第4章：面向对象方法概述

本章内容

- 传统开发方法中存在的问题
- 面向对象的基本思想
- 面向对象的主要概念及基本原则
- 面向对象方法的主要优点
- 面向对象方法的发展史及现状简介
- 什么是UML
- UML建模工具

本章内容

- 传统开发方法中存在的问题
- 面向对象的基本思想
- 面向对象的主要概念及基本原则
- 面向对象方法的主要优点
- 面向对象方法的发展史及现状简介
- 什么是UML
- UML建模工具

传统开发方法中存在的问题

- 在二十世纪六十年代以前
- 软件系统都是较小且相对简单的
- 所用的编程语言都是十分简单（**Fortran, Cobol等**）
- 时兴个人英雄主义，即崇尚程序员的个人技能
- 代码是面条式的，特别是代码中含有**GOTO**语句

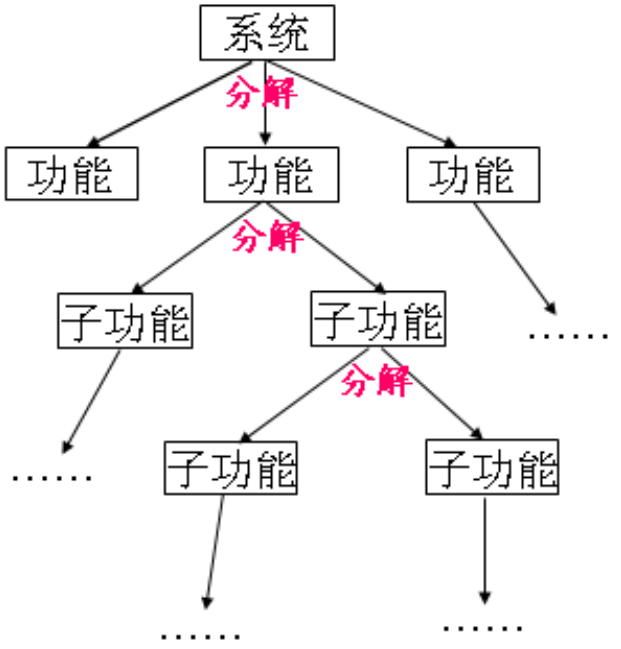
随着软件复杂性的增长，随心所欲的方法就不再是可接受的了，因为这样的代码是很难维护的。

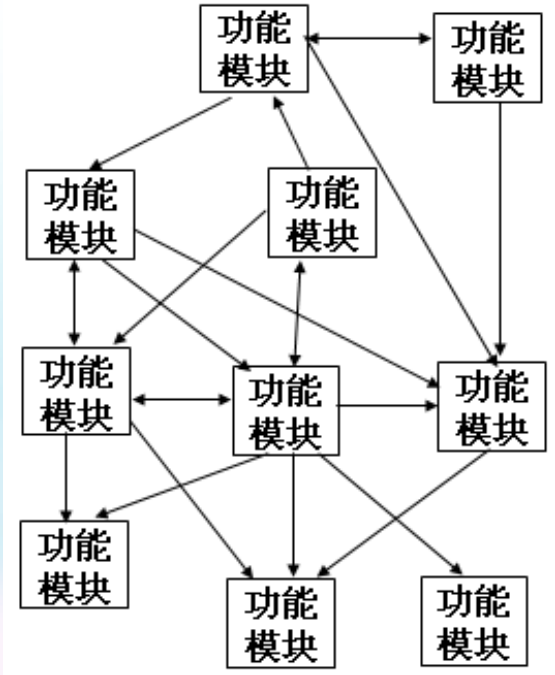
高级编程语言的引入有助于解决一些与复杂性有关的问题，但这些语言并不是充分的。

那时，无开发方法而言

功能分解法（起于二十世纪七十年代）

功能分解＝ 功能+子功能+功能接口



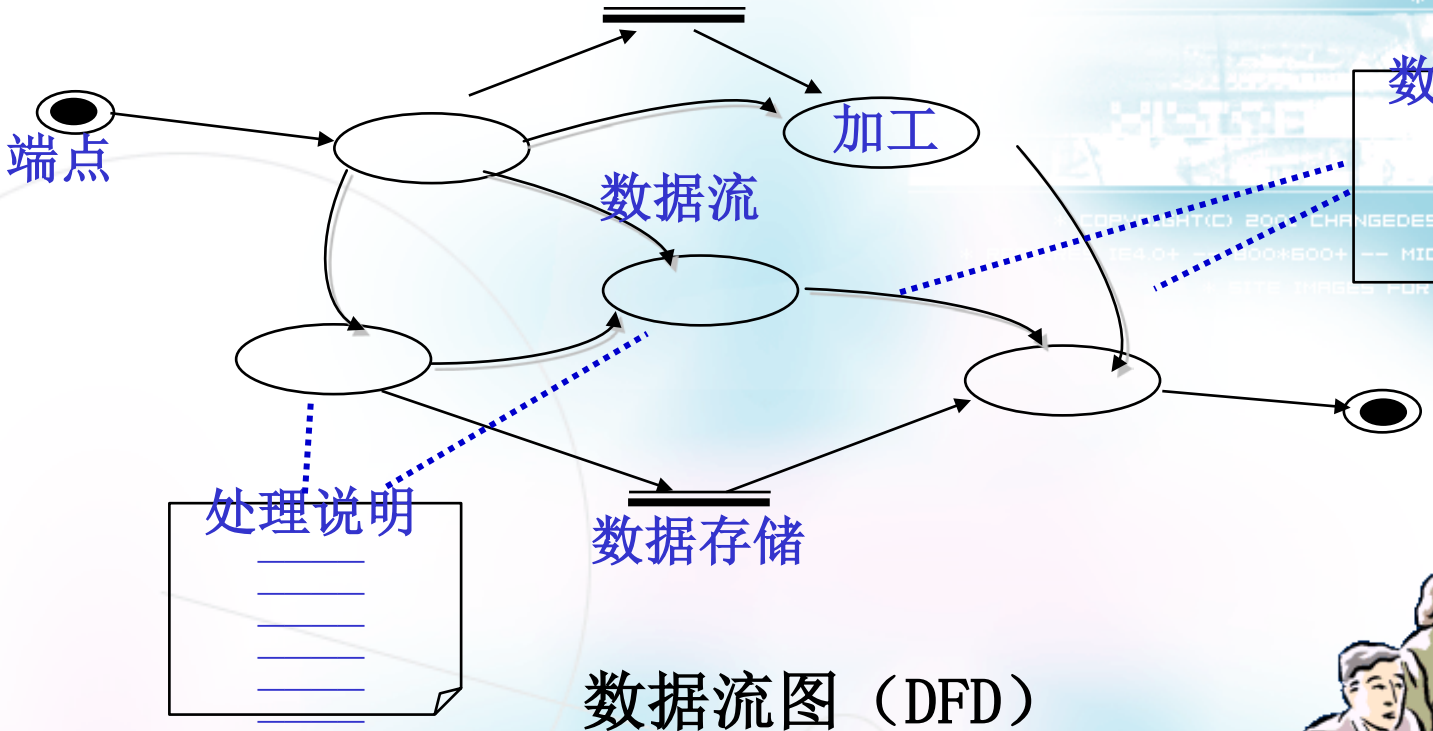


工作过程：
一层层地进行功能分解

得到的系统模型：
由模块及其接口构成

数据流法（结构化分析法）

数据流法＝ 数据流＋数据处理（加工）＋数据存储＋端点
＋处理说明＋数据字典



数据流图（DFD）

基于数据的方法

- 实体-联系图：用实体的数据集合作为构造块，以数据结构为中心。
- 当时软件职业者和研究人员认为数据是“企业应用”中最稳定的部分，并且关系数据库有一个极好的数学基础，在二十世纪八十年代大多数公司使用数据建模方法开发软件。
- 结构化的方法实际上帮助开发者处理数据，但数据建模方法却不能帮助开发者管理功能。

总结

上述方法都仅基于一个或两个角度看待系统，例如，

- 功能分解法集中于将功能作为系统的构造块
- 数据流法（结构化分析法）：数据流＋加工
- 在数据分析方法中（实体联系模型）构造块是实体，但在该方法中用来满足系统需求的功能被完全忽略掉了。

