

解决现实世界的问题，不在于开发语言本身，而在于对方法论的参透和对现实世界的理解，编程语言只是个工具。

Python网络程序开发

郭立峰 2012年11月

课程目标

- 掌握程序开发世界的思想理论基础
- 掌握python程序开发的基础（容器，语法特性）
- 掌握python解释性脚本语言的特性
- 掌握python进行网络开发应用（网络开发，爬虫使用，B/S开发）

- 第1周 解释型脚本语言初探
- 第2周 python开发环境
- 第 3 周 python使用基础
- 第 4周 python的内置容器之要点，Map & Reduce的使用
- 第 5周 并行开发 fork & thread
- 第6 周 python的正则是怎样的
- 第 7周 socket 编程之基础
- 第 8 周 实现一个抓取代理服务器的小爬虫
- 第 9 周 django开发web应用初探
- 第 10 周 django高级应用
- 第 11周 twisted 实现网络应用服务器
- 第12 周 twisted的高级应用

第一周 解释性脚本语言初探

- 1) 编程世界的主流思想
- 2) 解释型脚本语言的特点
- 3) 编程语言之间的比较
- 4) 方法论
- 5) 现实世界的变化
- 6) 程序员境界
- 7) 程序员的陋习
- 8) 作业

1) 编程世界的主流思想

- a) 结构化编程，面向过程的开发
- b) 函数式编程
- c) ER实体关系
- d) 面向对象 *

-
- ```
graph TD; A[功能] --> B[子功能1]; A --> C[子功能2]; B --> D[子功能1]; B --> E[子功能2]; C --> F[子功能]
```



