**UML类图使用说明**

2018-02-27 09:33:58 [Lzh\_Thinking](https://me.csdn.net/leizhiheng320" \t "_blank) 阅读数 1338更多

分类专栏： [设计模式](https://blog.csdn.net/leizhiheng320/article/category/7469406)

版权声明：本文为博主原创文章，遵循[CC 4.0 BY-SA](http://creativecommons.org/licenses/by-sa/4.0/)版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/leizhiheng320/article/details/79383747>

*在我们学习设计模式时，最重要的一点是要明白类与类之间、类与接口之间的关系。这种关系我们可以使用UML（Unified Modeling Language 统一建模语言）类图来表示。当然，UML不仅仅可以用于java类图设计，还有很多其他作用，比如：UML用例图、UML时序图、UML状态/活动图、UML部署图、UML组件图和UML实体关系图。这里我们对UML类图的使用做一个说明。*

*这里会分两部分说明：1个是类和接口的UML表示，2是类之间、类与接口之间的6中相互关系*

***一、类和接口的UML表示***

***1、类的UML视图***

*比如我们定义一个Person类，定义如下：*

publicclass Person {

private String name;

public void setName(String name) {

this.name = name;

}

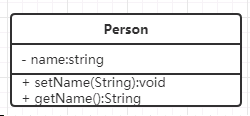
public String getName() {

return name;

}

}

*那么这个Person类的UML视图如下：*



我们可以看到，类图分为了三部分：

*（1）类名：类名在最上面*

*（2）属性列表*

*UML规定属性的表示格式为：*

*可见性 属性名：数据类型[= 缺省值]*

可见性符号：用于标识这个属性可不可以被外部类访问，可见性符号有四种：

*-：私有的，标识这个属性是私有属性，对应private*

*+ : 公有的，表示这个属性是共有属性，对应public*

*~ : 友好的，表示这个属性可以被同一个包中的类访问，对应package*

*# :受保护的，表示这个属性可以被子类访问，对应protected*

缺省值：可以没有

*（3）方法列表*

*UML规定方法的表示格式为：*

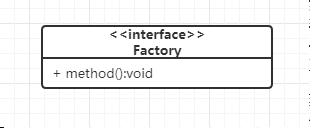
*可见性 方法名[参数列表]：返回值类型*

可见性：方法的可见性与属性的可见性是一样的

参数列表：参数列表中只需要标明每个参数的类型即可，不需要写明参数名称

*返回值类型：表示方法的返回值的类型，如果没有返回值，则是void*

*2、接口的UML表示*



接口和类的表示是有所区别的，首先，接口中没有属性列表，所以只分为两部分：接口名称和方法列表

接口名称：

*UML中相对于类名，接口名称的上面有一个<<interface>>的标识，表明这是一个接口而不是类。*

方法列表：

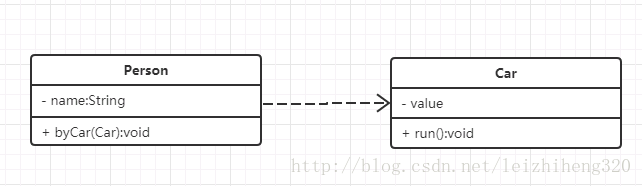
接口中的方法列表和类中的方法列表是一样的

***2、类之间的关系***

类与类之间并不是相互孤立的，他们之间存在多种关系：依赖关系、关联关系、组合关系、聚合关系、继承关系、实现关系。

*1、依赖关系（Dependency）*

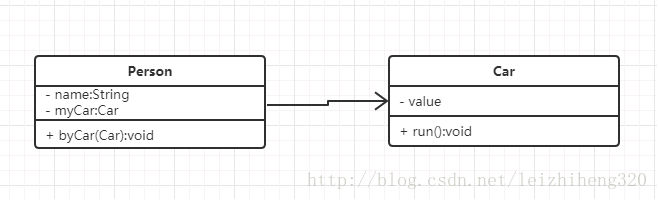
*依赖关系是类之间的最微弱的关系，是指一个类知道或者使用另一个类。这是典型的瞬时关系，依赖类和目标类进行简单的交互，依赖类并不维护目标类的对象，仅仅只是临时使用而已。在依赖关系下,目标类通常作为依赖类的局部变量，方法参数，静态方法调用时出现。依赖关系的两个类是由带箭头的虚线连接起来的，箭头由依赖类指向目标类，如下：*



