

面向对象设计 概述

- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法

面向对象的设计



北京大学

面向对象设计 概述

- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法



第一部分 什么是面向对象的设计（OOD）

概而言之，但不同时期有不同内容及特点。

一、早期的OOD（八十年代至九十年代初）：

历史：从OOP发展到OOD

G. Booch 1982 年发表“Object-Oriented Design”，首次称“面向对象的设计”。

G. Booch 1986 年发表“Object-Oriented Development”较完整地阐述了OOD思想。

两个术语都用OOD作为缩写，内容上也没有根本区别

R. J. Abbott 1983年提出正文分析方法，用规范的英语对问题进行陈述，然后从描述中提取对象及其特征。例：名词——对象，动词——操作。被后来的许多OOD方法所采用。

1986年后，相继出现了一批（早期的）OOD方法



北京大学

面向对象设计 概述

- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法

早期的OOD方法:

Booch86——Object-Oriented Development
面向对象的开发

GOOD——General Object-Oriented Development
通用面向对象的开发

HOOD——Hierarchical Object-Oriented Design
层次式面向对象的设计

OOSD——Object-Oriented Structured Design
面向对象的结构设计

.....



北京大学

面向对象设计 概述

- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法

早期OOD的特点:

- 1、不是基于OOA的
大多基于结构化分析结果（数据流图）
- 2、是OO编程方法的延伸
多数方法与编程语言有关，特别受Ada影响很大
- 3、不是纯OO的
对某些OO概念（如继承）缺少支持，
掺杂一些非OO 概念（如数据流、包、模块等）
- 4、不是只针对软件生命周期的设计阶段
OOD中的“D”——指的是Design 或 Development
多少涉及分析问题（如识别问题域的对象），但很不彻底

早期的
OOD可看
作现今
OOA&D
方法的雏
形



北京大学

面向对象设计 概述

- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法

二、现今（90年代）的OOD

背景：

从结构化分析文档识别OOD的对象并非良策，识别对象的关键问题在于用OO方法进行系统分析。

OO方法从设计发展到分析，出现OOA方法。

OOA和OOD构成完整的OOA&D方法体系。

OOD基于OOA，
识别对象由OOA完成，
OOD主要定义对象如何实现。

有多种**OOA&D**方法：

Booch方法

Coad-Yourdon方法

Firesmith方法

Jacobson方法(OOSE)

Martin-Odell方法

Rumbaugh方法(OMT)

Wirfs-Brock方法

.....



北京大学

面向对象设计 概述

- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法

特点：

1. 以面向对象的分析为基础，一般不依赖结构化分析。
2. 与相应的OOA方法共同构成一种OOA&D方法体系。OOA和OOD采用一致的概念与原则，但属于软件生命周期的不同阶段，有不同的目标及策略。
3. 较全面地体现面向对象方法的概念与原则。
4. 大多数方法独立于编程语言，通过面向对象的分析与设计所得到的系统模型可以由不同的编程语言实现。

定义：

面向对象的设计（OOD）就是在OOA模型基础上运用面向对象方法进行系统设计，目标是产生一个符合具体实现条件的OOD模型。



北京大学

面向对象设计 概述

- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法

- 在面向对象软件开发中运用的较为普遍的开发过程有：统一软件开发过程(Unified Software Development Process, USDP)、基于特定活动组织的开发过程和面向特征的开发过程等。
- 这些开发过程都分为OOA、OOD和OOP等阶段。本课程所讲述的开发过程，是一种基于特定活动组织的开发过程。



面向对象设计 概述

- OOD的发展历史
- **OOD的根本目标**
- OOA与OOD的关系
- OOD方法中的概念和表示法

三、OOD的根本目标: (coad/yourdon)

1、提高软件生产率

设计的投入在编程、测试时得到回报

OO方法使系统更易于理解

分析文档、设计文档、源程序对应良好

功能变化引起的全局性修改较少

OOD结果的复用

2、提高质量

现今的质量观点:

- 不仅是事后通过测试排除错误, 而是着眼于软件开发过程的每个环节, 从分析、设计阶段开始质量保证。

- 高质量不只是没有错误, 而是好用、易用、可移植、易维护, 用户由衷地满意。



北京大学

面向对象设计 概述

- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法

3、加强可维护性

需求是不断变化的（尽管可阶段性地“冻结”）

因素：客户业务、竞争形式、技术发展、规章制度……
——要求设计结果对变化有弹性

设计如何适应不可预见的变化？

——把易变部分和较稳定的部分隔离，
将变化的影响限制在局部

易变性：服务>接口>属性>类



面向对象设计概述

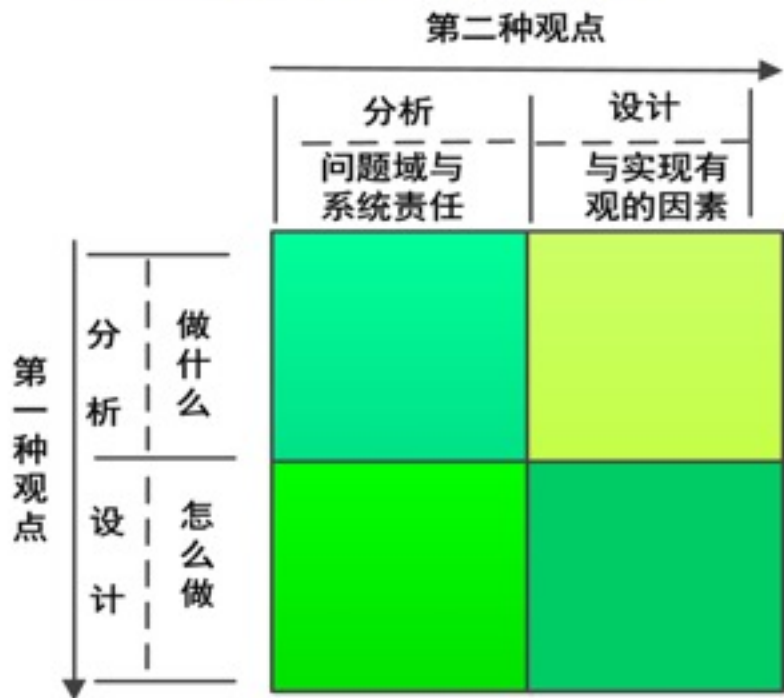
- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法



