

Python 进阶训练营

尹会生

⑤ 面向对象与设计模式



目录

CONTENTS

- 01 新式类的继承、重载、多态
- 02 魔术方法
- 03 设计模式
- 04 元类与类工厂
- 05 包和模块

学习目标

- 1.掌握面向对象编程、单例模式、工厂模式
- 1.掌握元类、C3 算法、MRO、魔术方法
- 1.掌握元编程与工厂函数
- 了解异常处理机制的原理和用法

面向对象编程

现实世界的对象和 Python 世界的对象

对象是一个数据以及相关行为的集合

Python 的经典类与新式类

类的两大成员： 属性和方法

属性

类属性与对象属性

类属性字段在内存中只保存一份

对象属性在每个对象都保存一份

属性描述符 property

描述符：实现特定协议的类

property 类需要实现 `__get__`、`__set__`、`__delete__` 方法

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

```
    def __get__(self):
        return self.name
```

```
    def __set__(self, value):
        self.name = value
```

```
pythonteacher = Teacher('yin')
```

```
pythonteacher.name = 'wilson'
print(pythonteacher.name)
```

属性作用域

私有成员和公有成员的命名

下划线开头

双下划线开头

特殊成员（魔术方法）

方法

三种方法：

- 普通方法 至少一个 `self` 参数，表示该方法的对象
- 类方法 至少一个 `cls` 参数，表示该方法的类
- 静态方法 由类调用，无参数

三种方法在内存中都归属于类

类的特殊成员——魔术方法

双下划线开头和结尾的方法，实现了类的特殊成员，这类称作魔术方法

不是所有的双下划线开头和结尾的方法都是魔术方法

魔术方法类似其他语言的接口

鸭子类型

自定义类实现自带的数据类型

如：

- `__call__` 允许类的实例成为可调用对象
- `__str__` 被 `str()` 调用时的行为
- `__getitem__` 使用 `[key]` 调用时的行为
- `__iter__` 被迭代时的行为

面向对象编程

类的继承

- 单一继承

- 多重继承

- 菱形继承（钻石继承）

- 继承机制 MRO

- MRO 的 C3 算法

类的重载

类的多态

设计模式

装饰器模式

单实例模式

通过 instance 方法 实现单实例模式

通过 __new__() 魔术方法实现单实例模式

SOLID 原则

抽象基类

抽象基类 (Abstract Base Class, ABC)

```
from abc import ABC
```

```
class MyABC(ABC):  
    pass
```

```
MyABC.register(tuple)
```

```
assert issubclass(tuple, MyABC)  
assert isinstance(), MyABC)
```

元类（元编程）

动态属性

工厂函数

工厂类（类工厂函数）

示例：

生产对象的工厂，负责接收传进来的数字字符串，生成对象的对象返回

调用 `factory()` 方法实现设计模式中的 Simple Factory 模式

THANKS! |  极客大学