对OO产生都做出了一定的贡献。

本章内容

- 传统开发方法中存在的问题
- **面向对象的基本思想**
- 面向对象的主要概念及基本原则
- 面向对象方法的主要优点
- 面向对象方法的发展史及现状简介
- 什么是UML
- UML建模工具

面向对象的基本思想

面向对象方法的解决问题的思路是从现实世界中的客观对象（如人和事物）入手，尽量运用人类的自然思维方式来构造软件系统，这与传统的结构化方法从功能入手和信息建模方法从信息入手是不一样的。在面向对象方法中，把一切都看成是对象。

面向对象开发方法举例

不使用面向对象描述一件事

青椒炒牛肉

- 取出青椒 500 g，用刀切成细丝，过油
- 取出牛肉 300 g，切丁，用酱油、酒、黑醋腌制 30 分钟
- 起油锅、放入牛肉炒及青椒大火快炒 1 分半
- 拿出太白粉、水调在一起，这个称为芡汁
- 將芡汁倒入锅中搅拌，会产生粘稠现象這叫芶芡
- 完成。

使用面向对象描述一件事

青椒炒牛肉

- 青椒
  数量：500 g
  处理：用刀切成细丝，过油
- 牛肉
  数量：300 g
  处理：切丁，用酱油、酒、黑醋腌制
- 芡汁
  制作：太白粉调上适量的水
  芶芡：將芡汁倒入锅中
- 青椒处理好、牛肉处理好、芡汁制作好，放入锅中快炒 1 分半后用芡汁芶芡即可。

青椒炒牛肉

青椒
数量：500 g
处理：用刀切成细丝，过油

牛肉
数量：300 g
处理：切丁，用酱油、酒、黑醋腌制

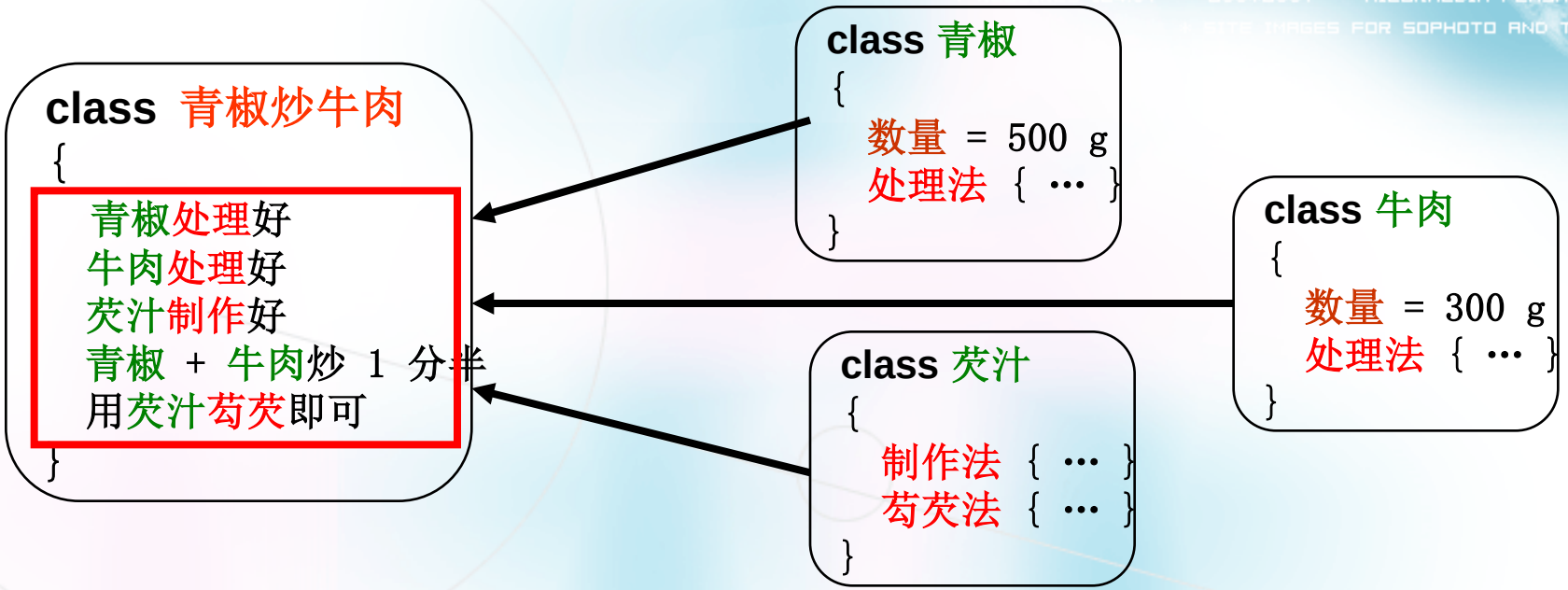
芡汁
制作：太白粉调上适量的水
芶芡：将芡汁倒入锅中

青椒处理好、牛肉处理好、芡汁制作好，
放入锅中快炒 1 分半后用芡汁芶芡即可。

先定义好参与这件事的物品有哪些
(这些物品叫做 “对象”)

再定义这些物品所需的数量及行为
(这些数量及行为叫做
“属性”与“操作”)

接下来就可用物品间的互动行为
来描述整件事情是如何发生的。
(这部分的描述称为“程序”)



面向对象的基本思想

- 从现实世界中客观存在的事物出发来建立软件系统，强调直接以问题域（现实世界）中的事物为中心来思考问题、认识问题，并根据这些事物的本质特征，把它们抽象地表示为系统中的对象，作为系统的基本构成单位。（对象）

