

## OOAD第一章

### 1. OOA

对问题域进行理解并抽象成概念（对象）

找到概念之间的联系

以及概念中的属性

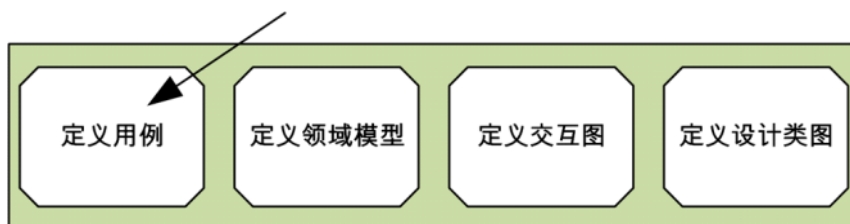
### 2. OOD

OOD是面向对象方法在软件设计阶段应用与扩展的结果，通常分为两个阶段

高层设计：建立应用的体系结构

低层设计：集中于类的详细设计

准则：抽象，信息隐藏，模块化，弱耦合，强内聚，可重用



### 3. 用例

需求分析可能包括人们如何使用应用的情节或场景，这些情节或场景可以被编写成用例 (use case)

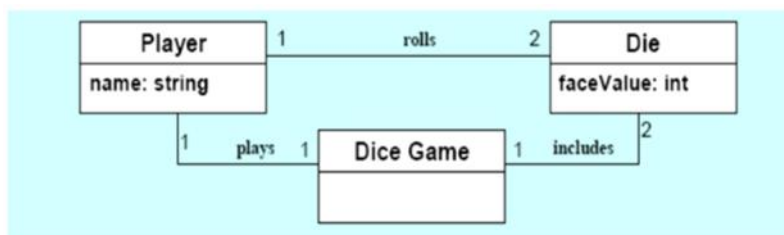
用例并不是面向对象的产物，它只是对情节的记录，但是用例是需求分析中的一种常用工具

### 4. 领域模型

面向对象分析的结果可以表示为领域模型 (domain model)，在领域模型中展示重要的领域概念或对象

领域模型表述了重要的概念及其关联和属性

领域模型并不是对软件对象的描述，它使真实世界领域中的概念和想象可视化，因此，它也被称为概念对象模型 (conceptual object model)



### 5. 交互图

由于面向对象设计关注定义软件对象以及对象间的协作。

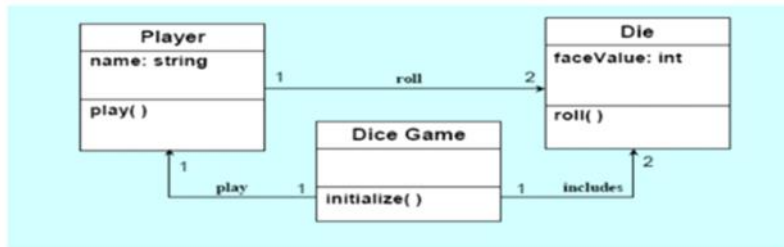
交互图是描述协作的常见表示法。

交互图分为通信图和顺序图。通信图描述的是对象之间如何传递消息，而顺序图还加入了时序表达

### 6. 类图

设计类图 (design class diagram) 来有效的表示类定义的静态图视图





## 7. 统一过程(UP)——初始、细化、构造、移交 软件开发过程

描述了构造、部署以及维护软件的方式

### 统一过程

已经成为一种流行的构造面向对象系统的迭代软件开发过程

UP实践提供了如何实施OOA/D的示范例子

UP具有灵活性,可以应用于轻量级和敏捷方法

### 1初始阶段

初始阶段是UP生命周期四个阶段中的第一个阶段。这一阶段有4个主要目标;

- (1)了解要创建什么样的系统。确定系统的外貌、范围和边界。
- (2)确定系统的主要功能。确定哪些用例是最重要的, 识别系统必须被实现的功能。
- (3)制定至少一个可行的方案。至少确定一个可行的系统架构, 确保在适当的风险和代价下创建系统。
- (4)了解与项目相关的成本、时间表和风险。

初始阶段结束时是第一个重要的里程碑: 生命周期目标(Lifecycle Objective)里程碑

### 2细化阶段

细化阶段是建立系统架构的基线, 以便为构造阶段的主要设计和实施工作提供一个稳定的基础的阶段。这一阶段的主要目标有:

- (1)更明确的认识需求。即是对系统的大部分需求达成更明确的认识。
- (2)设计、实现、验证系统架构并建立架构基线, 化解主要风险。
- (3)细化开发案例, 制定更精确的时间表和费用估算。

细化阶段结束时是第二个重要的里程碑: 生命周期架构里程碑。其评价项目基本的生存能力

### 3 构造阶段

构造阶段是将主要精力集中在用例设计、实现以及测试来充实一个完整的系统的阶段。

这一阶段的主要目标为:

- (1)在确定的架构基线基础上, 实现系统的并行开发
- (2)迭代开发准备给用户的完整产品。通过描述剩余的用例和其他需求、补充设计细节、实现并测试软件来开发出系统的第一个可运行的版本。

构造阶段结束时是第三个重要的里程碑: 初始功能(Initial Operational)里程碑, 它决定了产品是否可以在测试环境中进行部署

### 4移交阶段

移交阶段是系统正式投入运行前的阶段, 要达到的主要目标是确保软件完全满足用户需

求。这一阶段的主要任务有系统测试、系统部署和人员培训等。

交付阶段的终点是第四个里程碑：产品发布(Product Release)里程碑。此时，要确定目标是否实现。

## 8. OO开发

在OO开发中，至关重要的能力是熟练地为软件对象分配职责