## 1b) 函数式编程

- $\lambda$ 演算
- LISP , Erlang
- 尾递归 栈的使用
- 避免防御式编程
- 参考书 《计算机程序的构造和解释》

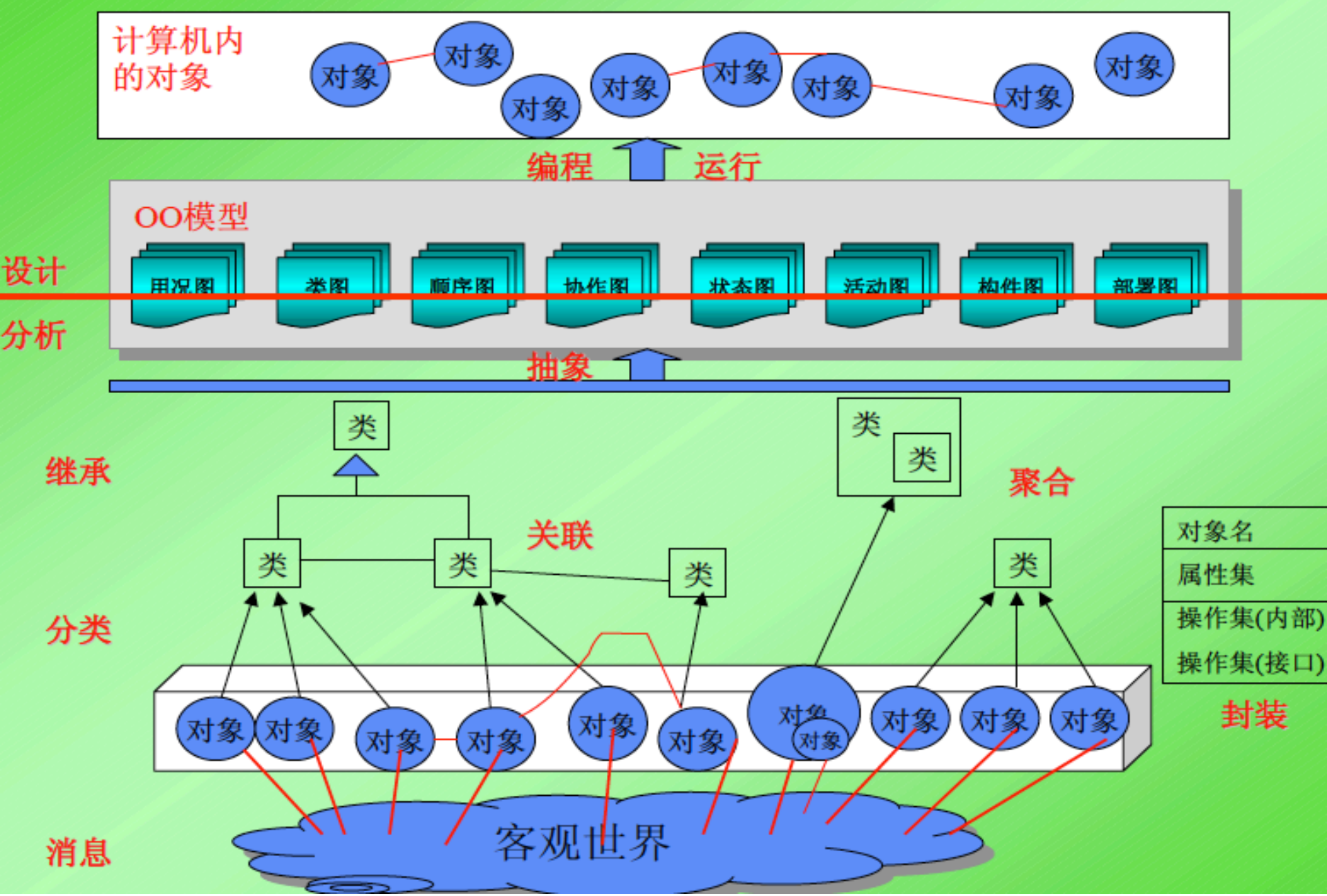
## 1c) ER实体关系

- 关系数据库 3NF
- 选择
- 投影
- 笛卡尔操作 join





# 1d)面向对象编程 (OOA OOD OOP)



## 2) 解释型脚本语言的特点

- 解释型和编译型
- 运行时和动态类型
- 强类型和弱类型
- 开发效率高
- 通用性强
- 内置方便的数据容器
- 易于扩展和嵌入

### 3) 编程语言的比较

|     | 嵌入式                     | 网络应用                                            | 网络界面                       | APP                    |
|-----|-------------------------|-------------------------------------------------|----------------------------|------------------------|
| 编译型 | C<br>C++                | Java<br>.net(c#)                                | flex AS<br>Java Applet     | Android<br>objective-c |
| 解释型 | Python<br>Erlang<br>lua | Python<br>Erlang<br>Perl<br>ASP<br>Shell<br>Lua | HTML<br>Javascript<br>lisp | HTML5<br>Lua           |

