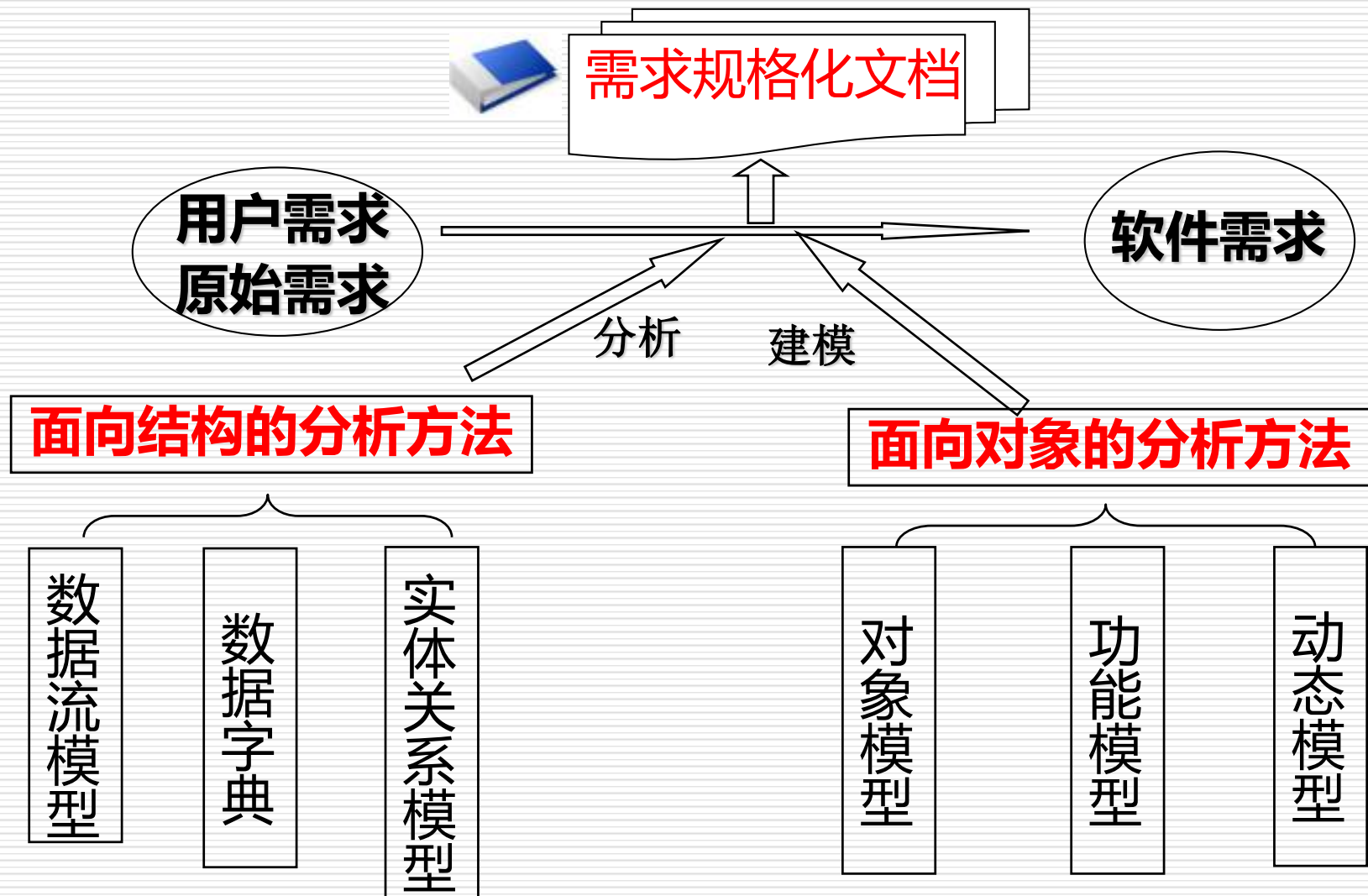


CHAPTER 3

Software Requirement Analysis

Outline



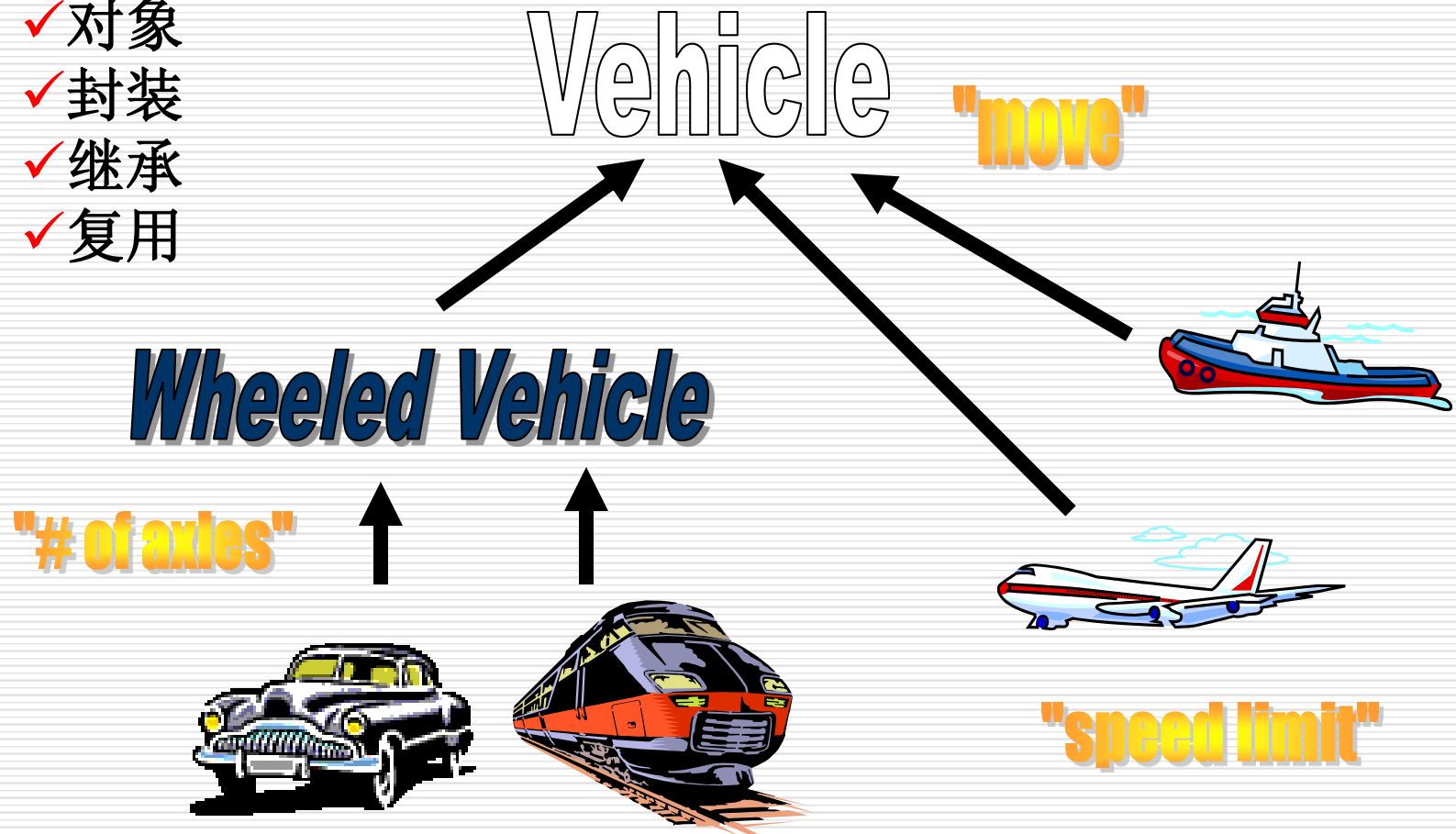
Object-Oriented Analysis: Models

➤ 面向对象分析模型

- ✓ Use-case Diagrams
- ✓ Class and Object Diagrams
- ✓ Behavioral Diagrams

Object-Oriented Concept

- ✓ 对象
- ✓ 封装
- ✓ 继承
- ✓ 复用



思考

- 软件开发中为什么要使用面向对象方法？
- 面向对象分析方法与结构化分析方法有哪些相似之处？有何区别？
- 面向对象方法是对过去的一个完全突破，还是“换汤不换药”？

