

单项练习——类图

实验目的：

1. 掌握类图的涵义和内容
2. 掌握类图的绘制方法
3. 掌握类图的使用范围

实验内容：

画出下列描述类图：

商品销售系统中，用户的信息包括两个类，基本信息和地址，用户购买商品下订单，有不同的付款方式：现金和刷卡。商品按类别分类。

要求给出类的属性和一些操作，以及类之间的各种关系。必要的地方要求定义接口。

实验指导：

1. 类图的简介

类图可以用来展示一切面向对象系统的构造块。类图是静态视图，描述了模型或模型的一部分以及这些模型具有的属性和行为而不是实现这些操作的详细方法。类图常常用来表达类和接口之间的关系。关系包括：实现、泛化、关联和依赖。

类是这样一种元素，它定义了一个对象能够产生的属性和行为。行为用类能够理解的可能的消息以及每个消息对应的操作的描述。类也可以使用标签值和原型定义约束。

类符号是用一个带有类名和可选操作和属性名的矩形表示。分隔线用来分离类名、属性和方法。一般情况下，类名在矩形的最上方，其次是属性，然后是方法。其中属性和方法是具有可见性的，通常“-“表示 private，“+”表示 public，“#”表示 protected，“~”表示 package 的可见性。见图 1 所示。

Rectangle		
-	length	: int
#	width	: int
*	color	: int
+	center	: Point
+	area ()	: int
#	getLength ()	: int
*	getWidth ()	: int

图 1 类示例

接口是一个约定，它定义了实现它的行为规范。要实现一个接口，类要求提供一个必须的行为，这个行为允许系统用同一个方法对待不相关的元素，这些是通过接口实现的。接口的符号和类的符号相似，但接口还可以用一个圆表示。见图 2。

实现被用来表达模型中的实现——事务处理或者需求被一个或者多个用例实现，这些模

型再被一个或者多个类实现。在系统的设计中，在模型的抽象层次上保证系统元素的大致的蓝图并同时反映出小图的约束和定义细节。实现用虚线加实心箭头表示。

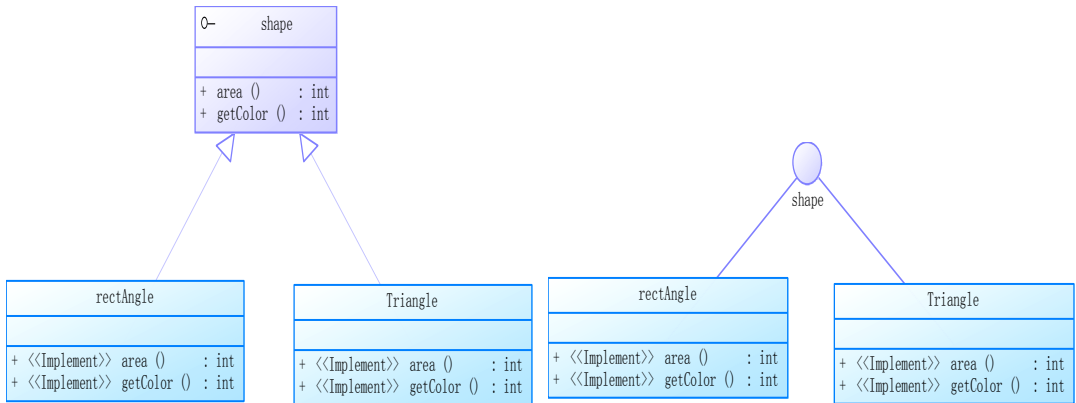


图 2 接口及实现

关联意味着两个元素之间有关系，通常是在一个类中实现了另一个的一个实例。关联可以包括每个的角色名、数目、方向和约束。还有两种特殊的关联，聚合和组合。聚合用来描述一个元素是另一个元素的组成部分，用空心菱形表示。组合是用来描述一种更强的包含关系，主体消失客体也跟着消失，客体消失主体还可以继续存在。

