Object oriented Analysis &Design
面向对象分析与设计

Lecture_01 面向对象概述

主讲：姜宁康 博士

日期：2018/11/7

- 1.4、“面向对象”思想的基本概念
  - 类　　Class
  - 对象　　Object
    - 注：台湾学者把Object翻译成 “物件”

类与对象

- 类和对象，这两者之间的关系
  - 有点像“先有鸡” 还是“先有蛋”的关系一样，有一点纠缠
  - 从他们的定义就可以看出来：
    - 用对象来定义类、用类来产生对象！
  - 我们在学习本课程的时候，可以忽略这个问题
    - 有的时候 在表达概念的时候，对象 与 类可以通用！
  - 在设计软件系统的时候，对象是不存在的，
  - 在软系统运行时，在内存中创建对象。类不存在于物理世界

1、What is a Class、Object?

- *A class* is a description of a set of objects that share the same attributes, operations, relationships, and semantics “共享相同属性、操作、方法、关系或者行为的一组对象的描述符”
  - —*Rumbaugh*
- An object is an *Instance* created from a class. 一个对象是根据一个类创建的一个实例
  - An instance’s *behaviour* and *information structure* is defined in the class. 类,定义了实例的行为和信息结构
  - Its current *state* (values of instance variables) is determined by operations performed on it. 对象的当前状态（实例变量的取值）取决于作用于该对象的操作。