- 用对象的属性表示事物的性质；用对象的操作表示事物的行为。（属性与操作）

- 对象的属性与操作结合为一体，成为一个独立的、不可分的实体，对外屏蔽其内部细节。（对象的封装）

- 对事物进行分类。把具有相同属性和相同操作的对象归为一类，类是这些对象的抽象描述，每个对象是它的类的一个实例。（类）

面向对象的基本思想

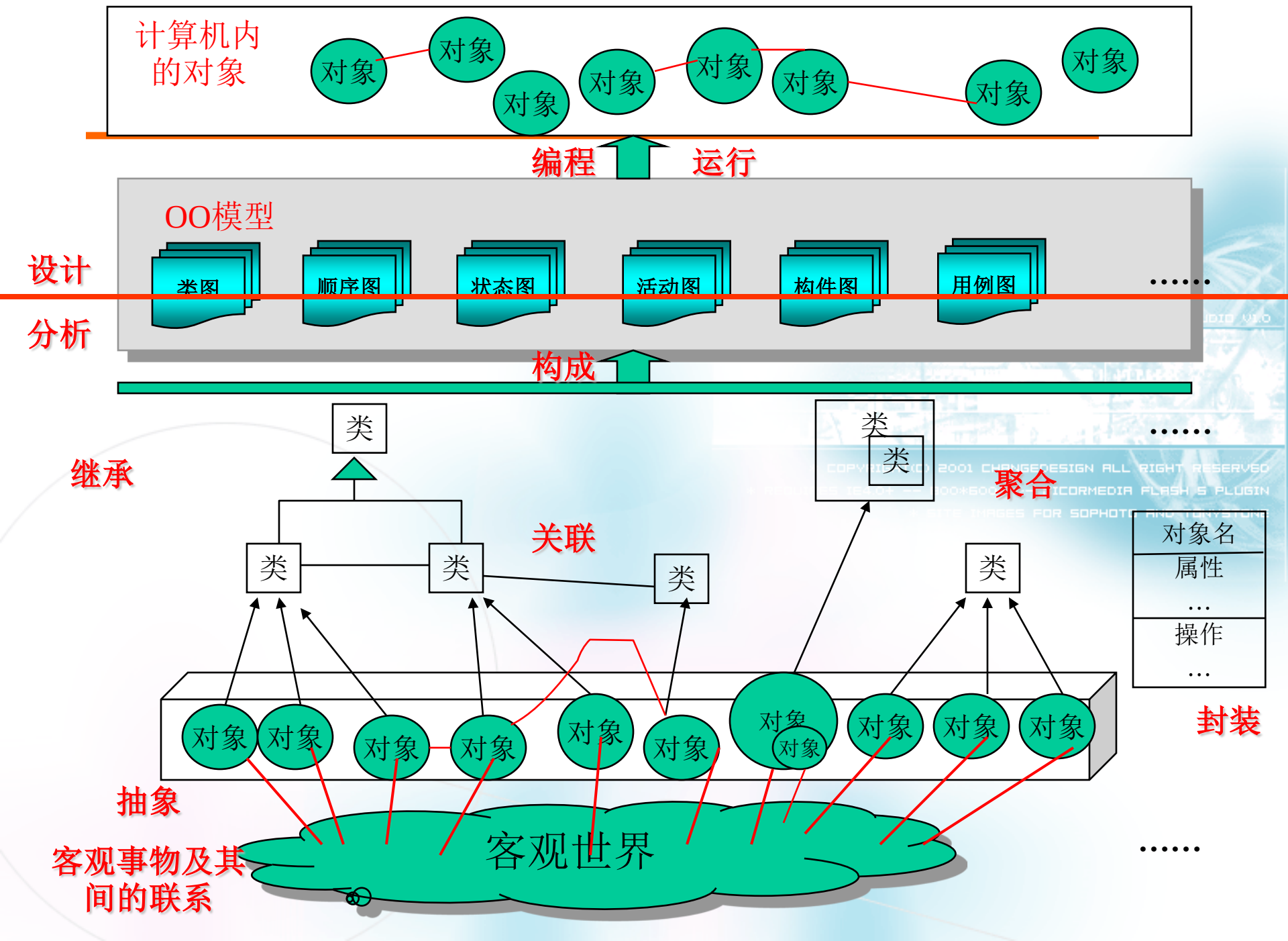
- 复杂的对象可以用简单的对象作为其构成部分。（聚合）

- 通过在不同程度上运用抽象的原则，可以得到较一般的类和较特殊的类。特殊类继承一般类的属性与操作，从而简化系统的构造过程及其文档。（继承）

- 类具有封闭性，把内部的属性和操作隐藏起来，只有公共的操作对外是可见的。（类的封闭性）

- 对象之间通过消息进行通讯，以实现对象之间的动态联系。（消息）

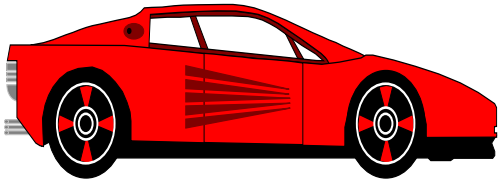
- 通过关联表达类(一组对象)之间的静态关系。（关联）


计算机内的对象
对象
对象
对象
对象
对象
对象
对象
对象
编程
运行
OO模型
设计
分析
类图
顺序图
状态图
活动图
构件图
用例图
……
构成
继承
类
类
类
类
类
类
关联
聚合
对象名
属性
…
操作
…
封装
对象
抽象
客观世界
客观事物及其间的联系
……

本章内容

- 传统开发方法中存在的问题
- 面向对象的基本思想
- **面向对象的主要概念及基本原则**
- 面向对象方法的主要优点
- 面向对象方法的发展史及现状简介
- 什么是UML
- UML建模工具

面向对象的主要概念：对象、属性、操作、对象标识



对象是现实世界中某个实际存在的事物，它可以是有形的（比如一辆汽车），也可以是无形的（比如一项计划）。对象是构成世界的一个独立单位。它具有自己的静态特征和动态特征。

对象是系统中用来描述客观事物的一个实体，它是构成系统的一个基本单位。一个对象由一组属性和对这组属性进行操作的一组操作构成。

属性是用来描述对象性质的一个数据项。

操作是用来描述对象行为的一个动作序列。

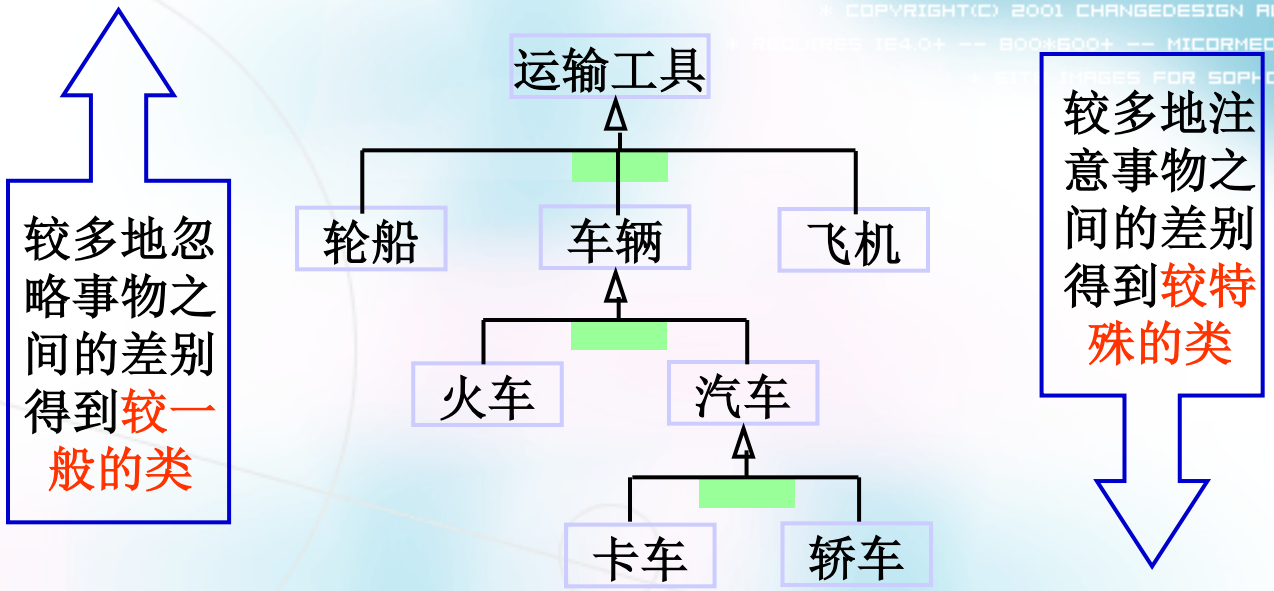
对象标识就是对象的名字。

面向对象的主要概念：类、一般类、特殊类、抽象

抽象(化)
忽略事物的非本质特征，只注意那些与当前目标有关的本质特征，从而找出事物的共性。

类是具有相同属性和操作的一组对象的集合，它为属于该类的全部对象提供了统一的抽象描述，其内部包括属性和操作两个主要部分。类的作用是用来创建对象，对象是类的一个实例。

