

OOAD 与 UML 简介

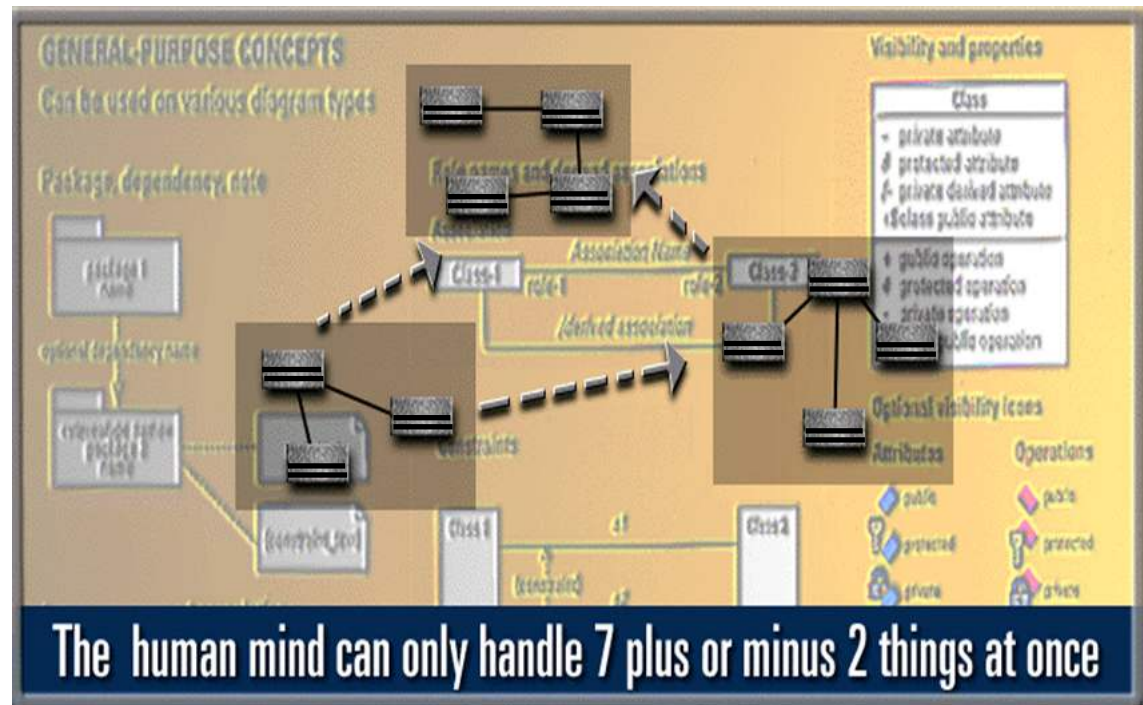
目的

- 了解 OOAD 的基本概念
- 了解建模的基本概念
- 了解 UML 的概念和使用
- 初步认识 UML 元素

软件发展的复杂性

- 伴随信息技术的发展，软件复杂性的增长使软件开发越来越困难

- 操作系统
- 应用软件
- 网络



软件可能是人类制造出来的最复杂的实体

面对对象的分析 and 设计 (OOA&D)

