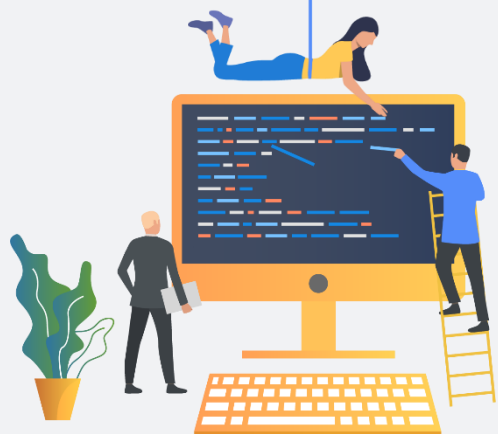




第五章 静态建模

知识点：类和类图





1. 类的定义和表示方法

2. UML中的类图

3. 类的分类

- 类是面向对象系统中最重要单位，它是一组具有相同属性、操作、关系和语义对象的描述。
- 一个类可以实现一个或多个接口。
- 可以用类来捕获正在开发的系统中的词汇。可以用类描述软件事物和硬件事物，甚至也可以用类描述纯粹概念性的事物。
- 结构良好的类具有清晰的边界，并形成了整个系统的职责均衡分布的一部分。

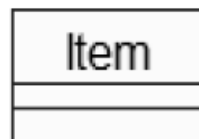
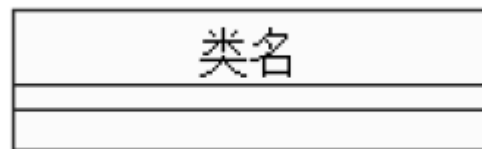


UML中类的表示方法

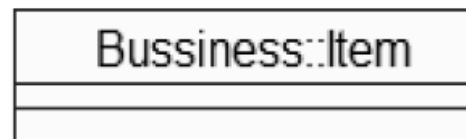
- 在UML中，类用一个矩形符号来表示。

该矩形被水平线划分为3个部分，分别用来定义类的

- 名称 (Name)
- 属性 (Attribute)
- 操作 (Operation)



简单名



路径名



辨析：类和对象的区别



名称:

- 类：类名是一个有意义的标识符
- 对象：对象名加下划线，对象名后面可以接冒号和类名



属性

- 类：定义属性的类型和属性名
- 对象：与类的属性名相同，但是有具体值



操作（方法）

- 类：定义操作（函数）——返回类型、操作名、传递参数。
- 对象：与类中的操作定义相同



类图



类图 (Class Diagram) 是描述类、接口以及它们之间关系的图，用来定义系统中各个类的静态结构。



类图通过系统中的类以及各个类之间的关系描述系统的静态视图。



类图与数据模型有相似之处。但是类图不仅显示了系统内信息的结构，也描述了系统内信息的行为。类图中的类可以直接在某些面向对象编程语言中被实现。






定义类的属性

属性用来描述类的特征。

- 语法格式：

可见性 属性名[多重性]：类型[=初值]

可见性	说明	UML 图注	Rose 图注
Public	表示其为公有成员，其他类可以访问（可见）。	+	
Protected	表示其为保护成员，一般用于集成，只能被本类及派生类使用。	#	
Private	表示其为私有成员，不能被其他类访问（不可见），可缺省。如果没有特别说明，属性都应该是私有的。	-	



可见性 属性名[多重性]：类型[=初值]



多重性：任选项，用多值表达式表示，格式为：低值...高值

- 低值、高值为正整数
- 0..* 表示0到无限多个
- 可缺省，表示1..1, 只有1个



类型：属性的类型，如整型、字符串、对象等。



初始值：任选项，初值可作为创建该类对象时这个属性的默认值。



类属性：用下划线标识，该类的所有对象之间共享该属性。

定义属性时，属性名和属性的类型是必须的，其他部分都是任选的。






定义类的操作（方法）

操作是类的行为特征或动态特征。

- 语法格式：

可见性 操作名（[参数表]）：返回列表[{特征描述}]

