Python面向对象语法精讲

嵩 天

面向对象编程模式

嵩 天

面向对象编程模式

单元开篇

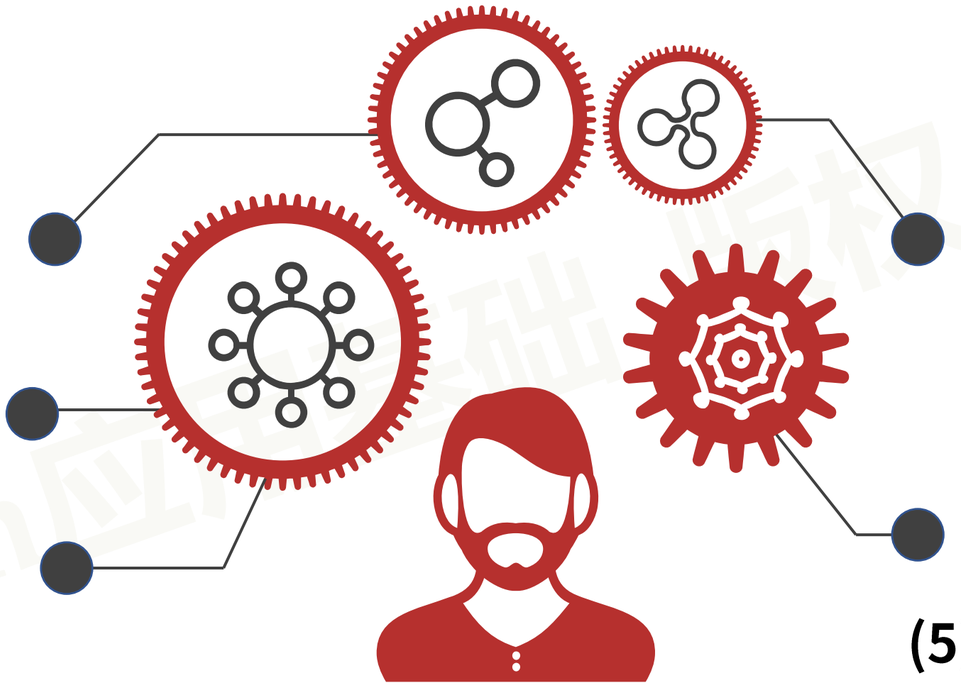
Python123

单元开篇

(3) 面向对象的三个特征

(2) 面向对象编程思想

(1) 万物皆对象



(4) Python面向对象术语概述

(5) Python面向对象实例入门

面向对象编程模式

面向对象编程模式

万物皆对象

万物皆对象

万物皆对象

万物皆对象

对象：独立的存在 或 作为目标的事物

- 独立性：对象都存在清晰的边界，重点在于划分边界
- 功能性：对象都能表现出一些功能、操作或行为
- 交互性：对象之间存在交互，如：运算和继承

万物皆对象

Python语言的“万物皆对象”

- Python语言中所有数据类型都是对象、函数是对象、模块是对象
- Python所有类都继承于最基础类object
- Python语言中数据类型的操作功能都是类方法的体现

面向对象编程模式

面向对象编程思想

面向对象编程

OOP：Object-Oriented Programming

- OOP：面向对象编程，一种编程思想，重点在于高抽象的复用代码
- OOP把对象当作程序的基本单元，对象包含数据和操作数据的函数
- OOP本质是把问题解决抽象为以对象为中心的计算机程序

面向对象编程

OOP：Object-Oriented Programming

- OOP在较大规模或复杂项目中十分有用，OOP可以提高协作产量
- OOP最主要价值在于代码复用
- OOP只是一种编程方式，并非解决问题的高级方法

面向对象编程

面向过程 vs. 面向对象

- 面向过程：以解决问题的过程步骤为核心编写程序的方式
- 面向对象：以问题对象构建和应用为核心编程程序的方式
- 所有OOP能解决的问题，面向过程都能解决

面向对象编程概念实例

商品价格统计

- 有3个商品：电脑、打印机和投影仪
- 每个商品有1个原始售价和1个折扣售价
- 求3个商品原始售价的和以及折扣售价的和

面向对象编程概念实例

面向过程的程序解法

列表1，存商品：电脑、打印机、投影仪

列表2，存原价：电脑原价、打印机原价、投影仪原价

列表3，存实价：电脑实价、打印机实价、投影仪实价

根据列表1、列表2和列表3进行计算，列表1/2/3功能需要文档标记

面向对象编程概念实例

面向对象的程序解法

设计“商品”类，属性包括：商品名、原价和实价

从“商品”类产生3个对象：电脑、打印机、投影仪

这三个对象中保存商品名、原价、实价等与商品相关的信息

使用对象1、对象2和对象3进行计算

面向对象编程模式

面向对象的三个特征

面向对象的三个特征

OOP三个重要特征

- 封装：属性和方法的抽象，用数据和操作数据的方法来形成对象逻辑
- 继承：代码复用的高级抽象，用对象之间的继承关系来形成代码复用
- 多态：方法灵活性的抽象，让对象的操作更加灵活、更多复用代码