四、OOA与OOD的关系

第二种观点的理由：

1、OOA与OOD的分工 ——两种不同的观点



关键问题：对象的特征细节
(如属性的数据类型和操作流程
图)，是在分析时定义还是在
在设计时定义？

(1) 在各种分析/设计方法中“做什么”和“怎么做”实际上没有严格的划分”。

(2) 过分强调“分析不考虑怎么做”将使某些必须在OOA考虑的问题得不到完整的认识。

(3) 由于OO方法表示形式的一致，不存在把细化工作留给设计人员的必然理由。

(4) 避免重复地认识同一事物，并有利于分析结果的复用。

例：Rumbaugh方法(OMT)
和Coad/ Yourdon方法



北京大学

面向对象设计 概述

- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法

第二部分 OOD方法概貌

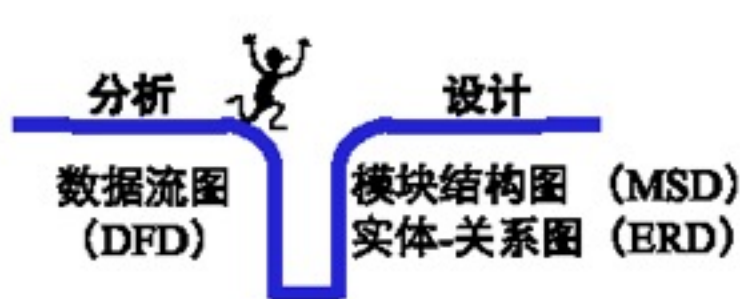
基本按Coad/Yourdon方法讲授，做适当的改进和补充。

概念： 运用与OOA部分相同的概念
——没有增加新概念

对象、类、属性、服务(操作)、封装、继承、消息、关联、聚合、多态、主动对象 等

表示法： 采用与OOA一致的表示法

使分析与设计之间不存在鸿沟