对象概念

➤ 对象是什么？（Object）

1. 经常看到或听到：一切的事物都是对象。（**广义上**）
2. “对象”一词在现实生活中，表示现实世界的某个具体的事物，即现实世界中的**客观实体**。（**具体地**）
例：汽车，飞机，学生，老师，数据库，文件，表格，窗口，菜单,....
3. 对象就是对具有共同**特性**及**行为**的事物或实体的抽象及概况。（**计算机**）
4. 对象是具有相同**状态**的一组**操作**的集合。（**编程上**）
5. 对象是实体**属性**（数据）和**方法**（服务）的一个封装（模块）（**编程上**）

对象概念

➤ 属性、特性、状态、数据的解释

用来描述客观事物的所具有的属性，状态，特征等的“数据”

例汽车：型号，马力，价格，生产厂商,...

例学生：姓名，学号，年龄，性别，籍贯，身份证，专业,...

例文件：文件名，类型，创建日期，位置，文件大小， ...

➤ 操作，方法、行为，服务，算法，功能的解释

描述客观事物的所具有的行为，操作，服务等“计算”

例汽车：前向行驶，后向行驶，换挡，转弯，刹车...

例学生：选专业，选课，注册报到，查阅考试成绩,...

例文件：打开文件，读文件，写文件，关闭文件， ...

对象概念

➤ 属性、特性、状态、数据的解释

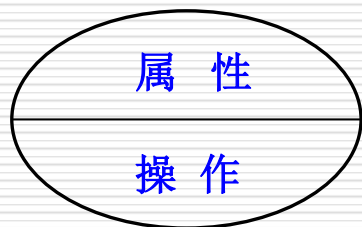
名词，数量词，形容词

➤ 操作，方法、行为，服务，算法，功能的解释

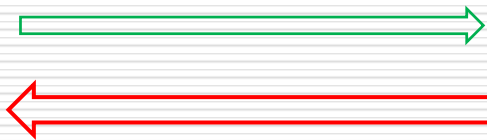
动词，动词性短语

➤ 计算机的再认识

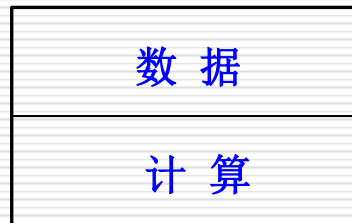
现实中的**实体**



抽象概括



编程中的**对象**



计算机：即对数据实施计算的一种工具

计算机：通过计算和数据来模型客观世界

面向对象编程：即用“对象”来模拟客观世界，编程时特别关心“对象”

对象方法的好处

1. 自然，接近现实

因为用数据和计算模拟客观实体的属性和操作，符合人们描述和讨论事物的习惯，使得程序设计贴近现实，容易普及应用，不要太多计算机专业技巧。

2. 方便建模

因为编程中的对象，与现实生活中的对象基本上是类似的概念，所以可以使用现实生活中对象概念进行抽象，稍作修改来进行建模，并利用业务流程形成算法流程，大大方便了建模和程序设计的过程。

3. 模块化

客观实体是独立存在的对象，形态独立，功能独立，与其它实体和对象容易区分，因此程序模块的边界清晰，不会混淆。对象内部内聚性强，对象与对象之间耦合性弱，这些都是模块化的优点。

对象方法的好处

4. 封装与信息隐藏

任何一个对象的定义（通常称为**类**的定义）包括两部分：
(1) 属性定义，(2) 操作定义。它们封装在一个独立的模块中，同时用**Public** 和 **Private** 说明公用和私用。

```
Define Class 对象的类名称
{
    [Public| Private] 属性变量1
    [Public| Private] 属性变量1
    .....
    [Public| Private] 方法函数1(...)
    [Public| Private] 方法函数2(...)
    .....
}
```

