四. 类之间的六大关系

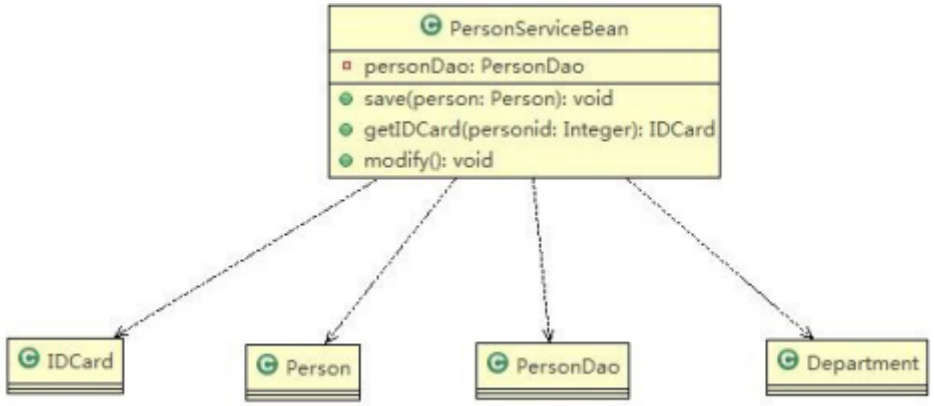
依赖、泛化（继承）、实现、关联、聚合与组合。

1.依赖

只要是在类中用到了对方，那么他们之间就存在依赖关系。如果没有对方，连

编绎都通过不了。

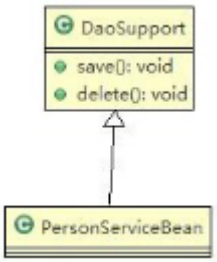
1) 类中用到了对方
2) 如果是类的成员属性
3) 如果是方法的返回类型
4) 是方法接收的参数类型
5) 方法中使用到



2.泛化(继承)

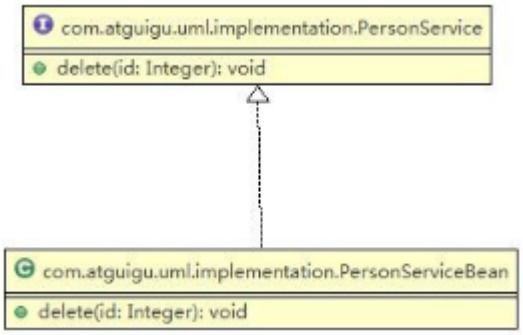
泛化关系实际上就是继承关系，他是依赖关系的特例

1) 泛化关系实际上就是继承关系
2) 如果 A 类继承了 B 类，我们就说 A 和 B 存在泛化关系



3.实现

实现关系实际上就是 A 类实现 B 接口，他是依赖关系的特例

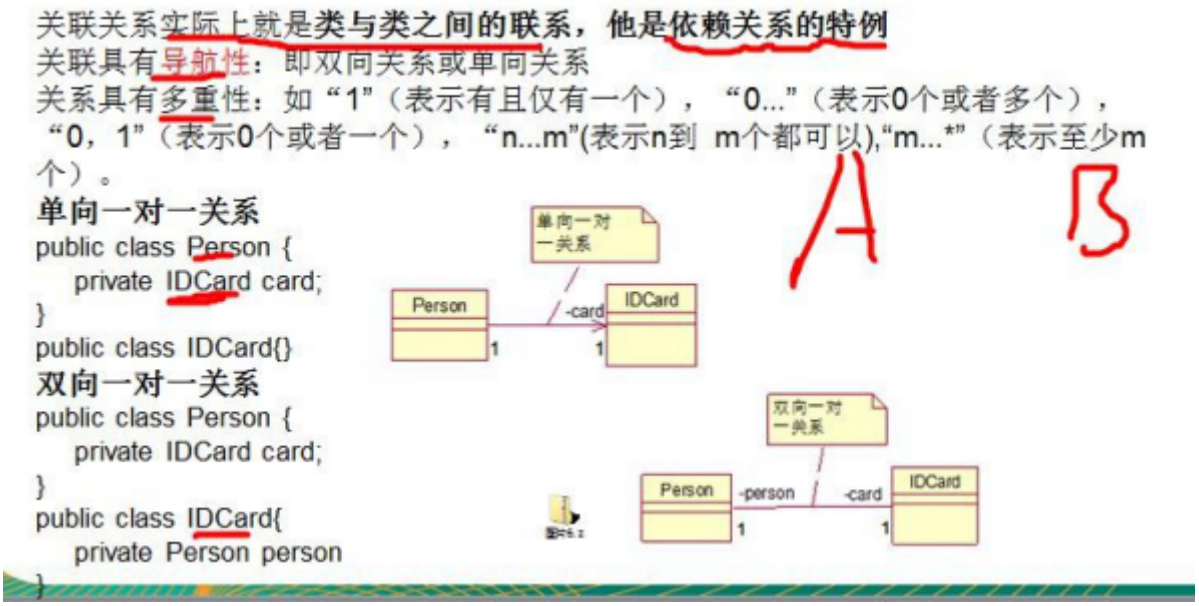


4.关联

关联关系实际上就是类与类之间的联系，他是依赖关系的特例

关联具有导航性：即双向关系或单向关系

关联具有多重性：



5.聚合

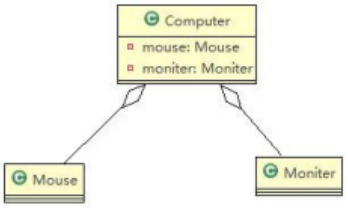
聚合关系（Aggregation）表示的是整体和部分的关系，整体与部分可以分开。聚合关系是关联关系的特例，所以他具有关联的导航性与多重性。

如：一台电脑由键盘(keyboard)、显示器(monitor)，鼠标等组成；组成电脑的各个配件是可以从电脑上分离出来的，使用带空心菱形的实线来表示：

```
public class Computer{
   private Mouse mouse;
   private Monitor monitor;

   public void setMouse(Mouse mouse){
      this.mouse = mouse;
   }

   public void setMonitor(Monitor monitor){
      this.monitor = monitor;
   }
}
```



7.组合

组合关系：也是整体与部分的关系，但是整体与部分不可以分开。

再看一个案例：在程序中我们定义实体：Person 与 IDCard、Head，那么 Head 和

Person 就是 组合，IDCard 和Person 就是聚合。

```
public class Person{
   private IDCard card;
   private Head head = new Head();
}
public class IDCard{}
public class Head{}
```

但是如果在程序中 Person 实体中定义了对 IDCard 进行级联删除，即删除 Person 时连同 IDCard 一起删除，那么 IDCard 和 Person 就是组合了.