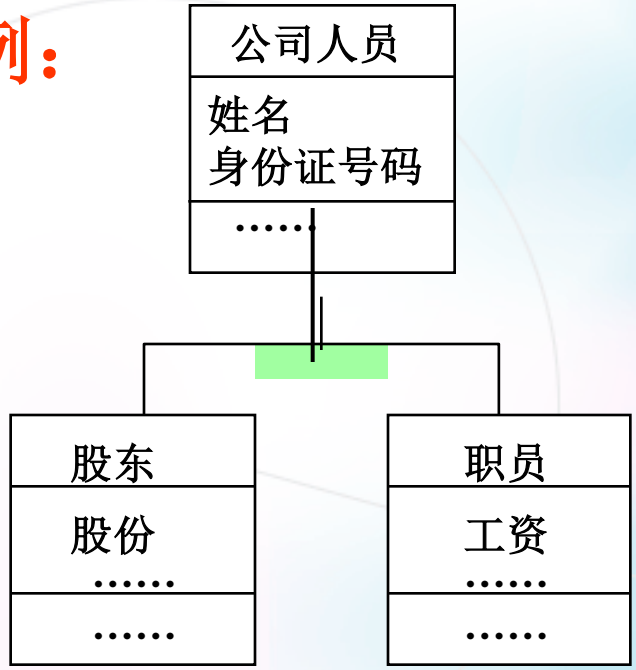
不同程度的抽象可得到不同层次的分类



面向对象的主要概念：继承

特殊类（子类）拥有其一般类（父类）的全部属性与操作，称作特殊类对一般类的继承。继承意味着自动地拥有，子类从父类中继承属性和操作，根据需要添加自己的属性和操作。

例：



继承简化了人们对事物的认识和描述，非常有益于软件复用，是OO技术提高软件开发效率的重要原因之一。

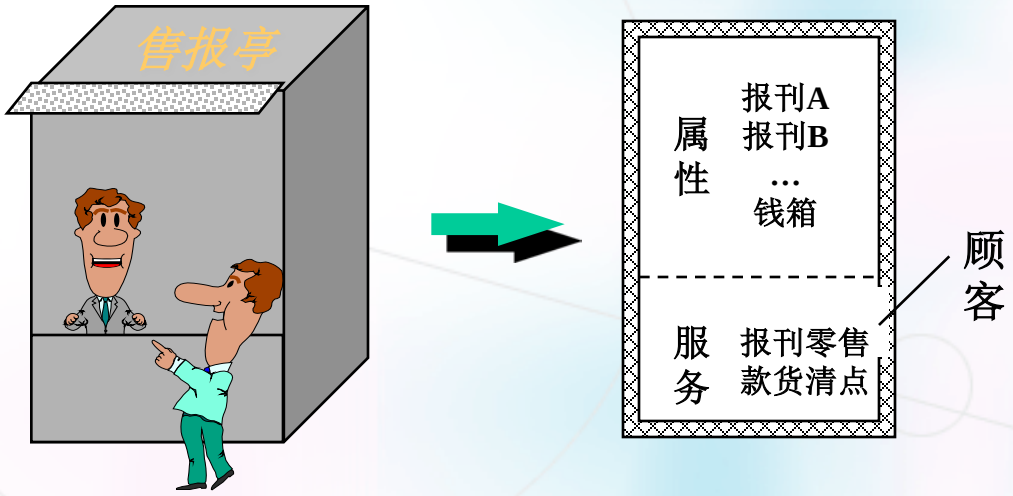
面向对象中的基本原则

（1）抽象

从事物中舍弃个别的非本质的特征，而抽取共同的、本质特征的做法叫抽象。

（2）封装

把对象的属性和操作结合成一个独立的系统单位，并尽可能隐蔽对象的内部细节。只是向外部提供接口，降低了对象间的耦合度。



由封装机制保证：数据不能被对象的使用者直接访问。只允许通过由对象提供的方法或代码访问数据。

面向对象中的基本原则

（3）继承

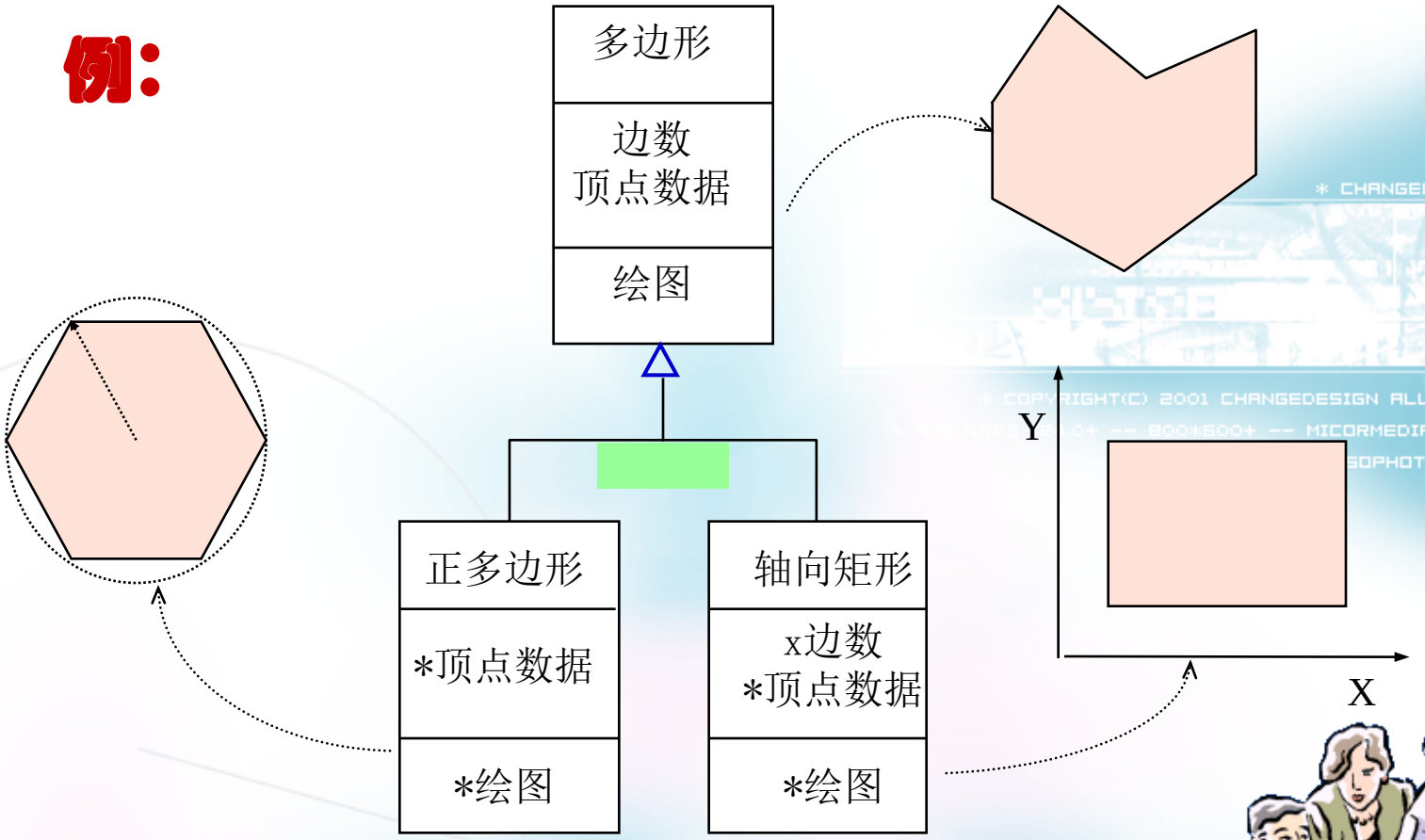
特殊类拥有其一般类的全部属性与操作，称作特殊类对一般类的继承。

（4）多态

指一般类和特殊类有相同格式的属性和操作，但这些属性和操作有不同的含义，即表现出不同的数据类型和行为。这样，针对同一消息，不同对象可对其响应，但体现出来的行为是不同的。

多态举例

例:



多变形: x

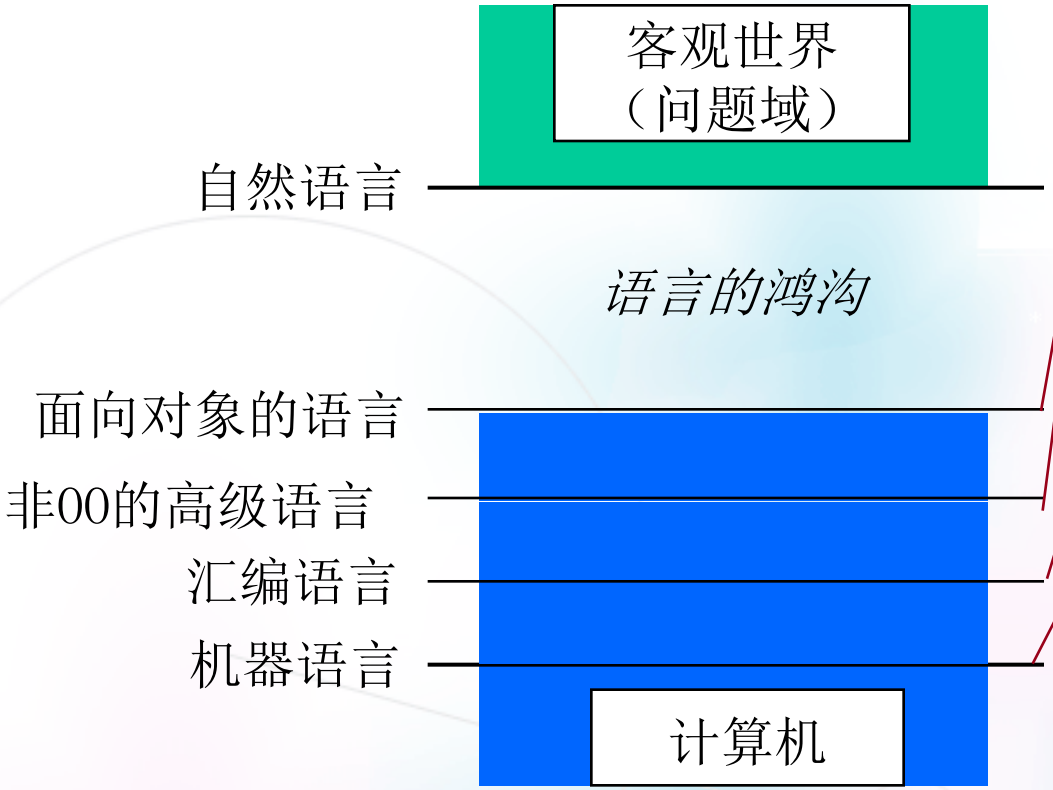
x.绘图

本章内容

- 传统开发方法中存在的问题
- 面向对象的基本思想
- 面向对象的主要概念及基本原则
- **面向对象方法的主要优点**
- 面向对象方法的发展史及现状简介
- 什么是UML
- UML建模工具

