

认识面向对象

讲师：尚玉杰

本章目录

- 面向过程与面向对象
- 类和对象
- 属性与方法

面向过程与面向对象

- 面向过程：

- 事件为中心：分析出解决问题所需的**步骤**，然后用函数把这写步骤实现，并按顺序调用
- 维护、复用、扩展性较差



- 面向对象：

- 与面向过程相辅相成，在软件开发过程中，**宏观**上，用面向**对象**来把握事物间复杂的关系
- 性能低于面向过程

类和对象

- 类：
 - 对象的抽象，一类事物的总称
- 对象：
 - 类的具象，一个具体的事物
- 先有类，通过类可以创建对象

练习

- 区 分 类 和 对 象：
 - 狗
 - 奥迪A8
 - 电脑
 - 我的电脑
 - 苹果
 - 我现在坐的椅子
 - 男人
 - 我的女朋友

创建类

`class` 类名(父类):

属性

方法

- 类中包含属性（特征）和方法（行为）
 - 它是怎样的
 - 它可以做什么
- 查询类中的所有内容
 - 类名.__dict__

创建对象

对象名 = 类名 ()

对象名. 属性名

对象名. 方法名 ()

- 为当前对象添加新属性：

对象名. 属性名 = 属性值

为当前对象添加新方法

- 先导入types模块，并调用方法
 - `Method Type(function, instance)`
 - `function`为要向对象中添加的方法，`instance`为接受方法的对象

```
def eat(self):  
    print("吃东西")  
xiaoMing.eat = types.Method Type(eat, xiaoMing)  
xiaoMing.eat()
```


类属性/实例属性

- 类属性（类变量）：
 - 属于类
 - 当前类的所有对象该属性值都一样
 - 类变量推荐直接用类名访问，但也可以使用对象名访问（不推荐）
 - 改变类变量的值会作用于该类所有的实例化对象
 - 可以通过类名在类外添加新的类属性

类属性/实例属性

- 实例属性（实例变量）：初始化方法

```
def __init__(self, name):
```

```
    self.name = name
```

#self: 在类内使用，类外无效

- 属于对象
- 当前类的对象可以拥有各自的属性值
- 实例变量只能通过对象名访问，无法通过类名直接访问
- 实例属性不仅可以在__init__里面添加，还可以在类的其他方法中添加
- 查询对象中所有属性：对象.__dict__

类属性/实例属性

- 注意：

- Python 允许通过对象访问类变量，但无法通过对象修改类变量的值
 - 通过对象修改类变量的值，不是在给“类变量赋值”，而是定义新的实例变量
- 实例变量和类变量可以同名，但是在这种情况下，使用类对象将无法调用类变量
- 修改一个对象的实例变量，既不会影响类变量的值，也不会影响其它对象的实例变量
- 练习：验证以上规则

删除对象的属性

- `delattr` (对象名, "属性名")
- `del` 对象名. 属性名
- `del`和`delattr`功能有限, 只针对实例对象, 类属性删除不了

系统类/自定义类（了解）

- `list`
- `list1 = list()`
- `list1.append()`

练习

- 创建一个英雄类：
 - 包含英雄的各项属性：例如血量魔法值等等（注意哪些应该是类属性，哪些应该作为实例属性）
 - 英雄有自残的能力，自残后会掉血
- 创建一个狗类：
 - 包含名字，颜色，品种，年龄，性别等属性
 - 有一个自我介绍的方法，打印自身的属性信息（我叫XXX。。。）
 - 狗有看家的能力，狗有叫的能力，在看家方法中调用叫的方法

00A/00D/00P

- Object-Oriented Analysis: 面向对象分析法
 - 指的是在一个系统的开发过程中进行了系统业务调查以后，按照面向对象的思想来分析问题
 - 00A的主要原则。
 - 1. 抽象：从许多事物中舍弃个别的、非本质的特征，抽取共同的、本质性的特征，就叫做抽象
 - 2. 封装：把对象的属性和服务结合为一个不可分的系统单位，并尽可能隐蔽对象的内部细节
 - 3. 继承：特殊类的对象拥有的其一般类的全部属性与服务
 - 4. 聚合：又称组装，其原则是 把一个复杂的事物看成若干比较简单的事物组装体，从而简化对复杂事物的描述。
 - 5. 关联：是人类思考问题时经常运用的思想方法，通过一个事物联想到另外的事物

OOA/OOD/OOP

- 面向对象设计 (Object-oriented Design, OOD)
 - 该方法是oo方法中一个中间过渡环节
 - 主要作用是对OOA分析的结构作进一步的规范化整理, 以便能够被oop直接接受
 - “根据需求决定所需的类、类的操作以及类之间关联的过程”
 - 是一个软件设计方法, 是一种工程化规范
- 面向对象编程 (Object Oriented Programming, OOP, 面向对象程序设计) 是一种计算机编程架构