self.name} 正在回答问题")

p = Person("老师")
s = Student("学生")
p.speak()
s.speak()
```

三、任务实战

定义类 Person，包含属性 name 和 age，方法 greet() 打印 '你好, {name}!'。

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def greet(self):
        print(f"你好, {self.name}! ")

p = Person("点头", 28)
p.greet()
```

四、总结

- 类是对象的蓝图，对象是类的实例
- 属性描述状态，方法定义行为
- `__init__` 用于初始化新对象的属性
- 继承可复用父类代码，多态让同名方法有不同表现