面向对象方法的主要优点

1.从客观世界到计算机语言的鸿沟变窄



能比较直接地反映客观世界的本来面目，并使软件开发人员能够运用人类认识事物所采用的一般思维方法来进行软件开发。

具体事物。特别是**结构化编程语言**更便于体现客观事物的结构和逻辑涵义，与人类的自然语言更接近，但仍有不少差距。

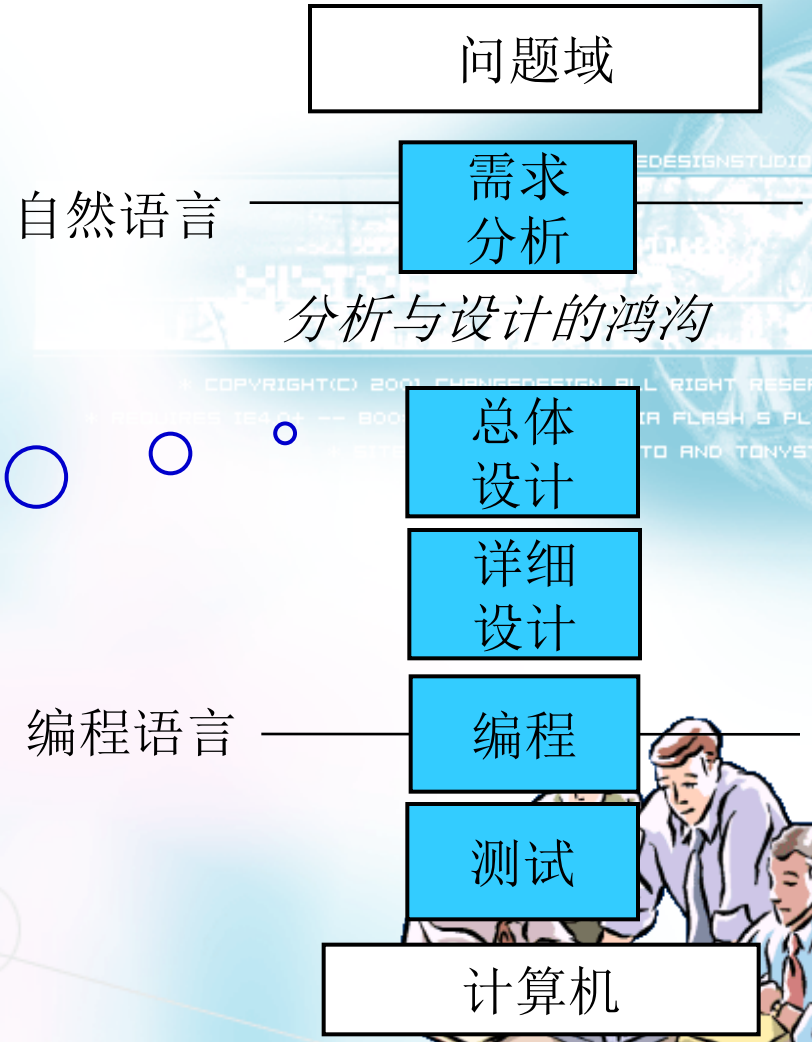
，仍需考虑大量的机器细节。

类的思维最远。

面向对象方法的主要优点

2.面向对象方法使得从问题域到计算机间的鸿沟变窄

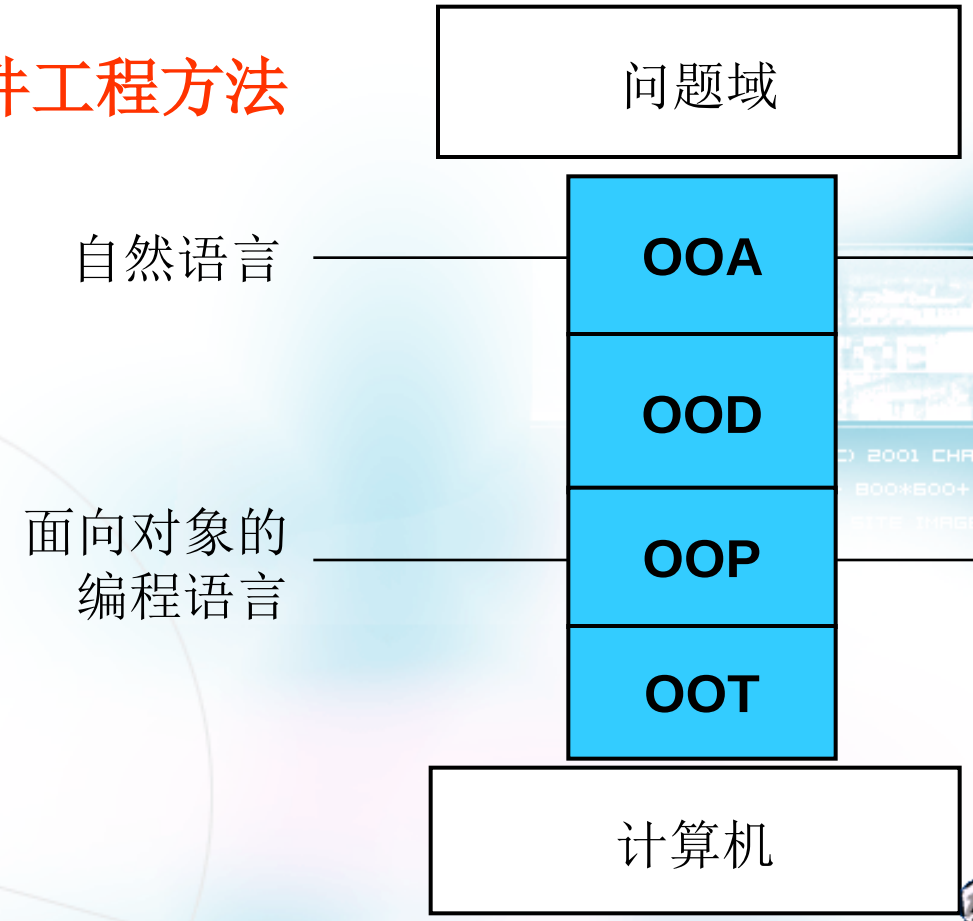
结构化的软件工程方法



分析与设计概念及表示法的不一致

面向对象方法的主要优点

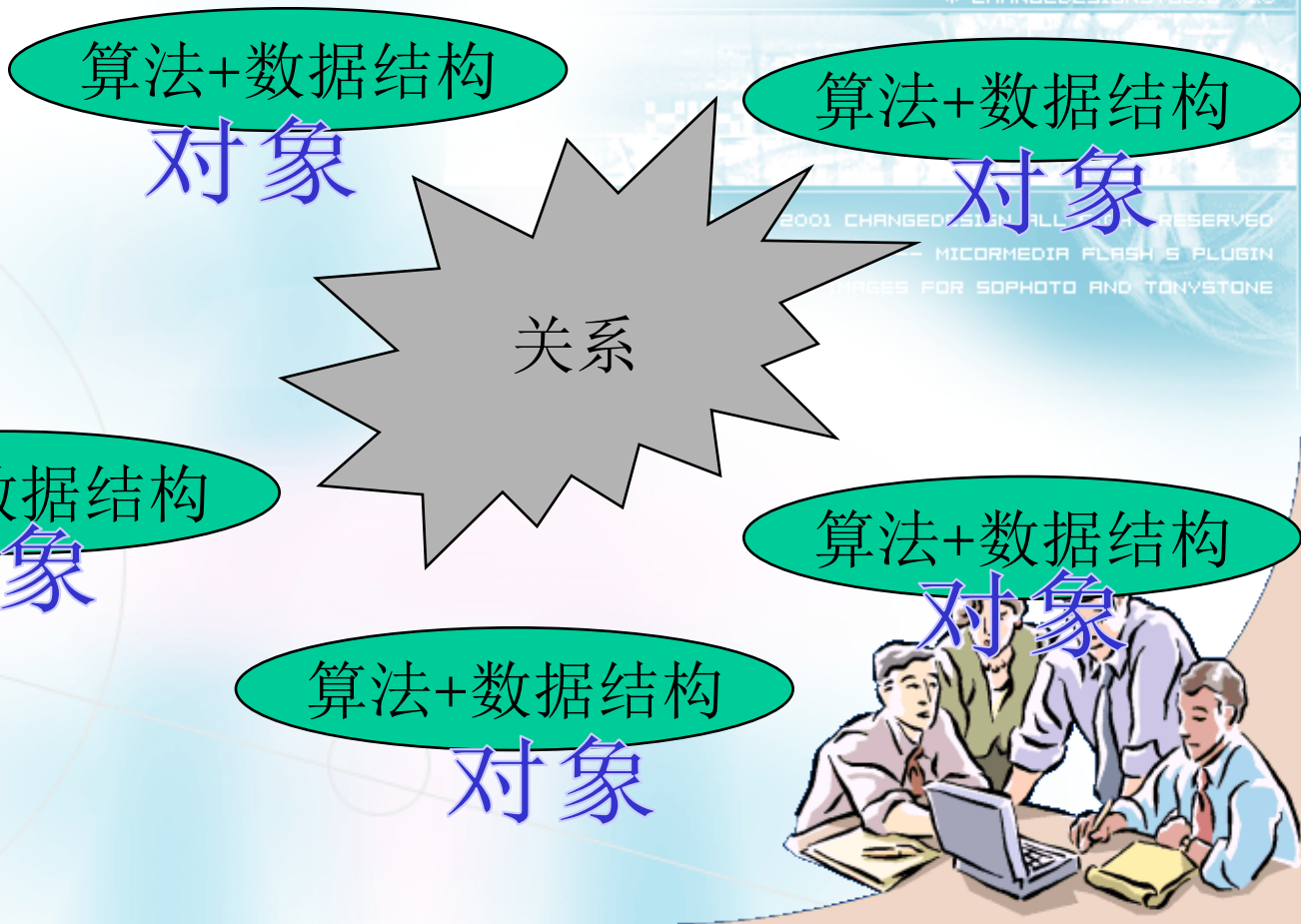
面向对象的软件工程方法



面向对象方法的主要优点

3. 面向对象方法有助于软件的维护与复用

面向对象程序设计把数据和处理数据的过程作为一个整体，即对象。



面向对象方法的主要优点

4. 面向对象方法有助于提高软件的质量和生产率

有数据表明，使用OO技术从分析到编程阶段能提高效率20%，在维护阶段（完善性维护、纠错性维护、适应性维护、预防性维护）提高得就更多。这主要体现在如下几方面:

- 设计的投入在编程、测试时会得到回报
- OO方法使系统更易于理解
- 分析文档、设计文档、源代码对应良好
- 功能变化引起的全局性修改较少
- 有利于OOA结果的复用

本章内容

- 传统开发方法中存在的问题
- 面向对象的基本思想
- 面向对象的主要概念及基本原则
- 面向对象方法的主要优点
- **面向对象方法的发展史及现状简介**
- 什么是UML
- UML建模工具

面向对象方法的发展史

1、 雏形阶段

- 60年代挪威计算中心开发的Simula67—面向对象语言的先驱和第一个里程碑(首先引入了类的概念和继承机制)。
- 70年代CLU、并发Pascal、Ada和Modula-2等语言对抽象数据类型理论的发展起到重要作用(支持数据与操作封装)。
- 犹他大学博士生Alan Kay设计了一个实验性语言Flex。从Simula 67中借鉴了许多概念，如类、对象、继承等。
- 1972年Palo Alno研究中心(PARC)发布了Smalltalk-72，其中正式使用了“面向对象”这个术语。
- Smalltalk的问世标志着面向对象程序设计方法的正式形成。但是这个时期的Smalltalk语言还不够完善。

面向对象方法的发展史

2、完善阶段

- PARC先后发布了Smalltalk-72，76，78等版本，直至1981年推出该语言最完善的版本Smalltalk-80。