类：
军队、部队、解放军

多个类：
汽车、火车、飞机、交通工具

类：
学生、小学生、少先队员

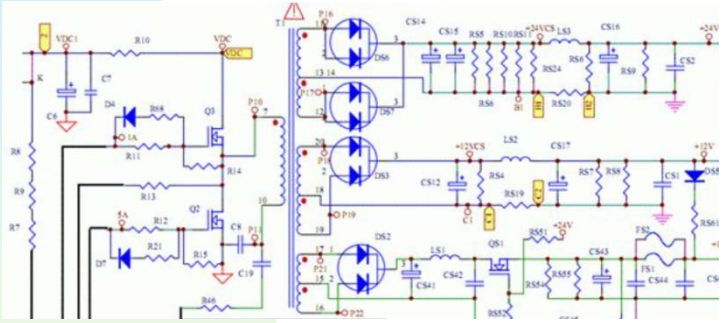
对象：
这是三班的陈刚
这是二排的王健康
这是一连长吴刚

对象：
这是张建伟同学、这是王海三同学
这是吴丽霞同学

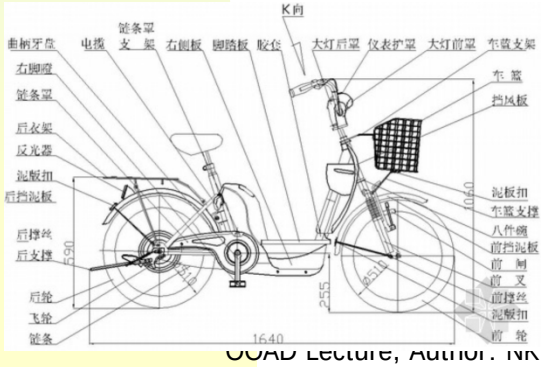
对象：
这是张三的红色小汽车
这是南航的7892号飞机

2、练习：类和对象

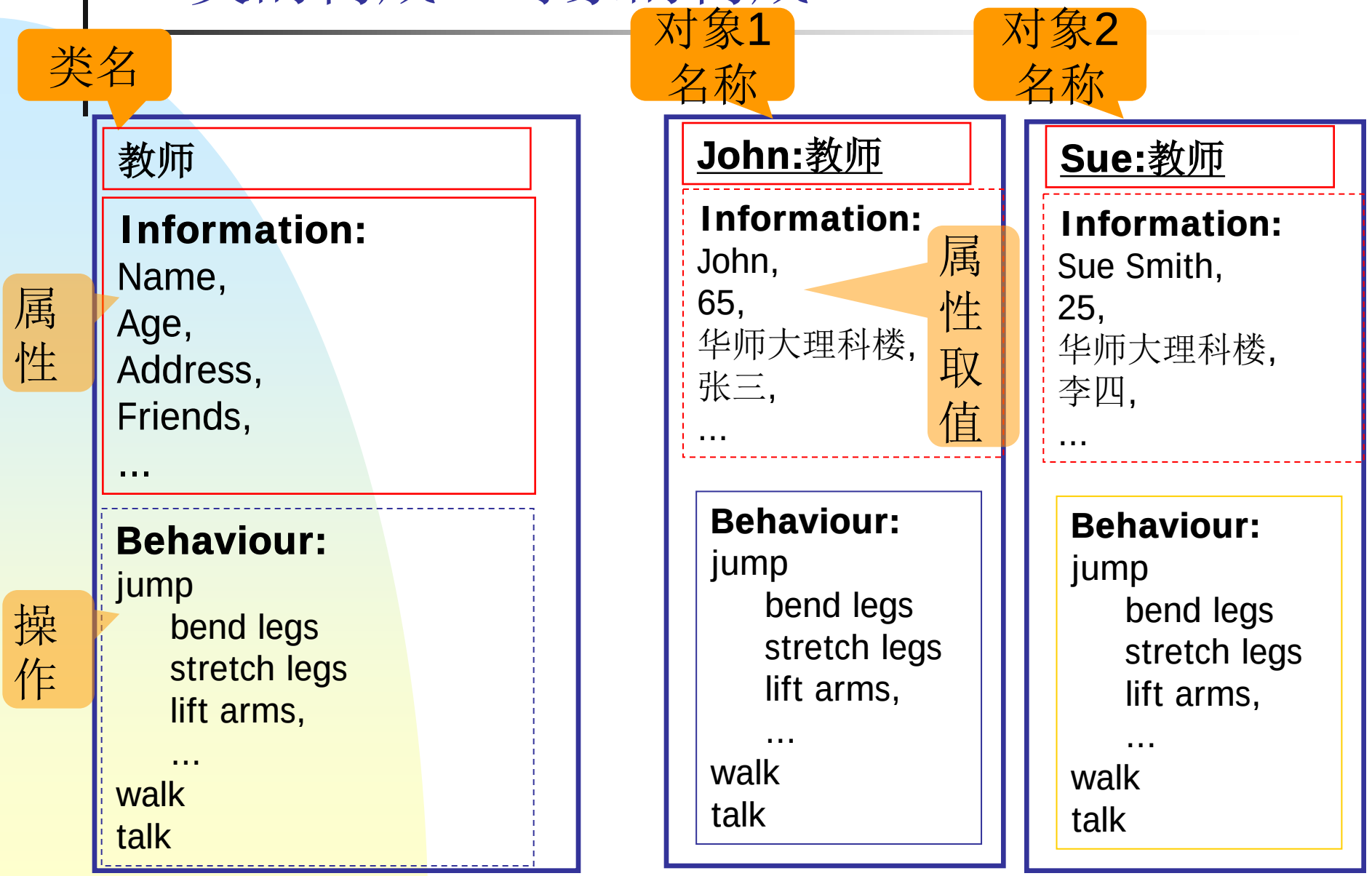
- 比喻
  - 做蛋糕的模板 ---（）；一块块蛋糕 ---（）
  - 电视机原理图 ---（）；一台台电视机 ---（）



  - 工厂生产自行车的蓝图 ---（）
    我的那辆黑色自行车 ---（）



3、类的构成、对象的构成


类名
对象1名称
对象2名称
教师
Information:
Name,
Age,
Address,
Friends,
...
Behaviour:
jump
bend legs
stretch legs
lift arms,
...
walk
talk
属性
操作
John:教师
Information:
John,
65,
华师大理科楼,
张三,
...
Behaviour:
jump
bend legs
stretch legs
lift arms,
...
walk
talk
属性取值
Sue:教师
Information:
Sue Smith,
25,
华师大理科楼,
李四,
...
Behaviour:
jump
bend legs
stretch legs
lift arms,
...
walk
talk

3、类的构成、对象的构成

- 注意：概念之间的互用
  - 属性Attribute == 数据Data == 状态state == 信息information
  - 操作operation == 方法Method ==行为behaviour = = 职责responsibility

4、软件功能是如何完成的？

- 类
  - 定义了对象群体的逻辑结构，包括属性和操作
  - 系统运行时，类作为产生对象的模板，在物理层面是不存在的
- 对象
  - 系统运行时必须为每一个需要的对象分配内存、保存数据
  - 对象存在于物理层面，每个对象都有自己的数据空间（内存）
  - 所有的对象共享同一块代码空间
- 消息
  - 对象之间的一种交流手段
    - 就像我们日常工作中的各种交流手段
- 所有相关对象之间相互协作完成软件功能

5、类和对象小结

- Everything is an object.
- A program is a bunch of objects telling each other what to do by sending messages.
- Each object has its own memory made up of other objects.
- Every object has a type.
  - each object is an instance of a class, in which "class" is synonymous with "type."
- All objects of a particular type can receive the same messages.
  - an object of type "circle" is also an object of type "shape," a circle is guaranteed to accept shape messages. This means you can write code that talks to shapes and automatically handle anything that fits the description of a shape.

- 本讲结束