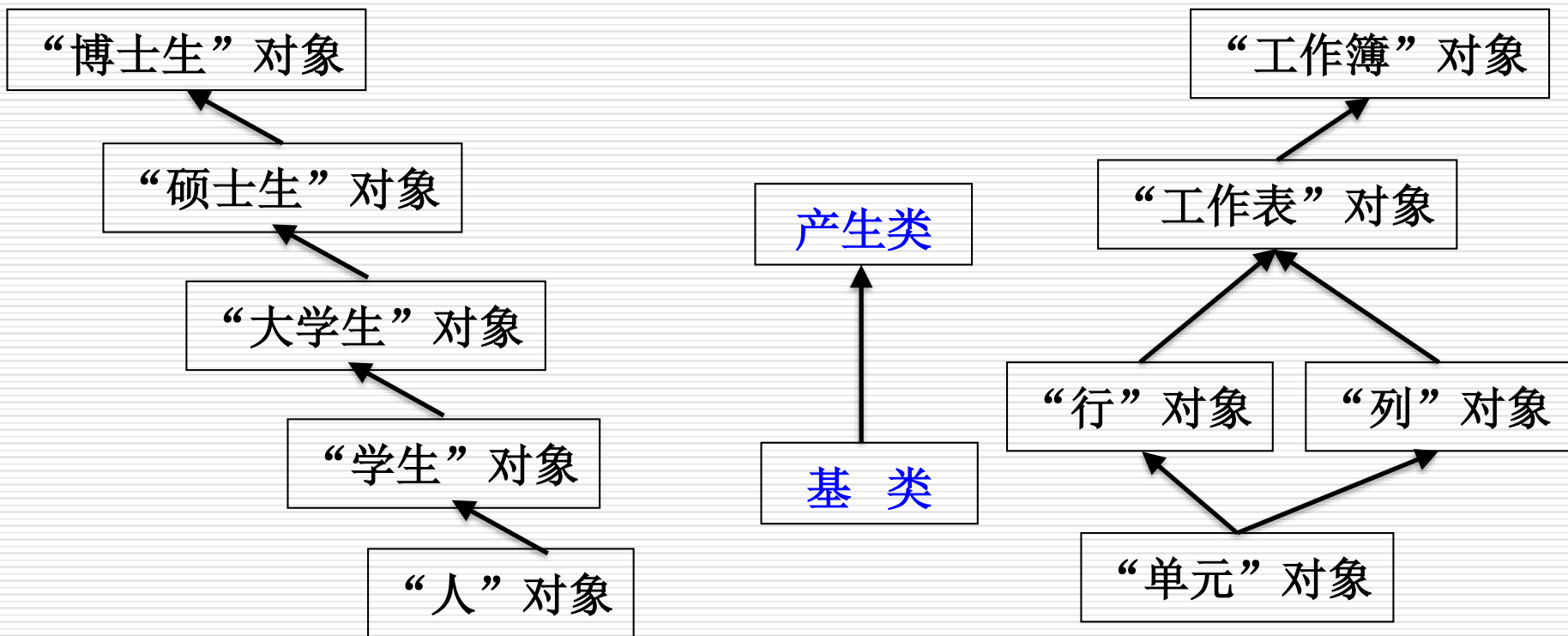
```
Class Student
{
    Public sNo as Integer
    Private sName as String
    Private hAdd as String
    Privave Function register(...)
    Public Function Vote(...)
    .....
}
```

对象方法的好处

5. 继承复用（重用）

多个对象（或类）一般具有层次，父子，上下，等继承关系。

例：人，学生，大学生，硕士生，博士生



对象方法的好处

- ✓ “子女”对象继承“父母”对象的属性和方法;
- ✓ “高层”对象继承“低层”对象的属性和方法;
- ✓ 高层对象可以复用低层对象的程序代码;
- ✓ 后面要写的程序可以复用以前已有的程序;
- ✓ 今后的程序设计方案可以复用以前的经验和积累;
- ✓ 未来的工作发展利用已有的文明和基础;
- ✓ 对象程序开发和设计--符合人类认识和改造世界的过程。

对象方法的好处

6. 扩展修改维护容易

由于以上特点：自然，容易理解，模块独立性强，信息封装和隐藏，可以继承复用，所以面向对象编程扩展修改维护容易。



C++, JAVA 的对象是什么？

- ✓ 是一段“程序”，如同“过程”一样。
- ✓ 过程描述一个特定或固定功能的程序模块。
- ✓ “对象（类）”是描述一个事物（实体）的程序模块，包括“属性变量”的描述，以及“操作函数”的描述。

Class Student

Student oneStudent

//学生对象

Public sNo as Integer

Private sName as String

Private hAdd as String

...