- Smalltalk-80的问世被今认为是面向对象语言发展史上最重要的里程碑。迄今绝大部分面向对象的基本概念及其支持机制在Smalltalk-80中都已具备。它是第一个完善的、能够实际应用的面向对象语言。

但是，Smalltalk的应用尚不够广泛，原因是：

① 一种新的软件方法学被广泛接受需要一定的时间。

② 商品化软件开发工作到87年才开始进行。

③ 追求纯OO的宗旨使许多软件开发人员感到不便。

面向对象方法的发展史

3、繁荣阶段

- 自80年代中期到90年代，是面向对象语言走向繁荣的阶段。其主要表现是大批比较实用的OOPL的涌现，例如 C++、Objective-C、Object Pascal、CLOS（Common Lisp Object System）、Eiffel、Actor等。
- OO编程语言分为纯OO语言和混合型OO语言。
  - 混合型语言是在传统的过程式语言基础上增加OO语言成分，在实用性方面具有更大的优势。如C++。
  - 此时的纯OO语言也比较重视实用性。如Eiffel、Smalltalk 、Actor。

面向对象方法现状简介

4、 当前的状况

- 在编程方面，普遍采用语言、类库和可视化编程环境相结合的方式，例如，Visual C++，Visual Basic和Delphi，Java等。

- 到九十年代，面向对象的分析与设计方法已多达数十种，这些方法都各有所长，现在趋于统一。

统一建模语言UML（Unified Modeling language），其推荐的方法是USDP（Unified Software Development Process）；UML是一种面向对象的建模语言，在软件产业界获得了很大的支持。

在面向对象的过程指导方面，目前还没有国际规范发布。

本章内容

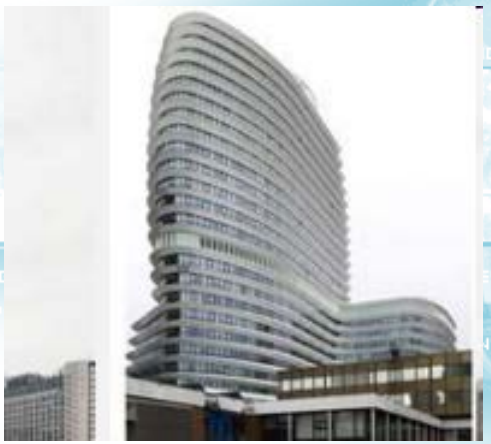
- 传统开发方法中存在的问题
- 面向对象的基本思想
- 面向对象的主要概念及基本原则
- 面向对象方法的主要优点
- 面向对象方法的发展史及现状简介
- 什么是UML
- UML建模工具

为什么需要建模？

- 因为不能完整地理解一个复杂的系统，所以要对它建模

VS

VS



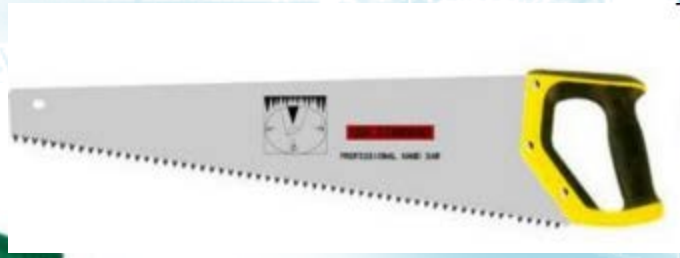
Dog house

Few hours

Little prior planning

lumber

nail



saw

hammer

Tap measure

family demands

lumber

nail

family house

More times

Some detailed Planning:

•time

•materials

hammer

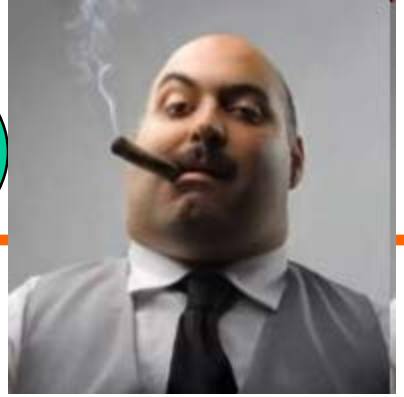
saw

blueprints

Tap measure

buying pre-built materials

User:
Size, shape,
style of the building;
change minds



developing and deploying group

Office building

Your team:
Do extensive planning;
Just a part of a much larger group

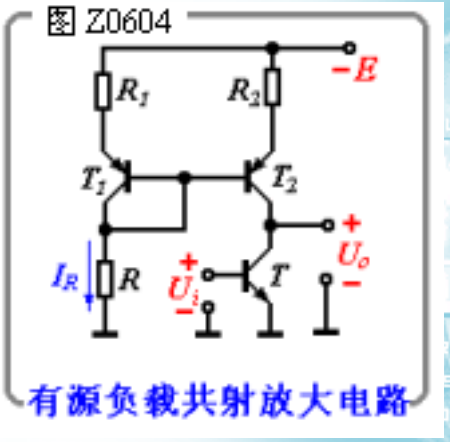
Blueprints

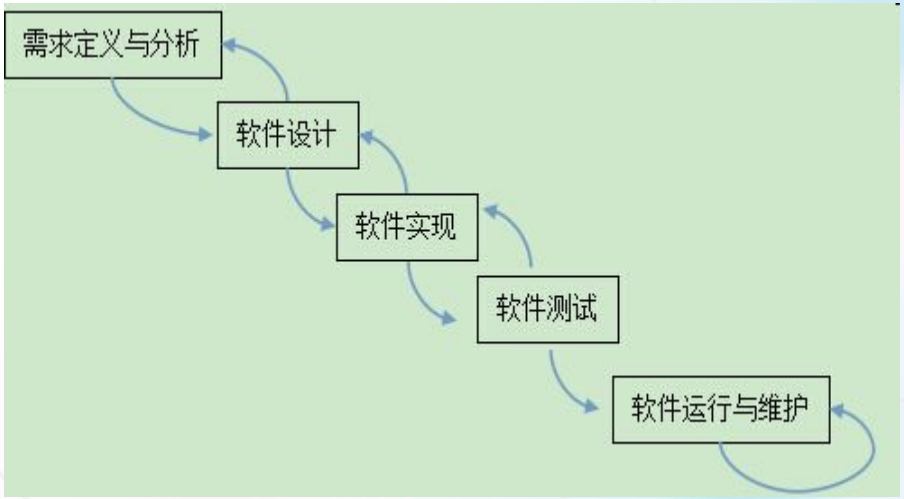
Models

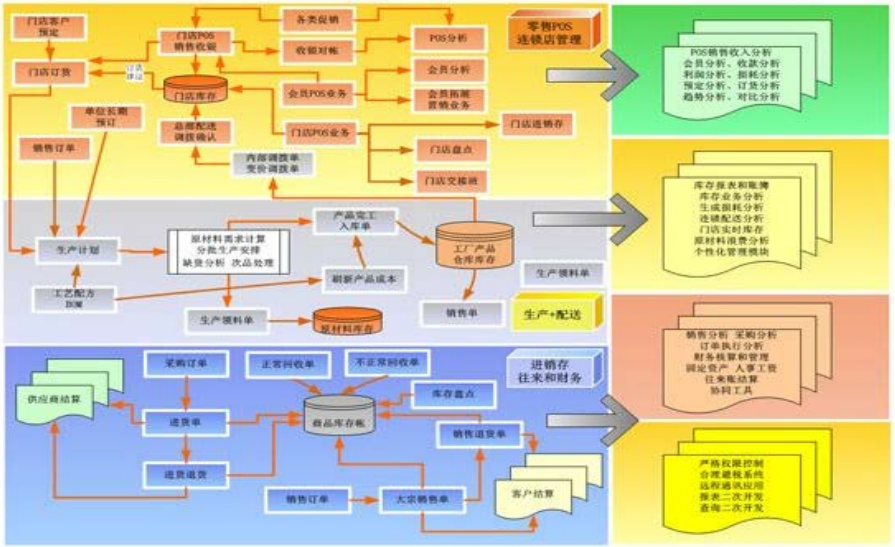
Other teams

模型是对现实的简化

A model is a simplification of reality







常见的模型

- 生活相关：气象图、道路交通图、交通标志…
- 展示相关：建筑物模型、沙盘、公司总部的3D复制品…
- 数据分析相关：条形图、饼状图…
- 业务分析相关：组织结构图、跨职能流程图……
- 设计相关：建筑平面图、管线图、电路板设计图

为什么需要建模？

- 帮助我们更好地理解正在开发的系统
  - ✓ 模型有助于按照实际情况或按照所需要的样式对系统进行可视化
  - ✓ 模型能够规约系统的结构或行为
  - ✓ 模型给出了指导构造系统的模板
  - ✓ 模型对做出的决策进行文档化

建模原理

- 选择要创建什么模型，对如何动手解决问题和如何形成解决方案有着意义深远的影响
- 可以在不同的精度级别上表示每一种模型
- 最好的模型是与现实相联系的
- 单个的模型或视图是不充分的，对每个重要的系统最好用一小组几乎独立的模型从多个视角去逼近

走向UML

- 太多的OO术语令人迷惑
  - Wirfs-Brock——Responsibility
  - Booch——Operation
  - Coad/Yourdon——Service
  - Stroustrup——Function
  - …Method

走向UML

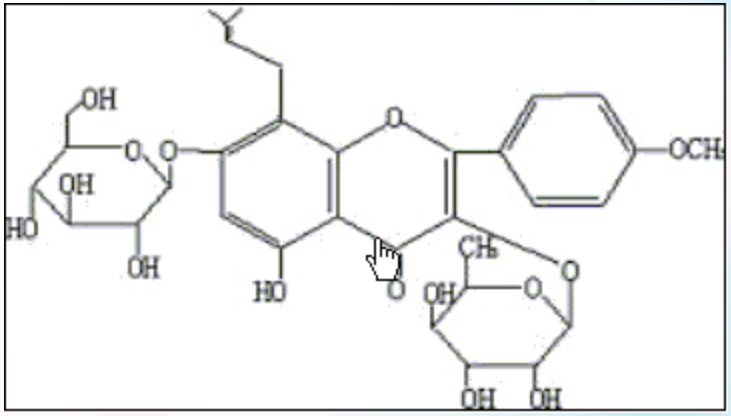
- 没有统一的符号，科学很难发展

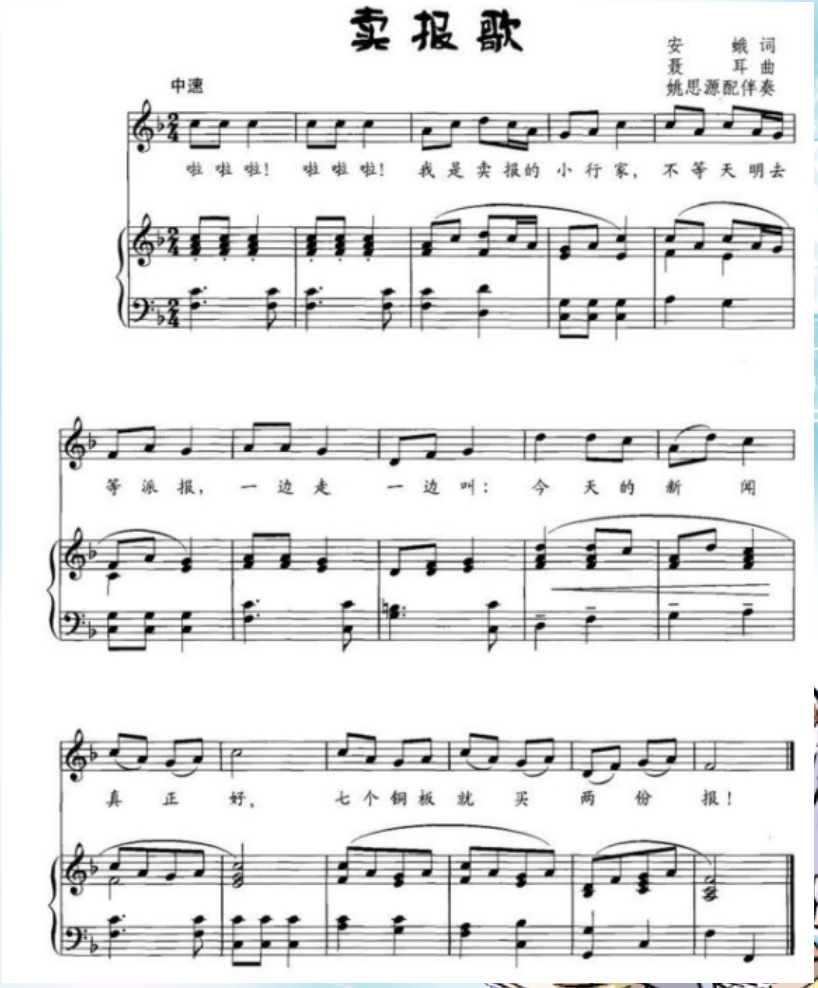
计算$[(-\frac{1}{2})^2+2\frac{1}{2}]\div\{1-[(\frac{\sqrt{3}}{2})^2+(-\frac{1}{2})]\}$