可见性	说明	UML 图注	Rose 图注
Public	表示其为公有成员，其他类可以访问（可见）。	+	
Protected	表示其为保护成员，一般用于集成，只能被本类及派生类使用。	#	
Private	表示其为私有成员，不能被其他类访问（不可见），可缺省。如果没有特别说明，属性都应该是私有的。	-	



可见性 操作名 ([参数表]) : 返回列表[{特征描述}]



参数表：用逗号分隔的形式参数序列。

每个参数的语法：**参数名**：**类型**[=初值]。

当操作的调用者未提供参数时，该参数就使用默认值。



返回列表：回送调用对象消息的类型，格式：**返回类型或返回值=类型**，.....。

返回类型：向调用对象回送一个类型的返回值。

返回值=类型1，类型2，.....：向调用对象回送多个返回类型的值。



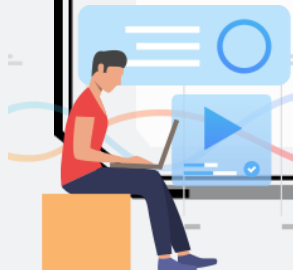
[{特征描述}]：任选项，描述该操作的特征，通常不直接展示在类图中。包括：

- 前置条件：满足该条件（为真）调用本操作。
- 后置条件：执行本操作后该条件为真。
- 某算法指定执行该操作。
- 用特征（操作名、回送型、参数表）来指定该操作。



类操作：用下划线标识的操作，在没有对象实例的情况下被调用，只允许访问本类属性，通常把一些通用的操作定义为类操作。

定义操作时，操作名（）和返回类型是必须的，其他部分是任选的。



命名规范



类名：首字母大写的名词（名词短语），如Student、BankAccount。




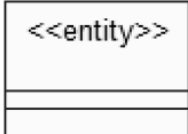

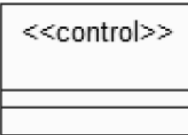
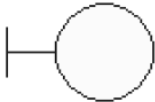
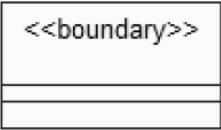
属性名可以是像类名那样的文本文字。在实际应用中，属性名是描述属性所在类的一些特性的简单名词或名词短语。通常要将属性名中除第一个词之外的每个词的第一个字母大写，例如familyName。



操作名是描述它所在类的一些行为的短动词或动词短语。通常要将操作名除第一个词之外的每个词的第一个字母大写，如move() 或 isEmpty()。



MVC架构思想

类型	Icon（图符型）	Label（标签型）
实体类		
控制类		
边界类		



实体类 entity

实体类是用于对必须存储的信息和相关行为建模的类。它主要是作为数据管理和业务逻辑处理层面上存在的类别。

实体类的主要职责是存储和管理系统内部的信息，它也可以有行为。实体对象经常是被动和永久性的。

我们经常是从词汇表（在需求阶段制定）和业务领域模型（如果进行了业务建模，则在业务建模阶段中建立）中找寻到实体类的。



控制类 controller

控制类用于对一个或几个用例所具有的事件流的控制行为进行建模，控制一个用例中的事件顺序。

控制类用于在系统中协调行为。简单情况下，系统可以在没有控制对象的情况下执行某些用例（仅使用实体对象和边界对象）。但是较复杂的用例一般都需要一个或多个控制类来协调系统中其他对象的行为。

控制类有效地将边界对象与实体对象分开，让系统更能适应其边界内发生的变更。

控制类还将用例所特有的行为与实体对象分开，使实体对象在用例和系统中具有更高的复用性。



边界类 boundary

边界类是对系统外部环境与其内部运作之间的交互进行建模。这种交互包括转换事件，并记录系统表示方式（例如接口）中的变更。边界类描述外部参与者与系统之间的交互，识别边界类可以帮助开发人员识别出用户对界面的需求。

边界对象将系统与其外部环境的变更（与其他系统的接口的变更、用户需求的变更等）分隔开，使这些变更不会对系统的其他部分造成影响。

一个系统可能会有多种边界类比如：用户界面类、系统接口类、设备接口类等。





软件系统分析与设计



苏州大学 杨 洋