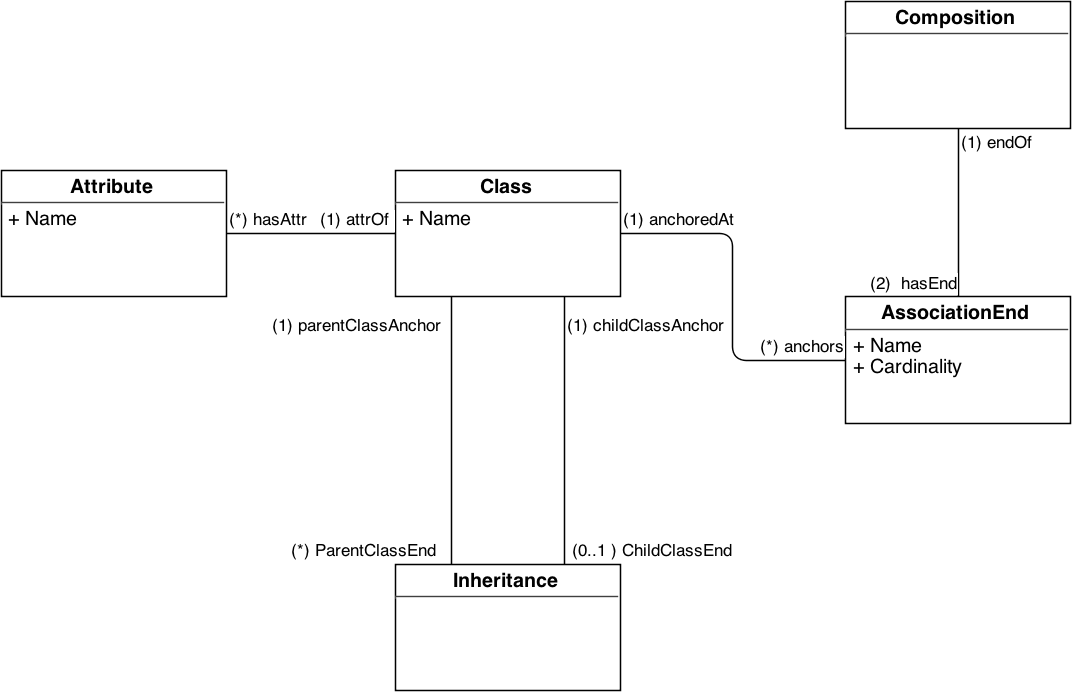
**Solution 1:**



Explanation:

I extended the existing diagram by adding an “Inheritance” class. This class is associated with the “Class” class, by two associations.

* First association: [parentClassAnchor (1) - ParentClassEnd (\*)]

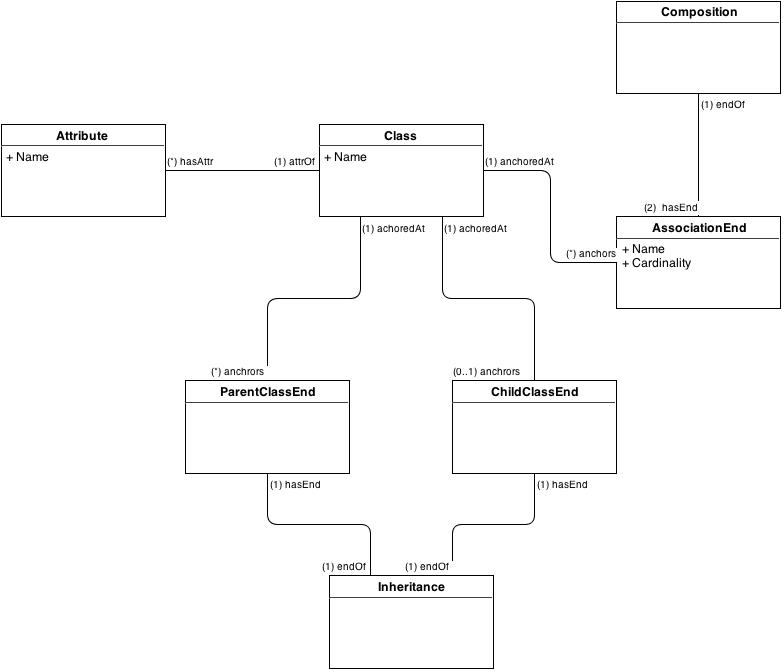
This tells us that multiple classes can extend from a single class. Therefore, a single class can have many “ParentClassEnd”s of an inheritance relationship.

* Second association: [childClassAnchor (1) - childClassEnd (0..1)]

This tells us that there is no multiple inheritance allowed. Therefore, a single class can have either 0 or 1 “ChildClassEnd” of an inheritance relationship.

***Note****:* Initially, simile to Composition relationship, I decided to create two more “InheritanceParentEnd” and “InheritanceChildEnd” classes, but that seemed to be overkill here. There are neither names nor cardinalities associated with “InheritanceParentEnd” and “InheritanceChildEnd”; therefore creating two separate classes does not seem to be a good refinement. Also, “Inheritance” has one-one relation with “InheritanceParentEnd” and “InheritanceChildEnd”. Those could be clubbed together.

*My earlier diagram looked something like:*

**

**(DISCARDED VERSION)**

Inheritance Specific Constraints:

1. Two classes cannot have both composition and inheritance relationship among each other. i.e*. It is not possible that if “class A extends B”, then there exists a association relationship among A and B*.
2. Parent class and child Class of an inheritance relationship should be different. i.e. *Class A should not extend “Class A”*.

Other Constraints (From previous assignment):

1. Name attribute of “Class” class should be unique.
2. Different classes can have same attribute names. However, attributes insides a class should be named uniquely.
3. Cardinality of at least one of the AssociatedEnd’s of a composition should be greater than zero i.e. No composition should be have (0,0) cardinality.

**SOLUTION 2:**

**% Inheritance related Constraints.**

*inheritance\_and\_composition* :- inheritance(\_,P,C), associationEnd(\_,\_,\_,CompId,P), associationEnd(\_,\_,\_,CompId, C), writeln('Inheritance and composition co-exist among two classes.').

*parent\_and\_child\_same* :- inheritance(\_,P,P), writeln('Parent and child of inheritance relation defined to be same.').

**% Homework-3 related constraints.**

*class\_name\_not\_unique* :- class(\_,C1), aggregate\_all(count, class(\_,C1), COUNT), COUNT\=1, write('Class name not unique.').

*attribute\_name\_not\_unique* :- attribute(\_,Name,C), aggregate\_all(count, attribute(\_,Name,C), COUNT), COUNT\=1, write('Attribute name for a class is not unique.').

*zero\_zero\_cardinality* :- composition(C), aggregate\_all(count, associationEnd(\_,\_,0,C,\_), COUNT), COUNT == 2, write('A composition has zero-zero cardinality').

**% Validate all constraints.**

*all\_constraints\_valid* :- not(parent\_and\_child\_same), not(inheritance\_and\_composition), not(class\_name\_not\_unique), not(attribute\_name\_not\_unique), not(zero\_zero\_cardinality) .