## 4) 方法论

### a) 笛卡尔方法论

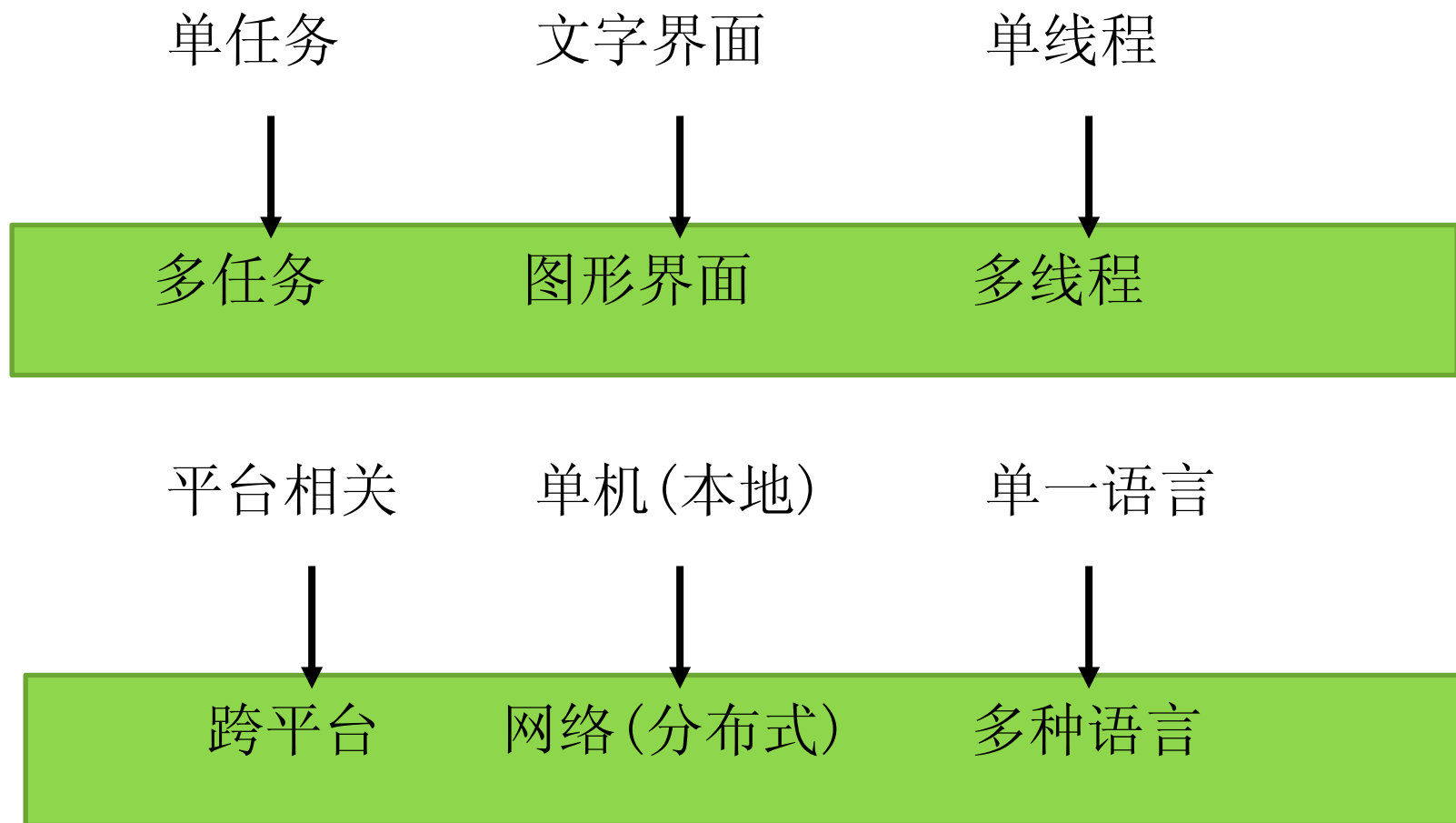
- ◆ 拆分
- ◆ 排序
- ◆ 处理
- ◆ 归并

### b) 软件工程过程方法论

- ◆ 软件生命周期
- ◆ RUP统一软件过程管理
- ◆ 敏捷开发

### c) 项目管理（计划、组织、执行、控制）

## 5) 现实世界的变化



## 6) 程序员的境界

- 线性级
- 逻辑级
- 架构级
- 工程级

## 7) 程序员的陋习

- 自我
- 封闭
- 惯性
- 沟通障碍
- 一叶障目
- 工作量估计乐观
- 拒绝变化
- 拒绝重构

## 8) 作业

- 1) 通读学习面向对象方法论，过程方法论，ER关系方法论的内容，形成学习要点笔记总结。
- 2) 互联网搜索收集总结比较C语言，.net系列语言，java，php，python的区别与特点。
- 3) 互联网上搜集整理解释性语言的内在特性。
  
- 注意：以上内容需要提交论坛上课程作业区。