传统方法分析与设计之间的鸿沟



面向对象的分析与设计之间不存在鸿沟



北京大学

面向对象设计 概述

- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法



OOD——按实现条件对OOA模型进行调整，并补充几个新的组成部分（也是由对象构成）

与实现有关的因素：

图形用户界面系统

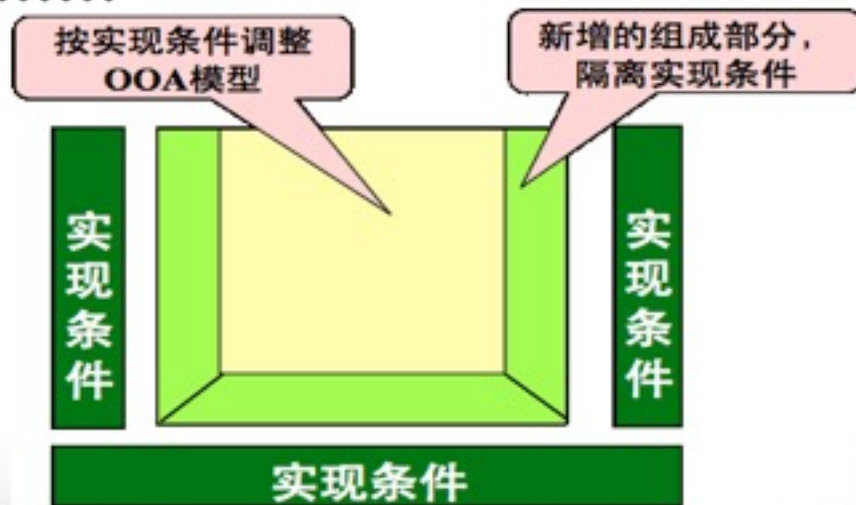
硬件、操作系统及网络

数据管理系统

其他——编程语言、可复用构件库……

基本思想：

尽可能隔离实现条件对系统的影响——提供独立的接口
对不可隔离的因素，按实现条件调整OOA模型



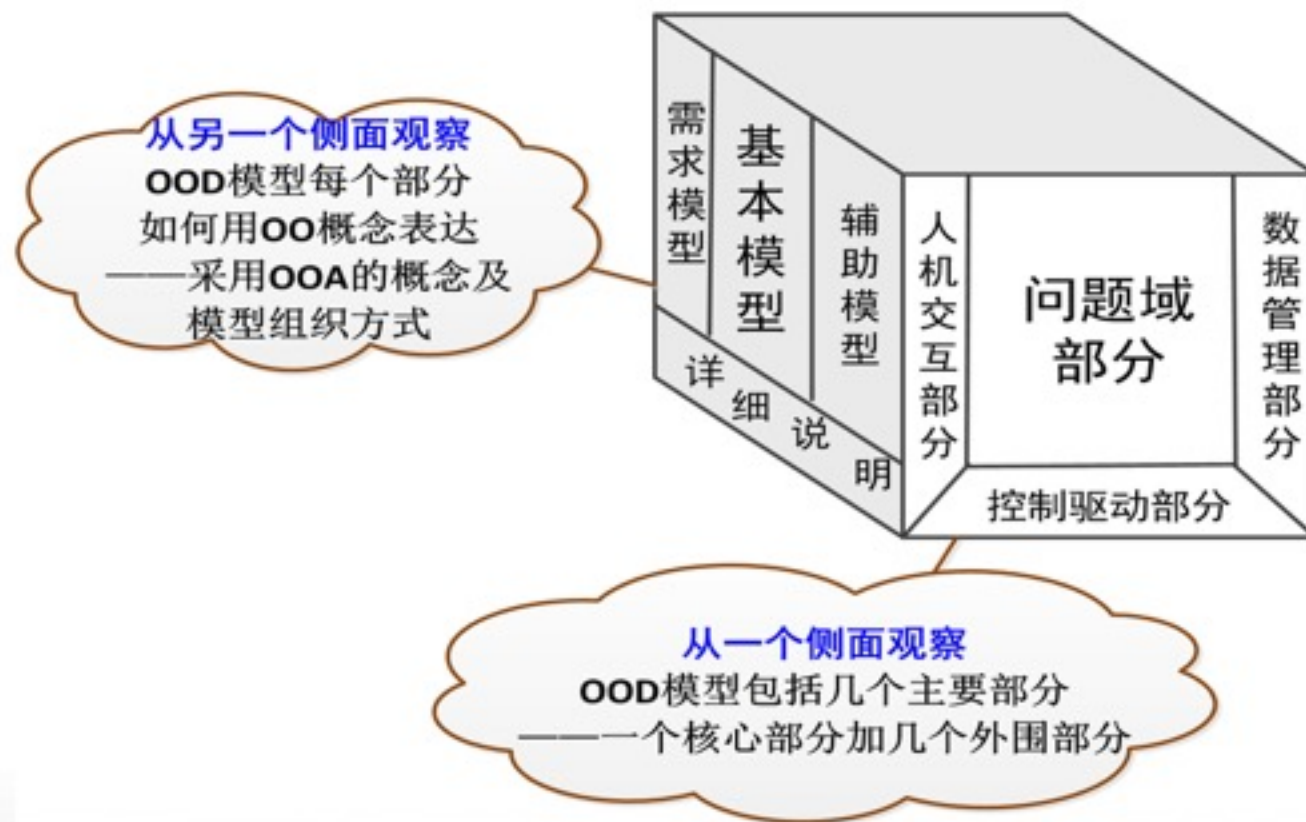
北京大学

面向对象设计 概述

- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法

OOD模型

——从两个侧面来描述



面向对象设计 概述

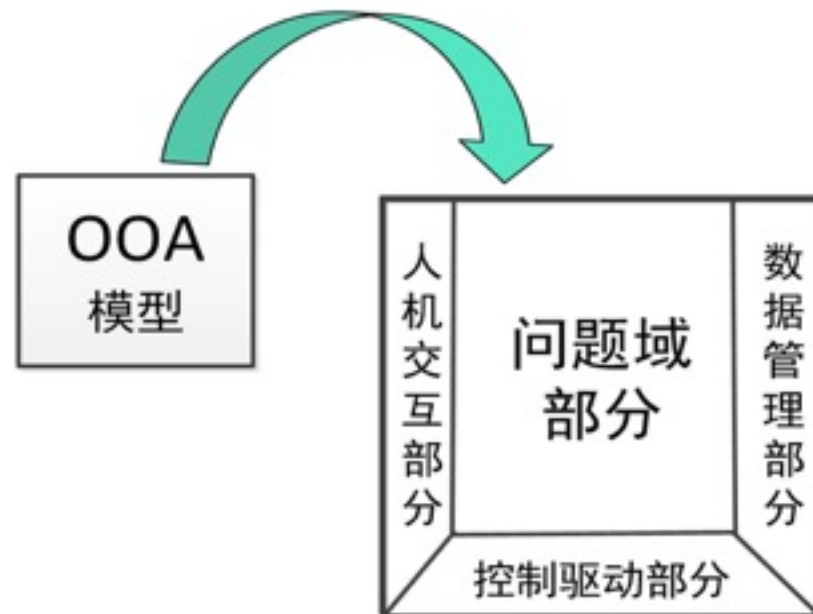
- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法

