





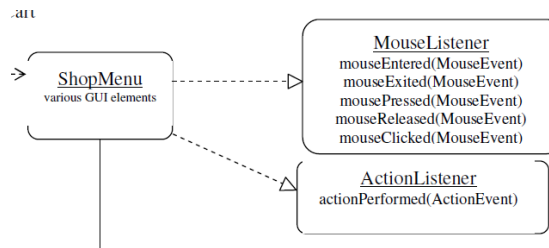
TABLE 12-1 UML Relationship Symbols

Relationship	Symbol	Line Style	Arrow Tip
Inheritance		Solid	Triangle
Interface Implementation		Dotted	Triangle
Aggregation		Solid	Diamond
Dependency		Dotted	Open

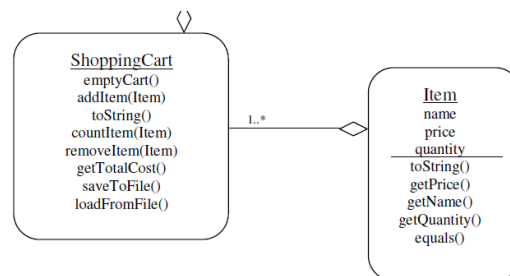
Inheritance exists when a class extends another class. For example if Car extends Vehicle, Car is-a Vehicle, and the relationship would be diagrammed like this:



Interface implementation exists when a class implements an interface. For example, the ShopMenu class implements MouseListener and KeyListener, so the relationship is diagrammed like this:



Aggregation exists when a class has instance fields which are other classes. For example, ShoppingCart has an ArrayList of Item objects, so it has-a Item. In fact, since it can have one or more Item, you add the 1..* notation along the line:



Dependency is when a class uses objects of another class, but doesn't own them (as instance fields). For instance, the Shop class instantiates a ShopMenu object, but it is not an instance field:

