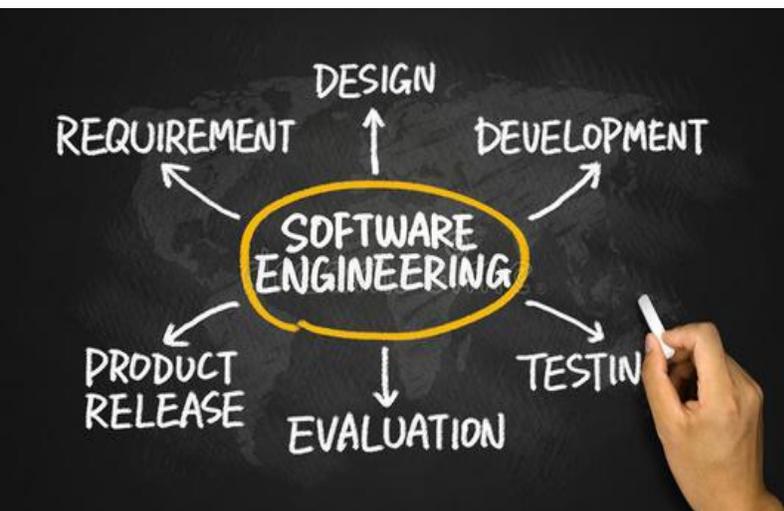




软件工程基础

—— 第9章 需求建模：基于类的方法



计算机学院 孟宇龙

9.1 识别分析类

9.2 描述属性

9.3 定义操作

9.4 类-职责-协作者建模

9.5 关联和依赖

9.6 包

关键概念

- 分析类
- 分析包
- 关联
- 属性
- 协作性
- CRC建模
- 依赖性
- 语法解析
- 运维
- 职责

- 结构化
 - 需求建模的一种视角，被称作结构化分析，它考虑数据和把数据转换为独立实体的过程。
 - 数据对象的建模方式是，定义对象的属性和关系。
 - 操纵数据对象的过程的建模方式是，表明当数据对象流过系统时它们如何转换数据。
- 面向对象
 - 分析建模的第二种方法，被称作面向对象分析，该方法关注于
 - 定义类
 - 类彼此之间协作以影响客户需求的方式

- 基于类的建模表示：
 - 系统操作的**对象**
 - 应用于对象间能有效控制的操作（也称为方法或服务）（**属性**）
 - **对象间的关系**（某种层级）
 - 定义出现在类之间的协作
- 基于类的模型的**元素包括类和对象、属性、操作、CRC模型、协作图和包。**

- 通过检查需求模型开发的使用场景，对系统开发的用例进行“语法解析”
 - 带有下列划线的每个名词或名词词组可以确定为类，并将这些名词输入到一个简单的表中。
 - 标注同义词。
 - 如果要求某个类（名词）实现一个解决方案，那么这个类就是解决方案空间的一部分；否则，如果只要求某个类描述一个解决方案，那么这个类就是问题空间的一部分。
- 不过一旦分离出所有的名词，我们该寻找什么？

- 分析类表现为如下方式之一：
 - 外部实体（例如，其他系统、设备、人员），产生或使用基于计算机系统的信息
 - 事物（例如，报告、显示、字母、信号），问题信息域的一部分
 - 偶然事件或事件（例如，所有权转移或完成机器人的一组移动动作），在系统操作环境内发生
 - 角色（例如：经理、工程师、销售人员），由和系统交互的人员扮演
 - 组织单元（例如，部门、组、团队），和某个应用系统相关
 - 场地（例如：制造车间或码头），建立问题的环境和系统的整体功能
 - 结构（例如：传感器、四轮交通工具、计算机），定义了对象的类或与对象相关的类
- 例子见p109

