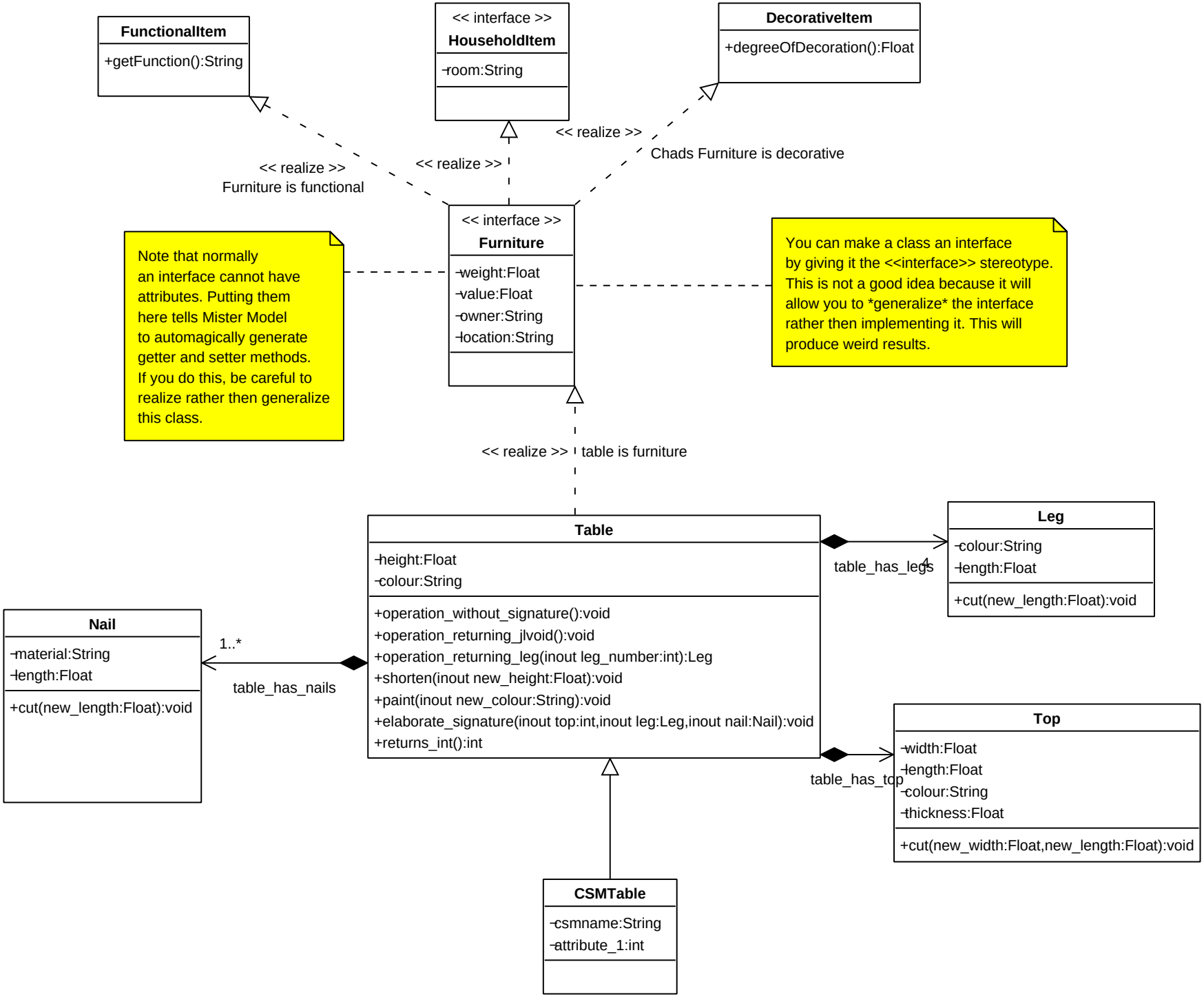
FunctionalItem
+getFunction():String
<< interface >>
HouseholdItem
-room:String
DecorativeItem
+degreeOfDecoration():Float
<< realize >>
Chads Furniture is decorative
<< realize >>
Furniture is functional
<< realize >>
<< interface >>
Furniture
-weight:Float
-value:Float
-owner:String
-location:String
Note that normally an interface cannot have attributes. Putting them here tells Mister Model to automagically generate getter and setter methods. If you do this, be careful to realize rather then generalize this class.
You can make a class an interface by giving it the <<interface>> stereotype. This is not a good idea because it will allow you to *generalize* the interface rather then implementing it. This will produce weird results.
<< realize >> table is furniture
Table
-height:Float
-colour:String
+operation_without_signature():void
+operation_returning_jlvoid():void
+operation_returning_leg(inout leg_number:int):Leg
+shorten(inout new_height:Float):void
+paint(inout new_colour:String):void
+elaborate_signature(inout top:int,inout leg:Leg,inout nail:Nail):void
+returns_int():int
table_has_legs
4
Leg
-colour:String
-length:Float
+cut(new_length:Float):void
Nail
-material:String
-length:Float
+cut(new_length:Float):void
1..*
table_has_nails
table_has_top
Top
-width:Float
-length:Float
-colour:String
-thickness:Float
+cut(new_width:Float,new_length:Float):void
CSMTable
-csmname:String
-attribute_1:int