- 面对对象的措施按照人类的自然思维的方式，面对客观世界建立软件模型。
- 充分体现了对复杂系统进行分解、抽象、模块化等思想

OOA根据顾客所了解的真实世界中的对象和概念，发觉和分析对象的内部构成和外部关系，建立精确而简洁的软件系统的对象模型。

OOD 是根据已建立的系统对象模型，利用面对对象技术，进行软件设计。

OOA&D 与老式设计分析相比优点

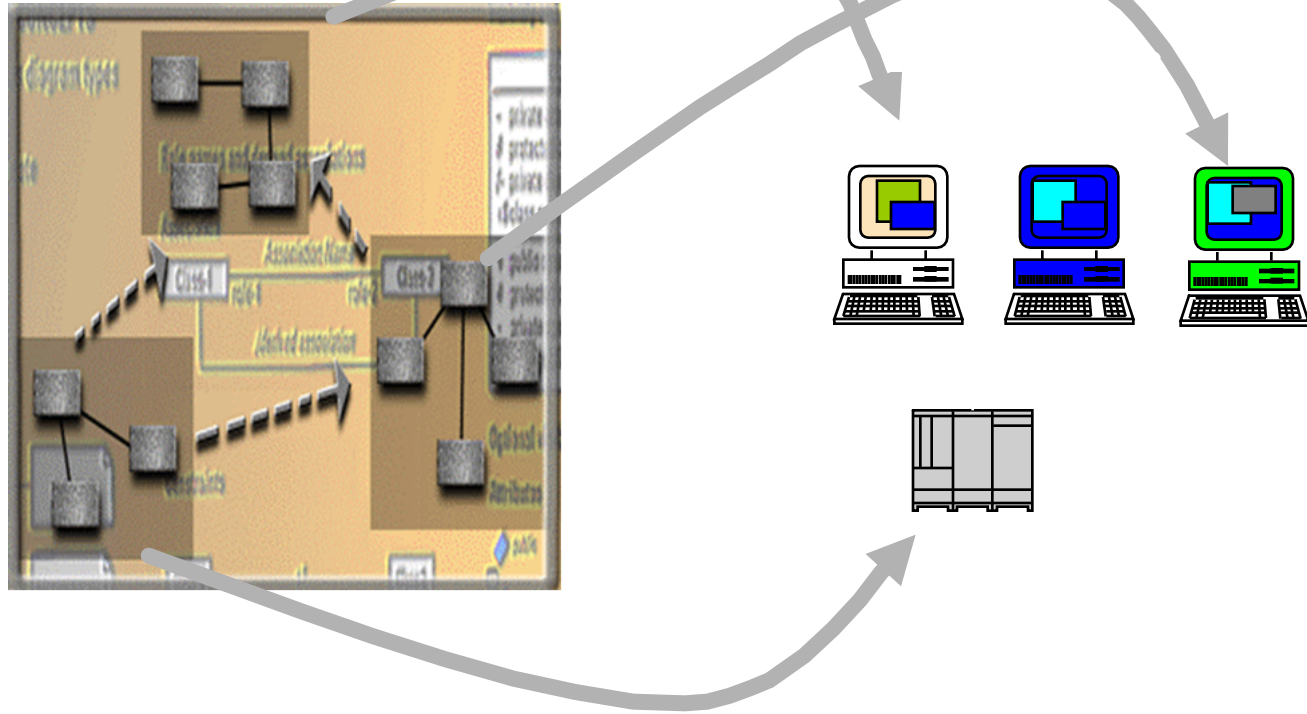
- 在实现的成果和实际问题之间存在一种很接近的匹配关系
- 增进对象的重用
- 因为对象的重用成为可能，从而能够降低错误和维护问题
- 对象的重用还加速了设计和开发的过程
- 符合人类认知的方式，因为这是我们自然的思索方式
- 加强数据封装
- 有利于处理软件开发的复杂性，并帮助生成可修改的、有弹性的软件系统

什么是模型

- 模型是现实的简化
- 模型提供了系统的设计图。模型能够涉及详细的规划，也能够涉及概括性的规划，这种规划高度概括了正在考虑的系统。好的模型涉及那些具有高度抽象性的元素。
 - 模型有利于按原样或根据需要使系统可视化
 - 经过模型能够详细阐明系统的构造或行为
 - 模型能够提供一种指导我们构建系统的模板
 - 模型能够统计已经做出的决策

为何要建模

- 建模是为了让我们更加好地了解将要开发的系统。



建模要实现的目的

- 模型有利于按原样或根据需要使系统可视化。
- 经过模型能够详细阐明系统的构造或行为。
- 模型能够提供一种指导我们构建系统的模板。
- 模型能够统计已经做出的决策。

建模的原则

- 选择创建什么模型对于怎样处理问题以及怎样形成处理方案有很深远的影响。
- 每一种模型能够在不同的精度级别上表达。
- 最佳模型与现实情况是紧密相连的。
- 一种模型不足以全方面地反应实际情况。每个具有一定规模的系统都是经过一组几乎独立的模型来表达的。

面对对象的建模

- 从算法角度
- 从面对对象的角度

选择以面对对象的方式来分析问题将衍生出许多成果：一种良好的面对对象的体系应具有什么构造？项目应该创建哪些部件？应该由谁来创建这些部件？应该怎样度量它们？

对面对对象的系统进行可视化、规格阐明、构建和文档化正是统一建模语言（UML）的目的。

UML 简介

- 统一建模语言UML (Unified Modeling Language) 是一种绘制软件蓝图的原则语言。
- 能够用UML对软件密集型系统进行可视化、详述、构造和文档化。
- 从企业信息系统到基于Web的分布式应用，甚至严格的实时嵌入式系统都适合于用UML来建模。
- 它是一种富有体现力的语言，能够描述开发所需要的多种视图，然后以此为基础开发系统。

UML 为模型可视化提供表达法

- 阐明顾客与系统的交互的用例图
- 阐明逻辑构造的类图
- 阐明对象和链接的对象图
- 阐明行为的状态图
- 阐明软件的物理构造的构件图
- 显示软件与硬件配置之间的映射关系的布署图
- 阐明行为的交互图（即协作图和时序图）
- 阐明用例中事件流的活动图

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/657015035012006163>