- **保留信息。** 只有记录潜在类的信息才能保证系统正常工作，在这种分析过程中的潜在类是有用的。
- **所需服务。** 潜在类必须具有一组可确认的操作，这组操作能用某种方式改变类的属性值。
- **多个属性。** 在需求分析过程中，焦点应在于“主”信息；事实上，只有一个属性的类可能在设计中有用，但是在分析活动阶段，最好把它作为另一个类的某个属性。
- **公共属性。** 可以为潜在类定义一组属性，这些属性适用于类的所有实例。
- **公共操作。** 可以为潜在类定义一组操作，这些操作适用于类的所有实例。
- **必须需求。** 在问题空间中出现的外部实体，和任何系统解决方案运行时所必须的生产或消费信息，几乎都被定义为需求模型中的类。

9.2 描述属性

- 属性描述了已经选择包含在需求模型中的类。
 - 为一个职业棒球手建立两种不同的类。
 - **对棒球手的统计软件:** 名字、位置、平均击球次数、担任防守百分百, 从业年限、比赛次数等相关的。
 - **对养老基金软件:** 平均工资、充分享受优惠权后的信用、所选的养老计划、邮件地址等。

9.2 描述属性

每个类都应该能够回答如下问题

什么数据项能够在当前问题环境内完整地定义这个类？

9.3 定义操作

当为分析类定义操作时，集中于面向问题的行为而不是实现所要求的行为

- 操作定义了某个对象的行为
- 做语法解析的处理说明，关注动词
- 操作可以划分为4种类型：
 - (1) 以某种方式操作数据（例如：添加、删除、重新格式化、选择）
 - (2) 执行计算的操作
 - (3) 请求某个对象的状态的操作
 - (4) 监定义操作
- 视某个对象发生某个控制事件的操作

9.4 类-职责-协作者建模

- **类-职责-协作者(CRC)**
- 建模提供了一个简单方法，可以识别和组织与系统或产品需求相关的类。
- 用如下文字解释CRC建模：
 - CRC模型实际上是表示类的标准索引卡的集合。这些卡片分为三部分，顶部写类名，卡片主体左侧部分列出类的职责，右侧部分列出类的协作者。

9.4 类-职责-协作者建模

ClassFloorPlan	
Description:说明	
Responsibility:职责	Collaborator:协作者
defines floor plan name/type	定义住宅平面图的名称/类型
manages floor plan positioning	管理住宅平面图的布局
scales floor plan for display	缩放显示住宅平面
scales floor plan for display	合并墙、门和窗
incorporates walls, doors and windows	Wall
shows position of video cameras 显示摄像头的位置	Camera