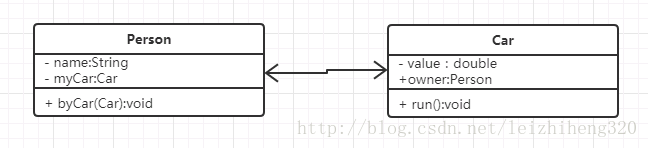
*2、关联关系*

*关联关系是比依赖关系更强的一种关系，是指一个类包含另一个类的对象，比如类A包含类B的对象，那么类B通常是作为类A的全局属性出现的，两者之间可以长期合作。具有关联关系的两个类使用带箭头的实线来连接。关联关系又分为单向关联和双向关联，他们的示意图如下：*

单向关联：

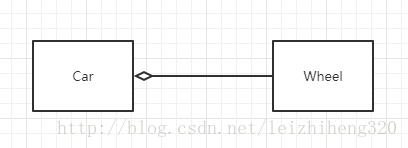


双向关联：



*3、聚合关系（Aggregation）*

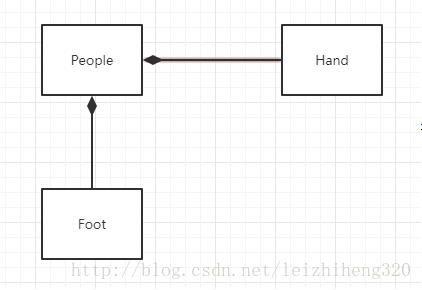
*聚合关系表示整体与部分的关系，在聚合关系中，成员对象是整体对象的一部分，但是成员对象可以脱离整体对象而存在。在UML中聚合关系中的类使用带空心菱形箭头的实线连接。菱形箭头指向整体类，关系示意图如下：*



*Wheel是Car的组成部分，但是Wheel又可以脱离Car而单独存在。*

*4、组合关系（Composition）*

组合关系也是部分与整体的关系。成员对象是整体对象的一部分，并且成员对象不能脱离整体对象而存在。这也是组合关系与聚合关系之间的区别之处。在组合关系中，两个类以带实心菱形箭头的实线连接。如下：



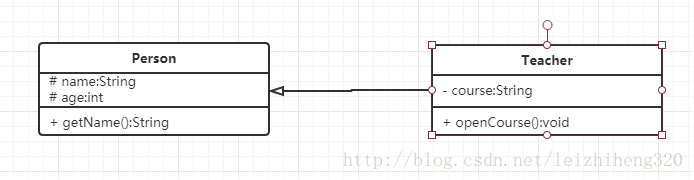
手和脚是人的组成部分，手和脚是不能脱离人体而单独存在的。

*5、泛化关系（Generalization）*

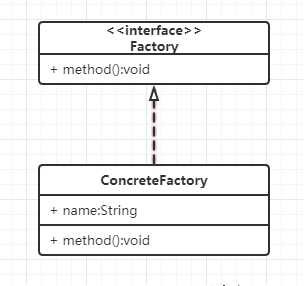
泛化关系用于描述父类和子类之间的关系，父类又称为基类和超类，子类又称作派生类。

繁华关系分为两种：继承关系和实现关系

继承关系是是两个具体类之间的泛化关系，两个类之间使用带空心三角箭头的实线连接，如下：



实现关系是指类和接口之间的泛化关系，类和接口之间使用带空心三角箭头的虚线连接，如下：



        在学习类与类之间的关系时，最重要的是要清楚每种关系的含义和类图。

        类与类之间的关系可以说是两两分类的：依赖关系和关联关系、聚合关系和组合关系 、泛化关系中的继承和实现关系。

同一类关系要表达的类与类之间的关系比较接近，而且类图也比较相似，这有助于我们记忆。