Privave Function register(...)

...

End Function

Public Function Vote(...)

...

End Function

.....

} 属性描述

} 操作描述

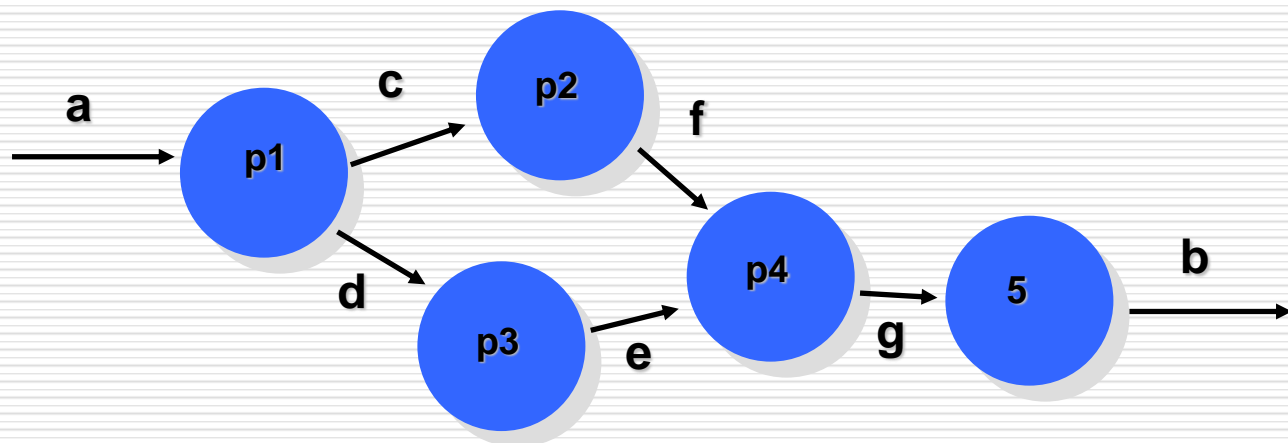
“对象” 和 “类”

➤ 类（Class），对象(Object)

造汽车的**图纸**----**类**
生产出的**汽车**--**对象**

- ✓ 类是对象定义，描述，设计时的称呼。
- ✓ 对象是类使用，调用，实例化时的称呼。类对应的程序运行使用的称呼
- ✓ 由于内置类库，用户不要定义和设计，直接调用和使用，因此常称对象。
- ✓ 总之，类是描述和抽象；对象是具体和实例化（instance）
- ✓ 以前熟悉和相似的例子：数据类型，如 整型 `int` 是类
`int myValue=123` ’ 123 则是整型类 `Integer` 的一个实例
`string myName=“Flora”` ’ 是字符串类型的一个实例
- ✓ 谈定义的话，则是定义类，即描述和设计类；之后使用，则是称呼对象。
- ✓ VBA开发环境，提供插入类模块的功能，即是自定义类模块，等待之后使用。

