§3.4 Python并发与并行程序设计
（语言基础部分）

徐悦甦(Yueshen Xu)

ysxu@xidian.edu.cn

软件工程系

西安电子科技大学

本节提纲

西安电子科技大学 XIDIAN UNIVERSITY

- Python语言基础
- Python基础语法

Python语言

- Python面向过程编程剩余
- Python面向对象编程
  - 类
  - 继承
    - 多继承
  - 封装
  - 类属性与类方法
  - 静态方法

Python面向对象

关键字： Python | 面向对象 | 定义类 | 继承与封装

第三次上机

西安电子科技大学 XIDIAN UNIVERSITY

- 时间
  - 第三次上机，6月6号（星期三）
  - 晚18：30至21：30
  - 地点：G346、348
- 主题
  - Python基础编程
- 内容
  - 题目一：使用Python基础编程，统计两个文件中单词 ‘book’出现的总次数，并将结果写入到文件中（文件名自己决定）
    - 题目描述：有两个文件，file_1.dat与file_2.dat（文件类型也可以是其它文本文件类型，如.txt）
    - 文件内容如下：

第三次上机

- file_1.dat内容

and, with, we, me, university, with, book, computer, country, book

- file_2.dat内容

bag, boy, book, school, teacher, student, book, book

➢ 要求

- 正确地使用Python的文件操作，分别统计出file_1.dat中出现3次，file_2.dat中出现2次，然后统计出”book”出现的总次数（5次）
- 将总次数写入文件中（文件名自己决定）

第三次上机

- 题目二：使用Python基础编程，完成快速排序函数的编写
  - 输入：从文件data.dat或data.txt中输入一列没有顺序的数，如

    3, 7, 12, 5, 3, 10, 11, 9, 4, 2, 4
  - 输出：经过快排，正确的排序结果，如

    2, 3, 3, 4, 4, 5, 7, 9, 10, 11, 12
- 要求：
  - 使用Python实现快速排序的函数
  - 将排序结果写入到文件中（文件名自己决定）

第三次上机

西安电子科技大学 XIDIAN UNIVERSITY

- 要求
  - 完成两道题目的程序编写，
  - 独立完成实验报告，发送电子版，可以是word，也可以pdf
  - 实验报告提交地址，同作业
    - xdsepc2018@163.com
    - 邮件命名规则：“实验报告三+学号+姓名”
    - 附件命名规则：“实验报告三+学号+姓名”
    - 上交时间：至第三次上机之后再上理论课之前，即6.18号，十天时间
    - 实验报告格式：见《并行计算实验报告结构及要求》

提醒第二次上机

西安电子科技大学
XIDIAN UNIVERSITY

- 时间与地点
  - 5月23号，G346、G348机房
  - 19：00~21：30
  - 主题：Java并行程序设计
  - 验收 + 实验报告

Python面向过程编程剩余

西安电子科技大学
XIDIAN UNIVERSITY

- 文件操作中的os模块

```
import os
os.mkdir(“myfolder")
os.getcwd()
os.listdir (”./”)
os.rmdir (”myfolder”)
os.path.exists()
os.path.join(path,name)
os.path.getsize(name)
os.path.abspath(name)
```

Python面向过程编程剩余

西安电子科技大学
XIDIAN UNIVERSITY

With语句

- Python 2.5开始，引入了with语句，
  - with语句适用于对资源进行访问的场合，
    - 确保不管使用过程中是否发生异常都会执行必要的“清理”操作，释放资源

```
with context_expr [as var]:
    body
```

  - context_expr：需要返回一个上下文管理器对象，该对象并不赋值给as子句中的 var
  - var：可以是变量或者元组
  - body：with语句包裹的代码块

Python面向过程编程剩余

西安电子科技大学
XIDIAN UNIVERSITY

With语句

```
file = open("/tmp/data.txt")
try:
    data = file.read()
finally:
    file.close()
```

```
with open("/tmp/data.txt") as file:
    data = file.read()
```

- 该代码虽然解决了产生异常的可能，但是这段代码过于冗长
- 此时，在示例中使用with语句处理上下文环境产生的异常

Python面向过程编程剩余

西安电子科技大学
XIDIAN UNIVERSITY

模块运行

- python提供了一个__name__属性，
- 每个模块都有一个__name__属性，当其值为‘__main__’时，
  - 表明该模块自身在运行，否则是被引用

```
if __name__=='__main__':
```

Python面向对象编程

西安电子科技大学
XIDIAN UNIVERSITY

类的定义

- 使用class关键字来声明一个类，基本格式如下：

```
class 类名:
    类的属性
    类的方法
```

- 根据类创建对象的语法格式如下：
  - 对象名 = 类名()
- 构造方法
  - 构造方法指的是__init__方法

Python面向对象编程

西安电子科技大学
XIDIAN UNIVERSITY

__init__与self

```
class Car:
    def __init__(self):
        self.color = “黑色”
    def print_content(self):
            print("%s的车在鸣笛..."%(self.color))
```

- 在方法的列表中，第1个参数永远都是self
- 当某个对象调用方法的时候，Python解释器会把这个对象作为第1个参数传给self
  - 开发者只需要传递后面的参数就可以了