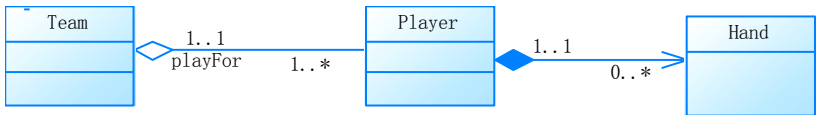


图 3 关联

关联类是这样一种构造，它允许关联具有方法和属性。见图 4。

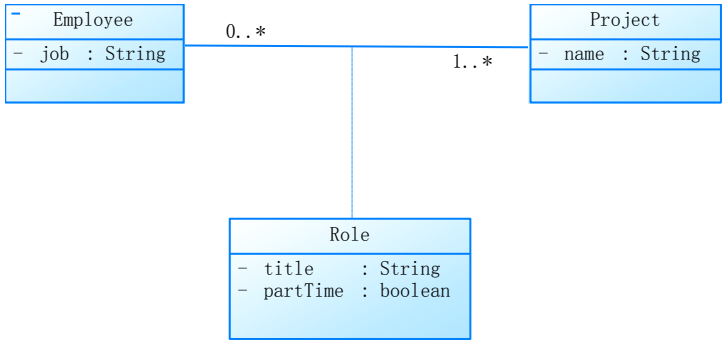


图 4 关联类

泛化通常用来指出层次结构。绘制时从父类到子类。实现是子类继承父类。

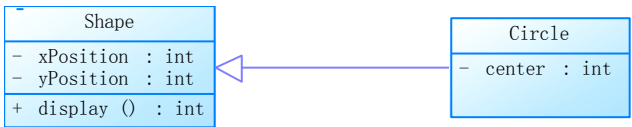


图 5 泛化示例

依赖用来在模型元素之间更广泛的依赖关系的建模。它通常被用在设计处理的早期来表达两个元素之间有某种关系，但这时还不知道它们之间的具体的关系。到设计后期，依赖会被原型化或者被更具体的关系来代替。用虚线箭头表示。

2. 类图示例

下面是一个学校中的类图。语义如下：学校有多个系，每个系有多个老师，一个老师可以讲多门课，一门课也可以有多个老师来讲，一个学生只能在一个学校上课，一个学校至少有一名学生，每个学生可以选多门课。类图示例见图 6：

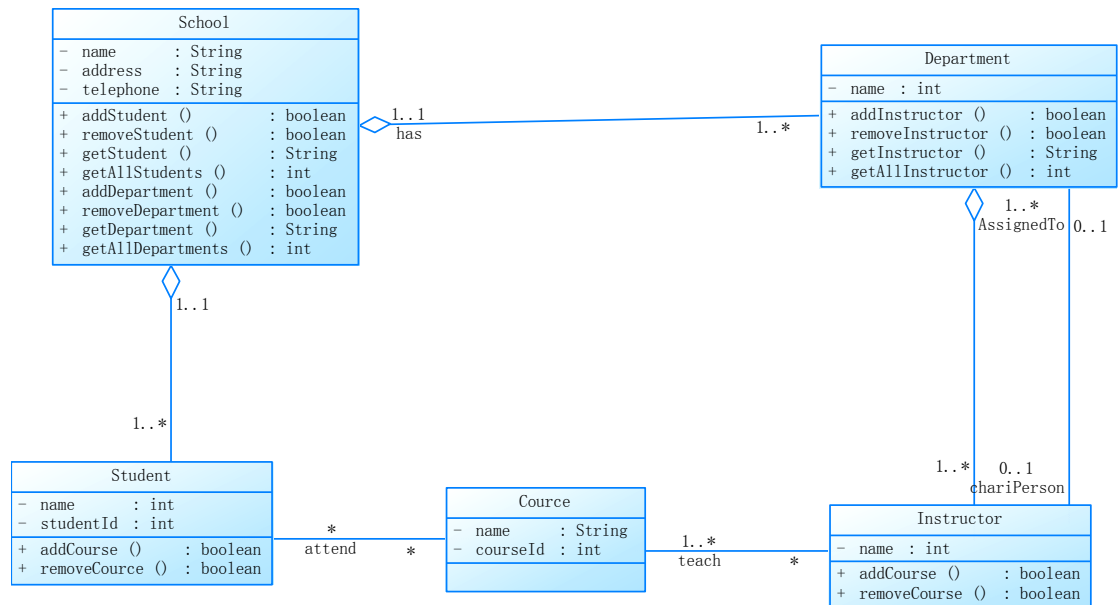


图 6 学校的类图