封装的理解

封装 Encapsulation：属性和方法的抽象

- 属性的抽象：对类的属性（变量）进行定义、隔离及保护
- 方法的抽象：对类的方法（函数）进行定义、隔离及保护
- 目标是形成一个类对外可操作属性和方法的接口

继承的理解

继承 Inheritance：代码复用的高级抽象

- 继承是面向对象程序设计的精髓之一
- 实现了以类为单位的的高抽象级别代码复用
- 继承是新定义类能够几乎完全使用原有类属性与方法的过程

多态的理解

多态 Polymorphism：仅针对方法，方法灵活性的抽象

- 参数类型的多态：一个方法能够处理多个类型的能力
- 参数形式的多态：一个方法能够接受多个参数的能力
- 多态是OOP的一个传统概念，Python天然支持多态，不需要特殊语法

面向对象编程模式

Python面向对象术语概述

Python123

常用术语概述

类 Class 和 对象 Object

- 类：逻辑抽象和产生对象的模板，一组变量和函数的特定编排
- 对象：具体表达数据及操作的实体，相当于程序中的“变量”
- 实例化：从类到对象的过程，所有“对象”都源于某个“类”

常用术语概述

面向对象术语概述

- 对象：类对象、实例对象
- 属性：存储数据的“变量”，包括：类属性、实例属性
- 方法：操作数据的“函数”

  包括：类方法、实例方法、自由方法、静态方法、保留方法

常用术语概述

类对象 vs. 实例对象

- 类对象：Class Object，维护每个Python类基本信息的数据结构
- 实例对象：Instance Object，Python类实例后产生的对象，简称：对象
- 这是一组概念，类对象全局只有一个，实例对象可以生成多个

常用术语概述

面向对象术语概述

- 三个特性：封装、继承、多态
- 继承：基类、派生类、子类、父类、超类、重载
- 命名空间：程序元素作用域的表达
- 构造和析构：生成对象和删除对象的过程

Python面向对象术语

实例对象 (Instance Object)

实例属性 (Instance Attributes)

__init__()

__del__()

对象引用 (Object Reference)

类(Class)的构建

默认

类对象 (Class Object)

类属性 (Class Attributes)

@classmethod → 类方法(Class Method)

def - self → 实例方法(Instance Method)

def → 自由方法(Namespace Method)

@staticmethod → 静态方法(Static Method)

__XX__ → 保留方法(Special Method)

面向对象编程模式

Python面向对象实例入门

Python123

面向对象编程概念实例

商品价格统计

- 有3个商品：电脑、打印机和投影仪
- 每个商品有1个原始售价和1个折扣售价
- 求3个商品原始售价的和以及折扣售价的和

Python OOP实例

```
class Product():
    def __init__(self, name):
        self.name = name
        self.label_price = 0
        self.real_price = 0

c = Product("电脑")
d = Product("打印机")
e = Product("投影仪")
c.label_price, c.real_price = 10000, 8000
d.label_price, d.real_price = 2000, 1000
e.label_price, e.real_price = 1500, 900
s1, s2 = 0, 0
for i in [c, d, e]:
    s1 += i.label_price
    s2 += i.real_price
print(s1, s2)
```

Python OOP实例

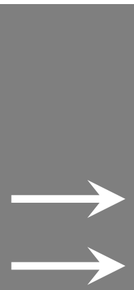
```
class Product():
    def __init__(self, name):
        self.name = name
        self.label_price = 0
        self.real_price = 0

c = Product("电脑")
d = Product("打印机")
e = Product("投影仪")
c.label_price, c.real_price = 10000, 8000
d.label_price, d.real_price = 2000, 1000
e.label_price, e.real_price = 1500, 900
s1, s2 = 0, 0
for i in [c, d, e]:
    s1 += i.label_price
    s2 += i.real_price
print(s1, s2)
```