CRC模型索引卡

- **实体类**

- 也称作模型或业务类，是从问题说明中直接提取出来的（例如FloorPlan和Senor）。

- **边界类**

- 用于创建用户可见的和在使用软件时交互的接口（如交互屏幕或打印的报表）。

- **控制类**，自始至终管理“工作单元” [UML03]。也就是说，设计控制类可以管理

- 实体类的创建或更新；
- 当边界类从实体对象获取信息后的实例化；
- 对象集合间的复杂通信；
- 对象间或用户和应用系统间交换数据的确认。

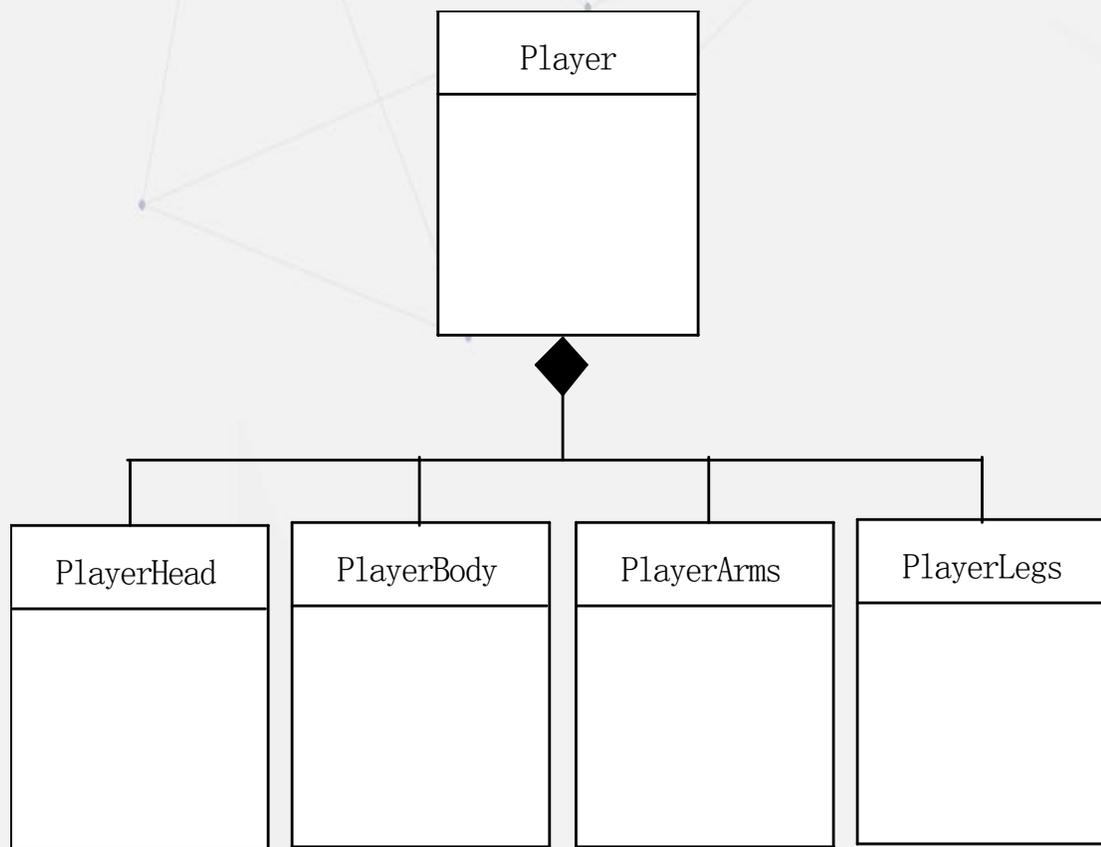
9.4 类-职责-协作者建模

给类分配职责时的5个基本原则：

- 系统智能应用分布在所有类中以求最佳地满足问题的需求。
- 每个职责的说明应尽可能具有普遍性
- 信息和与之相关的行为应放在同一个类中
- 某个事物的信息应局限于一个类中而不要分布在多个类中。
- 适合时，职责应由相关类共享。

- **类可以通过下面的一种或两种方法实现其职责：**
 - 类可以使用其自身的操作控制各自的属性，从而实现特定的职责。
 - 一个类可以和其他类协作。
- 协作是以客户职责实现的角度表现从客户到服务器的请求
- **要识别协作可以通过确认类本身是否能够实现自身的每个职责。**
- 类之间三种不同的通用关系[WIR90]:
 - is-part-of (是.....一部分) 关系
 - has-knowledge-of (有.....知识) 关系
 - depends-upon (依赖.....) 关系

9.4 类-职责-协作者建模



复合聚合类

Has-knowledge-of

- 一个类必须从另一个类中获取信息时，就建立了Has-knowledge-of 关系。
- 如ControlPanel 和Sensor在determine-sensor-status 职责中的关系。传感器信息需要从Sensor对象获取。

Depends-upon

- PlayHead通常必须连接到PlayerBody。其位置属性由PlayerBody确定。因此， PlayHead依赖于 PlayerBody
- 上述关系的实例和CRC模型的评审见p116