Python面向对象编程

西安电子科技大学
XIDIAN UNIVERSITY

构造方法

- 构造方法指的是__init__方法
  - 当创建类的实例的时候，系统会自动调用构造方法
  - 从而实现对类进行初始化的操作

析构方法

- 当删除一个对象来释放类所占用资源的时候，
  - Python解释器默认会调用另外一个方法，即__del__( )方法
  - __del__方法被称为析构方法

Python面向对象编程

西安电子科技大学
XIDIAN UNIVERSITY

self

```
class Dog:
    def __init__(self, newColor):
        self.color = newColor
    def printColor(self):
        print("颜色为：%s"%self.color)
```

```
dog1 = Dog("白色")
dog1.printColor()
```

Python面向对象编程

西安电子科技大学 XIDIAN UNIVERSITY

运算符重载

- 运算符重载是通过实现特定的方法使类的实例对象支持Python的各种内置操作
- 例如：+运算符是类里提供的__add__这个函数
  - 当调用+实现加法运算的时候，
    - 实际上是调用了__add__方法

方法	说明	何时调用方法
__add__	加法运算	对象加法：x+y，x+=y
__sub__	减法运算	对象减法：x-y，x-=y
__mul__	乘法运算	对象乘法：x*y，x*=y
__diy__	除法运算	对象除法：x/y，x/=y

Python面向对象编程

西安电子科技大学
XIDIAN UNIVERSITY

继承

```
class 子类名(父类名):
```

- 假设有一个类为A，A派生出来了子类B，示例如下：

```
class B(A):
class A(object):
```

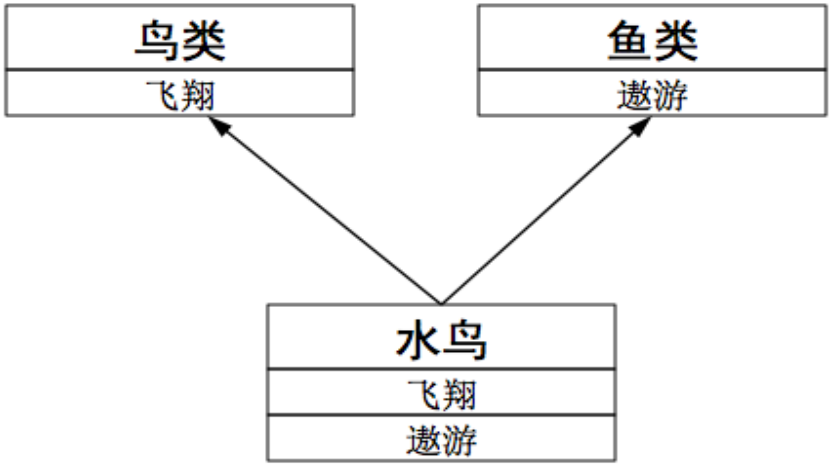
- 多继承

```
class 子类名(父类1，父类2...):
```

Python面向对象编程

多继承

- Python支持多继承，
- 多继承就是子类拥有多个父类，
- 并且具有它们共同的特征，
  - 即子类继承了父类的方法和属性



Python面向对象编程

西安电子科技大学
XIDIAN UNIVERSITY

多继承

```
class 子类名(父类1，父类2…):
```

- 如果子类继承的多个父类间是平行的关系，
  - 子类先继承的哪个类就会调用哪个类的方法

在继承关系中，子类会自动拥有父类定义的方法，

- 但是有时子类想要按照自己的方式实现方法，
  - 即对父类中继承来的方法进行重写，
  - 需要注意的是，
    - 在子类中重写的方法要和父类被重写的方法具有相同的方法名和参数列表

Python面向对象编程

西安电子科技大学
XIDIAN UNIVERSITY

封装

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age
laowang = Person(“老王”, 30)
laowang.age = 300
print(laowang.age)
```

- 人的年龄可以随便设置，不可行
- 为了保护类里面的属性，可以采用如下方式解决：
  - 1. 把属性定义为私有属性，即在属性名的前面加上两个下划线；
  - 2. 添加用于设置或获取属性值的两个方法供外界调用

Python面向对象编程

西安电子科技大学
XIDIAN UNIVERSITY

类属性

- 类属性是类所拥有的属性，
  - 它需要在类中进行显示地定义（位于类内部，方法的外面），
  - 它被所有类的实例对象所共有，在内存中只存在一个副本

类属性示例代码：

```
class Cat(object):
    #类属性
    num = 0
```

实例属性

- 通过“实例.属性”添加属性的属性都是实例属性

```
def __init__(self):
    #实例属性
    self.age = 1
```

Python面向对象编程

西安电子科技大学
XIDIAN UNIVERSITY

类方法

- 使用修饰器@classmethod来标识类方法

```
class 类名:
    @classmethod
    def 类方法名():
        方法体
```

要想调用类方法，

- 既可以通过对象名调用类方法，
- 又可以通过类名调用类方法

Python面向对象编程

西安电子科技大学
XIDIAN UNIVERSITY

- 静态方法
  - 使用修饰器@staticmethod来标识静态方法

```
class 类名:
    @staticmethod
    def 静态方法名():
        方法体
```

- 静态方法是没有self参数，
  - 在静态方法中无法访问实例变量
- 静态方法中不可以直接访问类属性，
  - 但是可以通过类名引用类属性

西安电子科技大学
XIDIAN UNIVERSITY

§3.4 Python并发程序设计

http://web.xidian.edu.cn/ysxu/

结构 **编程** 算法 应用