

MICROOH 麦可网

Android-从程序员到架构师之路

出品人：Sundy

讲师：高焕堂（台湾）

<http://www.microoh.com>

A01_a

复习基本OOP技术(a)

By 高煥堂

内容

1. “-Oriented” 的涵意
2. 从对象(Object)谈起
3. 类的用途：叙述软件对象
4. <基类/子类> 结构用途(一)：表达继承
5. <基类/子类> 结构用途(二)：表达组合
6. <基类/子类> 结构的接口(卡榫函数)
7. IoC机制与Default函数
8. 主动型 vs. 被动型API
9. 接口与类(别)

1、 “-Oriented”的涵意

“-Oriented” 的涵意

- ◎ “-Oriented” 意味着一种信仰。
- ◎ “Object-oriented” 相信任何软件都是由对象所构成的，而且Nothing else。
- ◎ 上述信仰，电脑语言的设计就简化了，写程序只要定义类别(Class)就行了。

- ◎ " -Oriented"
- " -Based"
- " -Driven"
- " -Centered"

它们之间有何区别呢？

◎ “-Based” 的涵意；例如：
Requirement-based

--基于需求，有先后的顺序，例如先进行需求分析；然后基于需求分析的结果(文件)，而进行后续的软件开发活动；就称为：
Requirement-based software development

◎ “-Driven” 的涵意；

例如：Model-driven、Use Case-driven

--其实“-driven” 是“引导”，而不是大家常说的“驱动”。

-- 就向北极星引导我们，指出方向而已。
也向汽车司机(Driver)只是引导汽车方向，并没有去驱动汽车；而是引擎才是驱动汽车。

◎ “-Centered” 的涵意；
例如：Architecture-centered、
Architecture-centric

--一切软件开发的活动都围绕着架构，就像圣诞节的糖果和礼物都挂在圣诞树上一样。

练习

◎ 请问：Service-Oriented Architecture (SOA) 是什么涵意呢？

2、从对象(Object)谈起

对象(Object)

◎自然界含有各式各样的东西(Thing)，如阳光、绿野、铁路、行人等。人们随着阅历增长，对自然界的東西，认识愈多。对个人而言，所认识的东西，皆是「对象」(Object)。例如，李白心中最清楚的对象是他的诗，每首诗皆是对象。至于当时欧洲的英文诗，李白可能不认识，就非李白心中之对象了。

- ◎ 一旦认识某样东西，就能说出其特点，并与别的对象比较一番。其特点包括：
 - ◇ 对象之特征或属性(Attribute)。
 - ◇ 对象之行为(Behavior)。

例如，鸟儿的特征是：有翅膀、尾巴；其行为是：唱歌、会飞等等。

- ◎ 软件之对象(Software Object)是由数据(Data)和函数(Function)所组成。

3、类的用途：叙述软件对象

- ◎ 类(Class)是群体(或集合)，而对象是类中的一份子。人们常用「是一个」(Is A)来表达对象与类间之关系。

例如：

- ◎ 月亮是一个星球
- ◎ 毕加索是一个艺术家
- ◎ 毕加索是一个画家
- ◎ 张大千是一个画家

.....

- ◎ 所以「月球」是对象，属于「星球」类的一个实例(Instance)。毕加索是对象，艺术家是类，同样地画家也是类，其中画家是艺术家群体中的小群体（部分集合）。毕加索和张大千同属于「画家」类，所以具有共同特点：精于美术绘画

- ◎ 类是一群具有共同重要特性之对象。类之定义就是说明这群对象具有什么的重要特性。特性包括对象之特征及行为。软件中之对象以数据来表达特征，以函数来表达行为。因之，类之定义就是说明软件中之对象，应含那些数据及那些函数。

例如：定义类如下：

```
class Rose {  
    private double price;  
    private int month;  
    public void say(){  
        System.out.println("Color is RED.");  
    }  
}
```

- ◎ 这Rose就是我们新创的类别，将用来诞生对象，以描述自然界的玫瑰花。于是可诞生对象如下：

`Rose rose = new Rose();`

- ◎ 这new Rose()指令就诞生一个Rose类的对象

```
public class JMain {  
    public static void main(String[] args) {  
        Rose rose = new Rose();  
        rose.say();  
    }  
}
```

~ Continued ~