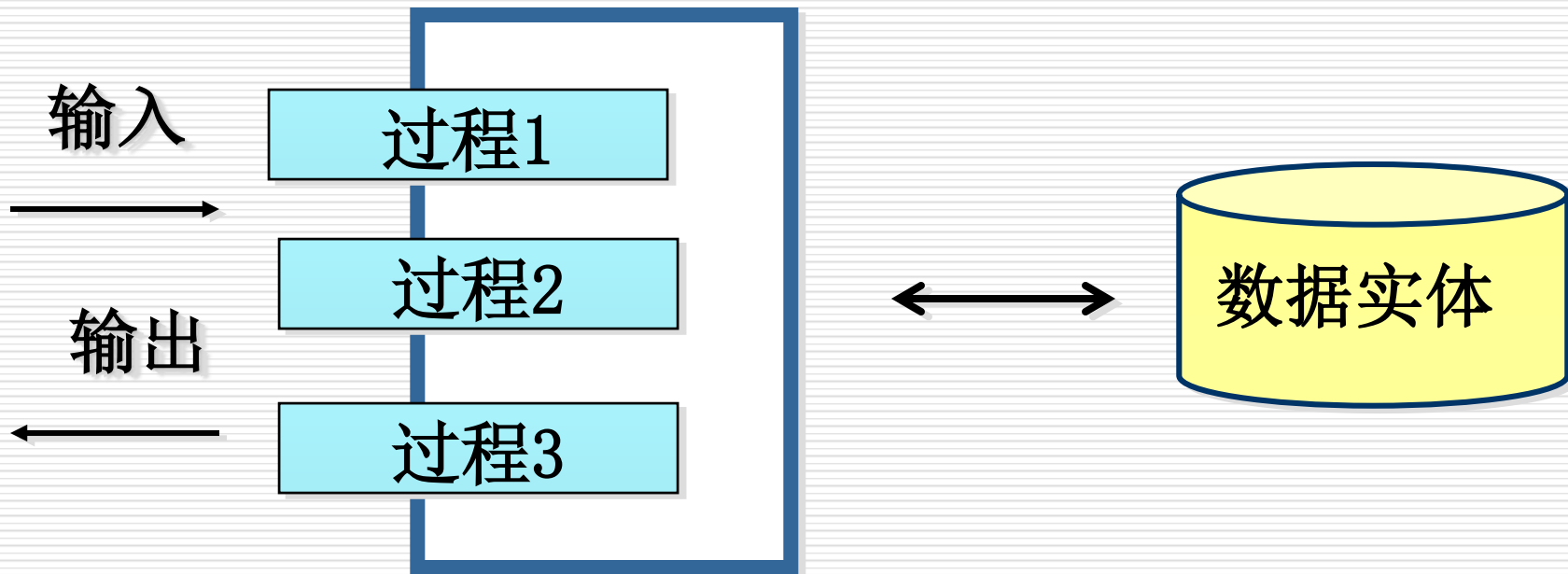
面向结构和面向对象方法的比较



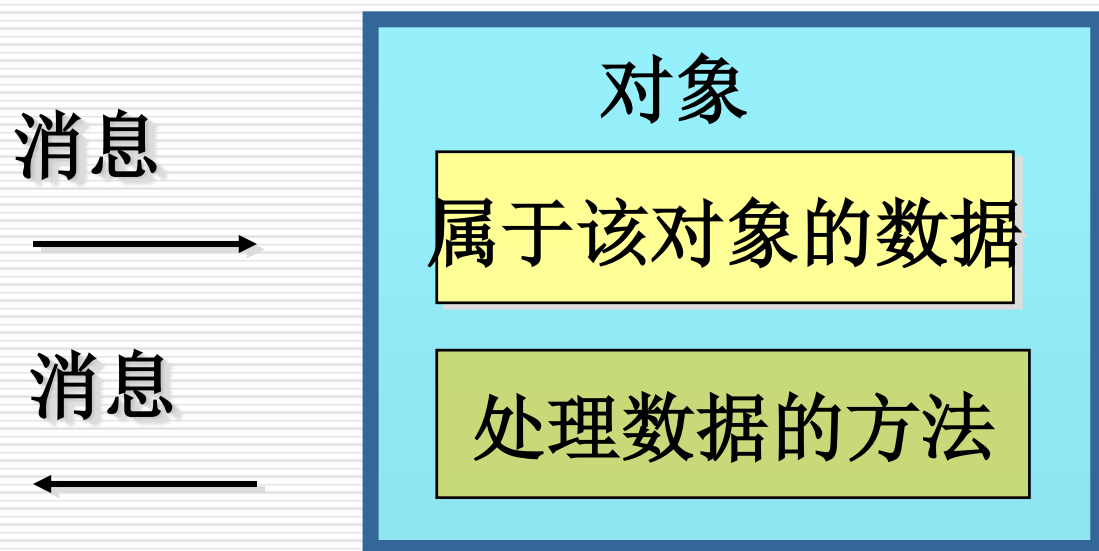
过程的集合

实体的集合





传统方法: 数据与过程是分离的



对象方法: 数据和处理数据的方法封装成一个单元

面向结构和面向对象方法的比较

SOA方法：

面向功能, 把系统看成一组功能（过程）的集合

OOA方法：

把问题当作一组相互作用的实体，并确定实体间关系

面向结构和面向对象方法的比较

传统方法

软件系统是过程的集合
过程与数据实体交互
过程接受输入并产生输出



面向对象方法

系统是交互对象的集合
对象与人或其它对象交互
对象发送与响应消息

编程哲学、思想上不同

传统方法和面向对象方法的比较

SOA方法：

入门，基础的方法

OOA方法：

后续，先进，容易普及的方法

两种方法不排斥，OOA方法中的操作、行为的分析和描述，可以SOA方法

OOA目的或任务

分析和定义所有和被求解的问题相关的类及同类关联的关系和行为），为了达到这个目标，必须完成以下任务：

- （1）必须在客户和软件工程师之间沟通了解基本的用户需求；
- （2）必须标识类（定义属性和方法）；
- （3）必须刻划类层次；
- （4）表示对象/对象关系（对象连接）；
- （5）必须建模对象行为；
- （6）任务(1)到(5)递进地反复使用，直至分析清楚，完成建模

OOA建立分析模型5个基本原则

面向对象分析方法使得软件工程师能够通过对象、属性和操作（作为主要的建模成分）的表示来对问题建模。

建立分析模型5个基本原则：

- (1) 建模信息域；
- (2) 描述模块功能；
- (3) 表示模型行为；
- (4) 分解以模型显示更多细节；
- (5) 早期模型表示问题的本质，而后期模型提供实现细节。

流行的几种面向对象分析方法

- ✓ **Booch方法**
- ✓ **Coad-Yourdon方法**
- ✓ **Rumbaugh 方法(简称OMT)**
(Object Modeling Technology)
- ✓ **Jacobson 方法 (简称OOSE)**
- ✓ **由Rumbaugh 、 Booch 、 Jacobson**
提出的统一建模语言
(Unify Modeing Language简称UML)