定义了抽象的Product类

保存：名字和价格等信息

Python OOP实例



```
class Product():
    def __init__(self, name):
        self.name = name
        self.label_price = 0
        self.real_price = 0

c = Product("电脑")
d = Product("打印机")
e = Product("投影仪")
c.label_price, c.real_price = 10000, 8000
d.label_price, d.real_price = 2000, 1000
e.label_price, e.real_price = 1500, 900
s1, s2 = 0, 0
for i in [c, d, e]:
    s1 += i.label_price
    s2 += i.real_price
print(s1, s2)
```

Product类

初始化价格为0

Python OOP实例

```
class Product():
    def __init__(self, name):
        self.name = name
        self.label_price = 0
        self.real_price = 0

c = Product("电脑")
d = Product("打印机")
e = Product("投影仪")
c.label_price, c.real_price = 10000, 8000
d.label_price, d.real_price = 2000, 1000
e.label_price, e.real_price = 1500, 900
s1, s2 = 0, 0
for i in [c, d, e]:
    s1 += i.label_price
    s2 += i.real_price
print(s1, s2)
```

产生3个对象

分别对应3个具体商品

Python OOP实例

```
class Product():
    def __init__(self, name):
        self.name = name
        self.label_price = 0
        self.real_price = 0

c = Product("电脑")
d = Product("打印机")
e = Product("投影仪")
c.label_price, c.real_price = 10000, 8000
d.label_price, d.real_price = 2000, 1000
e.label_price, e.real_price = 1500, 900
s1, s2 = 0, 0
for i in [c, d, e]:
    s1 += i.label_price
    s2 += i.real_price
print(s1, s2)
```

为3个对象商品赋值价格

Python OOP实例

```
class Product():
    def __init__(self, name):
        self.name = name
        self.label_price = 0
        self.real_price = 0

c = Product("电脑")
d = Product("打印机")
e = Product("投影仪")
c.label_price, c.real_price = 10000, 8000
d.label_price, d.real_price = 2000, 1000
e.label_price, e.real_price = 1500, 900
s1, s2 = 0, 0
for i in [c, d, e]:
    s1 += i.label_price
    s2 += i.real_price
print(s1, s2)
```

求和并输出价格

价格由label_price和real_price表示

易于理解

面向对象编程模式

单元小结

单元小结

(3) 面向对象的三个特征

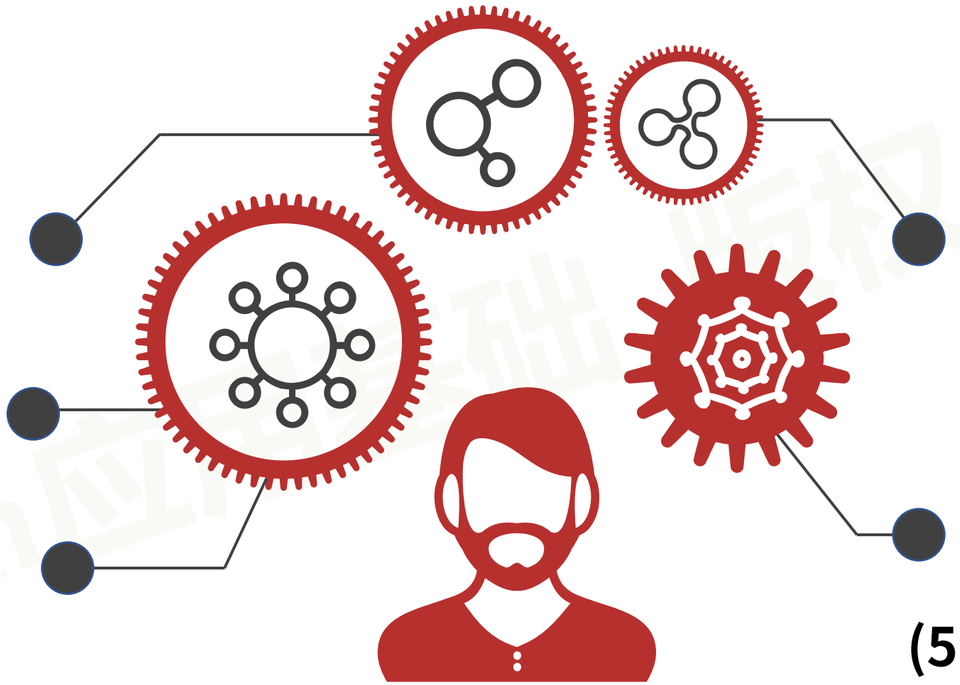
封装、继承、多态

(2) 面向对象编程思想

OOP价值、面向过程/面向对象

(1) 万物皆对象

自然界和Python语言的对象



(4) Python面向对象术语概述

类、对象、属性、方法…

(5) Python面向对象实例入门

体会Python面向对象编程

面向对象编程模式


Python 123
Thank you