9.4 类-职责-协作者建模

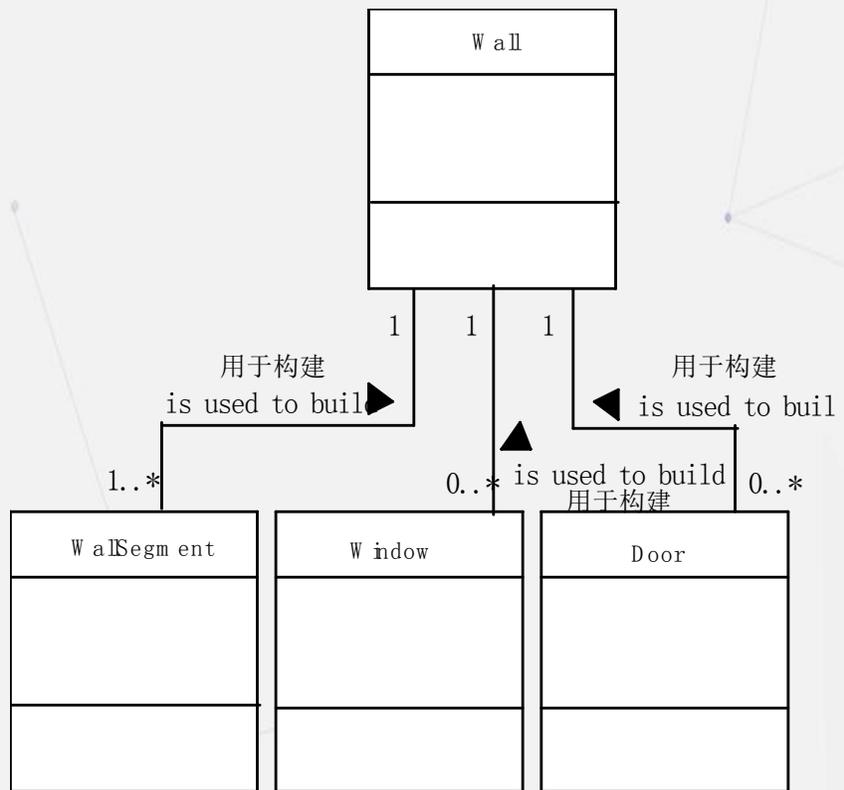
评审CRC模型

- 所有的参与者（CRC模型）评审的人员拿到一部分CRC卡模型索引卡。拆分协作卡片（也就是说每个评审员不得有两张存在协作关系的卡片）。
- 分类管理所有的用例场景（以及相关的用例图）。
- 评审组长细致地阅读用例。当评审组长看到一个已命名的对象时，给拥有相应类索引卡的人员一个令牌。
- 当令牌传递时，类卡的拥有者需要描述卡上记录的职责。评审组确定（一个或多个）职责是否满足用例需求。
- 如果记录在索引卡上的职责和协作不能满足用例，就需要修改卡片。修改可能包括定义新类（和相关的CRC索引卡），或者在已有的卡上说明新的或修改的职责、协作。

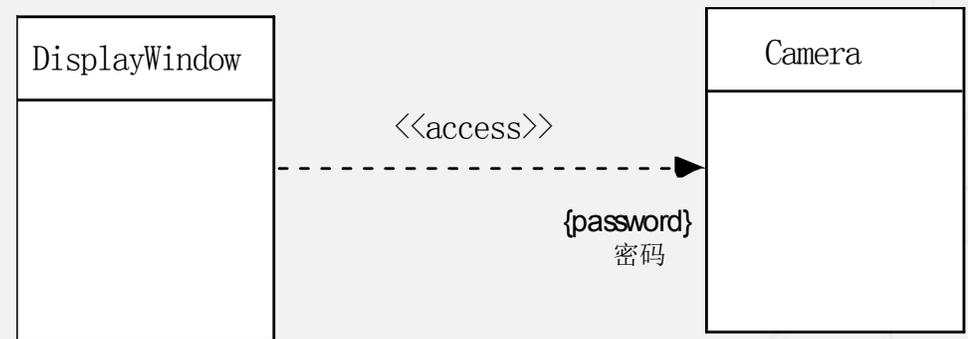
关联可以更进一步的指出多样性

- 两个分析类以某种方式相互联系着，在UML中，这些联系被称作关联
- 关联可以进一步地指出多样性（术语基数用于数据建模中）
- 在很多事例中，两个分析类之间存在客户-服务器关系。
- 这种情况下，客户类以某种方式依赖于服务器类并且建立了依赖关系

9.5 关联和依赖



多样性



依赖

9.6 分析包

- 将分析建模的各种元素（如用例、分析类）以一种方式分类，**分组打包**后称其为分析包。
- 每个包中分析类名字前的加号表示该类是公共可见的，因此可以从其他包访问。
- 包中的任何元素之间可以添加其他符号。负号表示该元素对其他包是隐藏的，#号只能由指定包中的类访问表示该元素。