OOA与OOD的关系：

1、从OOA到OOD不是转换，不是细化；
——是调整和增补

将OOA模型搬到OOD；
进行必要的调整，
作为OOD模型的问题
域部分；

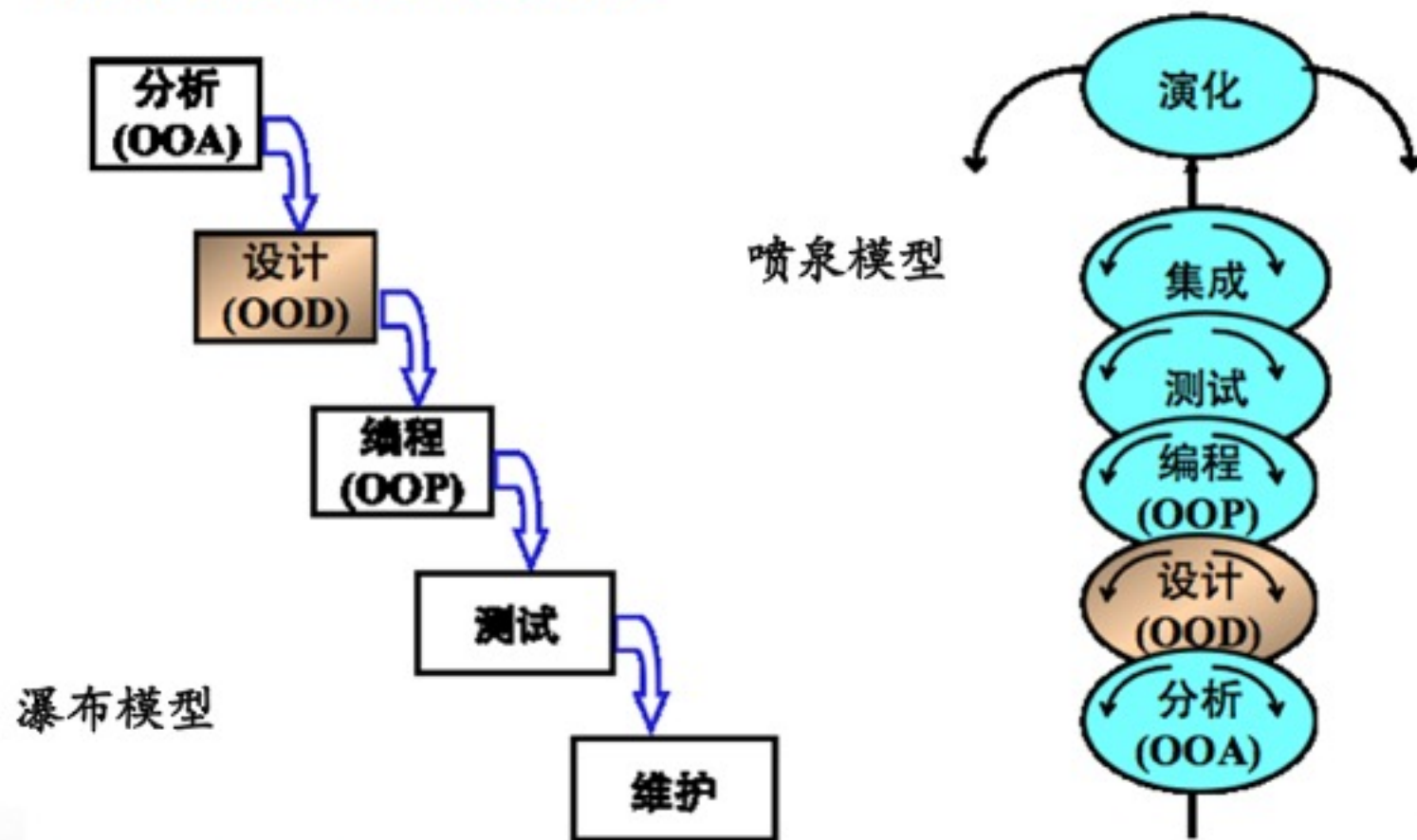
增补其它三个部分，
成为完整的OOD模型



面向对象设计 概述

- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法

不同过程模型中的OOD



面向对象设计 概述

- OOD的发展历史
- OOD的根本目标
- OOA与OOD的关系
- OOD方法中的概念和表示法



OOD过程：
逐个设计OOD模型的四个部分

问题域部分的设计
人机交互部分的设计
控制驱动部分的设计
数据接口部分的设计

不强调次序

每个部分均采用与OOA一致的概念、表示法及活动，
但具有自己独特的策略。



北京大学