$=[\frac{1}{4}+2\frac{1}{2}]\div\{1-[\frac{3}{4}+(-\frac{1}{2})]\}$

$=2\frac{3}{4}\div\{1-\frac{1}{4}\}$

$=2\frac{3}{4}\div\frac{3}{4}=3\frac{2}{3}.$





UML发展历程

- 第一阶段：OO方法学家的联合行动
  - 1995.10，G Booch和J Rumbaugh联合推出Unified Method 0.8
  - 1996.6，I Jacobson加入，推出UML0.9（不再称”方法“，而改称”建模语言“）
- 第二阶段：公开的联合行动
  - 1997.推出UML1.0，被OMG接纳
- 第三阶段：OMG主持下修订
- 第四阶段：UML的重大修订——UML2
- 第五阶段：2012年，UML2.4提交到ISO申请成为国际标准，成为建模语言国际标准。

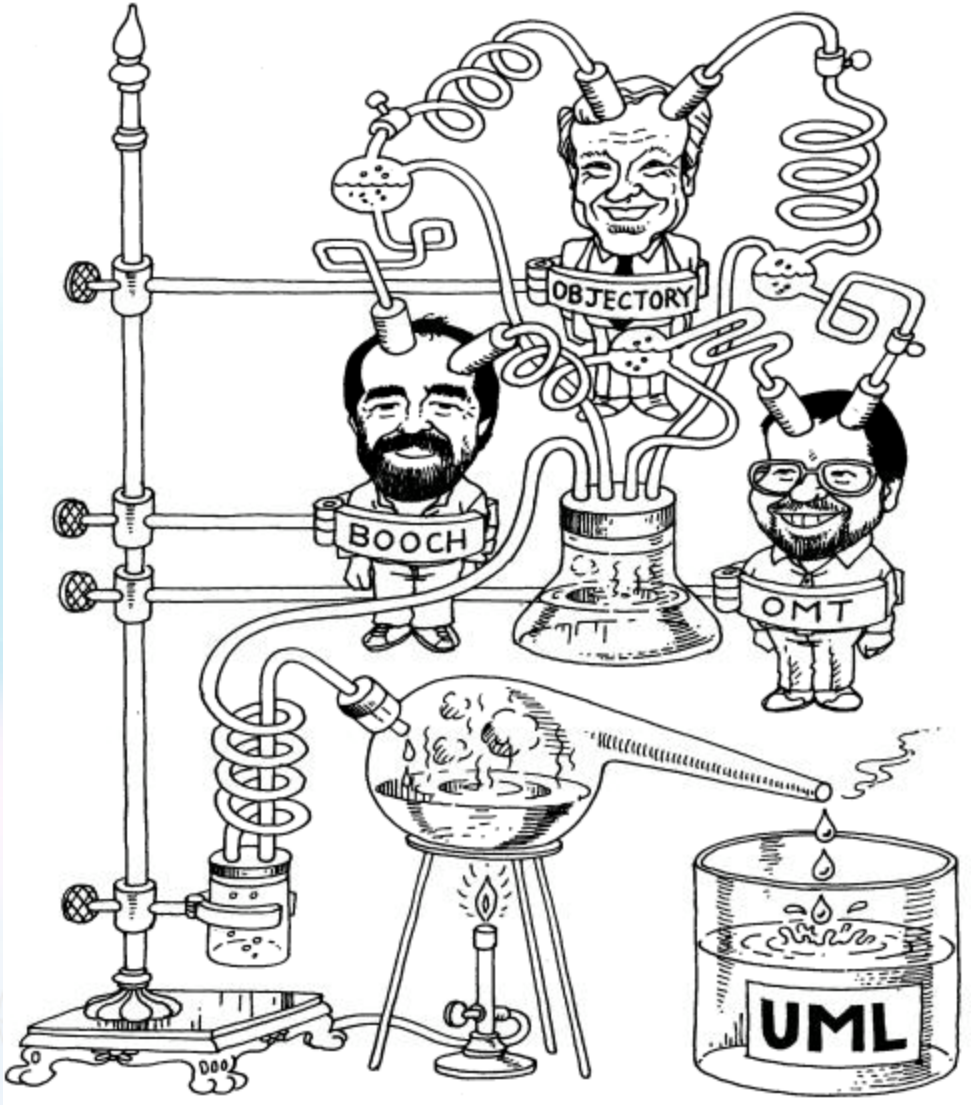
UML标准　http://www.omg.org/spec/UML/

Rational三友

James Rumbaugh

Ivar Jacobson

Grady Booch



UML的特性与发展现状

- UML是一种Language（语言）
- UML是一种Modeling（建模）Language
- UML是Unified（统一）Modeling Language

- 对软件密集型系统的制品进行可视化、详述、构造和文档化
- 应用领域正在逐渐扩展，包括嵌入式系统建模、业务建模、流程建模等多个领域
- 成为“产生式编程”的重要支持技术：

  MDA、可执行UML等

为什么使用UML建模，可以建立什么模型

- **UML是一种应用面很广泛的建模语言**

模型的种类	模型的用途
业务模型	对业务过程、工作流、组织的建模
需求模型	对捕获的需求进行整理和分析的工具，辅助开发人员与用户进行沟通
设计模型	包含高层设计（架构模型）和详细设计模型，用于统一开发人员、沟通设计信息
数据库模型	设计数据库的结构、表结构以及与应用系统的交互
实现模型	用来理清软件的组成、部署方案，为安装与维护人员的工作提供指导

谁应该建模

- 业务建模：以领域专家为主，需求分析人员是主力，系统分析员、架构师可参与
- 需求模型：以需求分析人员为主，系统分析员是主力，领域专家提供指导，架构师和资深开发人员参与
- 设计模型：高层设计模型以架构师为主，系统分析员从需求方面提供支持，资深开发人员从技术实现方面提供支持。详细设计模型则以资深开发人员为主，架构师提供指导。
- 实现模型：以资深开发人员（设计人员）为主，架构师提供总体指导。
- 数据库模型：以数据库开发人员为主，架构师提供指导，资深开发人员（设计人员）予以配合。

常见误区

- UML是一种方法论
- UML就是一堆图形
- UML只能够应用于面向对象开发中
- UML就是Rose里的符号
- UML的学习周期很长、很复杂

本章内容

- 传统开发方法中存在的问题
- 面向对象的基本思想
- 面向对象的主要概念及基本原则
- 面向对象方法的主要优点
- 面向对象方法的发展史及现状简介
- 什么是UML
- **UML建模工具**

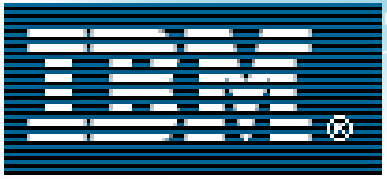
UML建模工具

- MS Visio

- StarUML

http://staruml.io/

- Rational Software Architect



- Rational Rose

- 在线工具 https://www.processon.com/

本章内容回顾

- 传统软件开发方法
- 面向对象的基本思想、主要概念、基本原则及主要优点
- 面向对象方法的发展史及现状简介
- 建模的作用
- UML的发展历程
- UML的概念