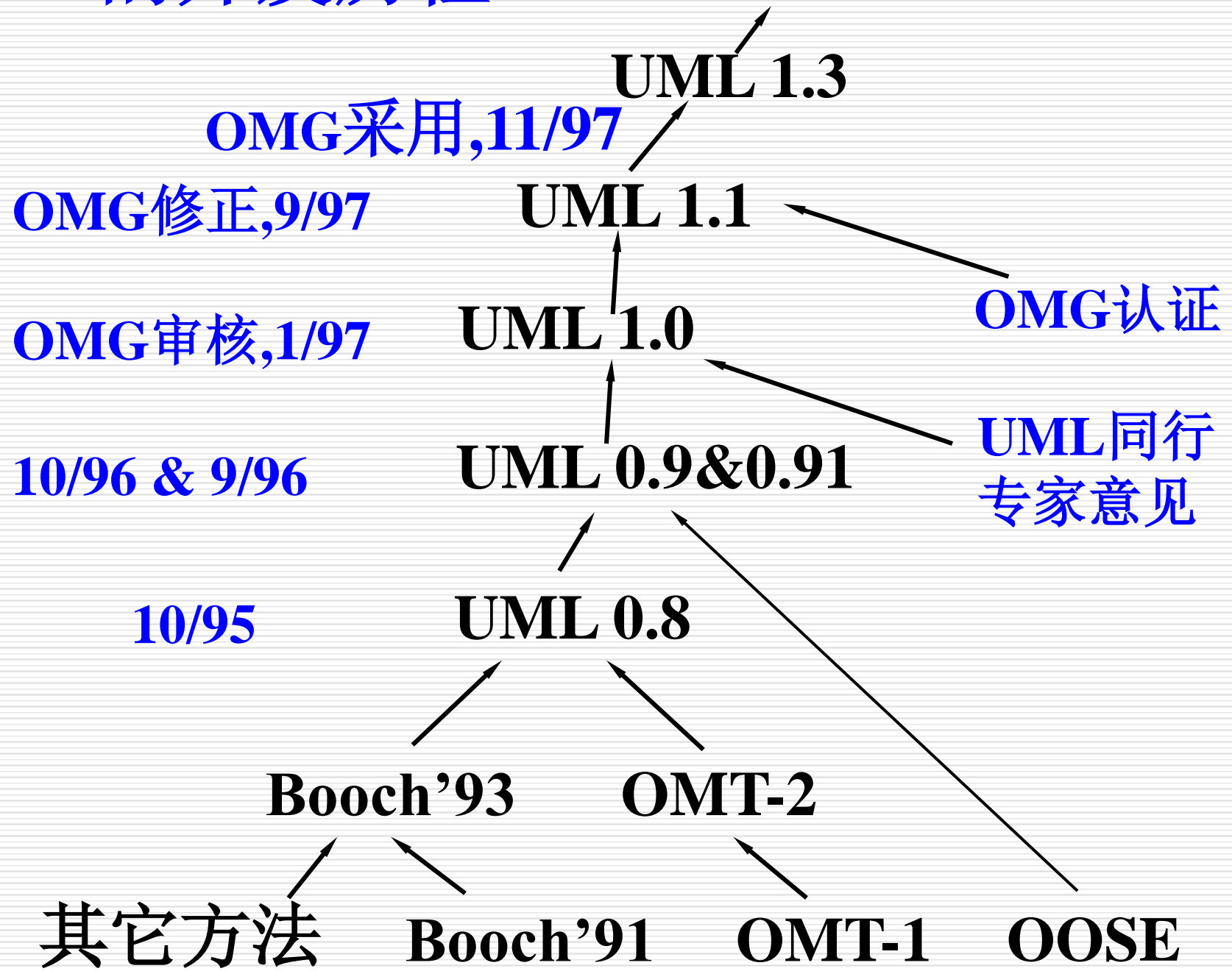
UML

Unified Modeling Language

统一建模语言

通过画图进行软件分析

UML的开发历程



UML的用意

UML是非常有用的一种图，在需求分析中，可以让人们从繁重的文档中解脱出来，并且促使人们在做需求时能够更加准确、直观的表现自己的意思。常用的语言文字往往是不能将一种事物表达得清晰，这时候就需要用其它的方式来进行表达，UML就是这种方法。

UML能为我们做什么

- ❑ UML可以做软件需求分析
- ❑ UML可以做软件开发设计
- ❑ UML可以做系统部署设计
- ❑ UML也适用非软件领域的系统建模，如企业机构或业务过程，以及处理复杂数据的信息系统、具有实时要求的工业系统或工业过程等。

What is UML?

- ❑ UML stands for Unified Modeling Language
- ❑ The UML combines the best from
 - ✓ **Data modeling concepts (entity relationship diagrams)**
 - ✓ **Business modeling (work flow diagrams)**
 - ✓ **Object modeling**
 - ✓ **Component modeling**

What is UML?

- ❑ The UML is a **language standard** for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system
- ❑ It can be used with all processes, throughout the development life cycle, and across different implementation technologies

UML的定义

UML（统一建模语言，Unified Modeling Language）是一种定义良好、易于表达、功能强大且普遍适用的可视化建模语言。它融入了软件工程领域的新思想、新方法和新技术。它的作用域不限于支持面向对象的分析与设计，还支持从需求分析开始的软件开发的全过程。在系统分析阶段，我们一般用UML来画很多图，主要包括用例图、状态图、类图、活动图、序列图、协作图、构建图、配置图等等，要画哪些图要根据具体情况而定。其实简单的理解，也是个人的理解，UML的作用就是用很多图从静态和动态方面来全面描述我们将要开发的系统。

UML的9种图

	图名称	图定义	图性质
1	类图	一组类、接口、协作及它们的关系	静态图
2	对象图	一组对象及它们的关系	静态图
3	用例图	一组用例、参与者及它们的关系	静态图
4	顺序图	一个交互，强调消息的时间顺序	动态图
5	协作图	一个交互，强调消息发送和接受的对象的结构组织	动态图
6	状态图	一个状态机，强调对象按事件排序的行为	动态图
7	活动图	一个状态机，强调从活动到活动的流动	动态图
8	构件图	一组构件及关系	静态图
9	配置图 (实施图)	一组接点及它们的关系	静态图

包图: 包中的类以及包与包之间的关系(静态图)

UML图的画法

1. 参考资料
2. 到网上搜索参考文献
3. Rational Rose 软件练习
4. www.umlchina.com
5. www.uml.org.com
6. www.rational.com
7. www.uml.net.cn