Inheritance Pattern Practice

1 Basic Structures

For each of the three patterns, design the database tables for the hierarchy in Figure 1. In addition, specify exactly how keys would get allocated. Pay careful attention to the relationships between the tables.

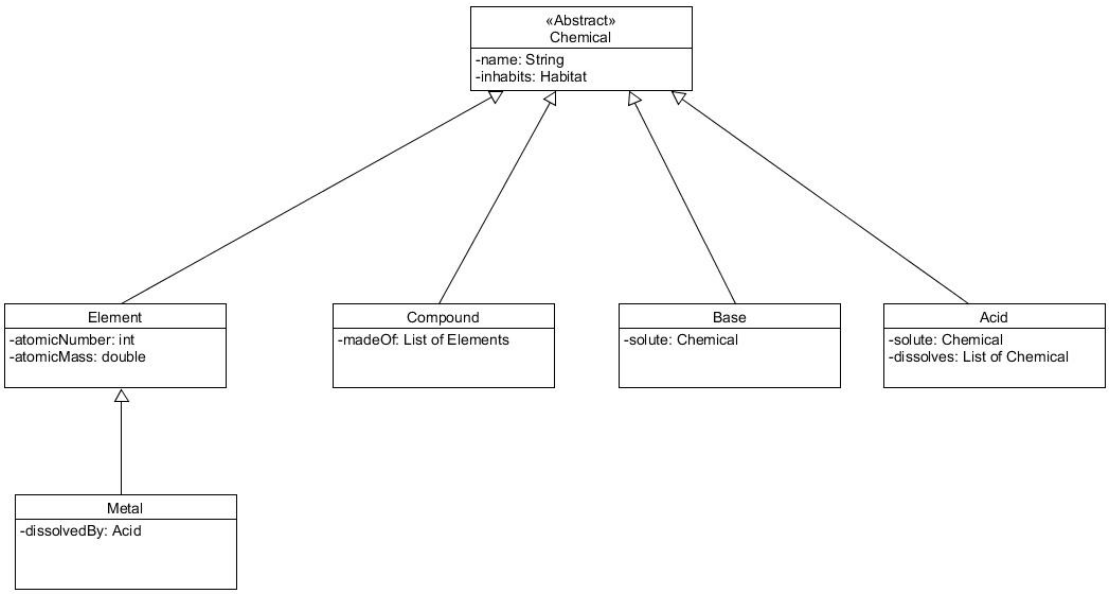

Figure 1: Chemical Example

2 Changing the Hierarchy

Each of the following questions refers to the diagram in Figure 2. Do not process them on top of each other - for each, start with the given diagram. For each situation, specify exactly how the tables would change for each pattern

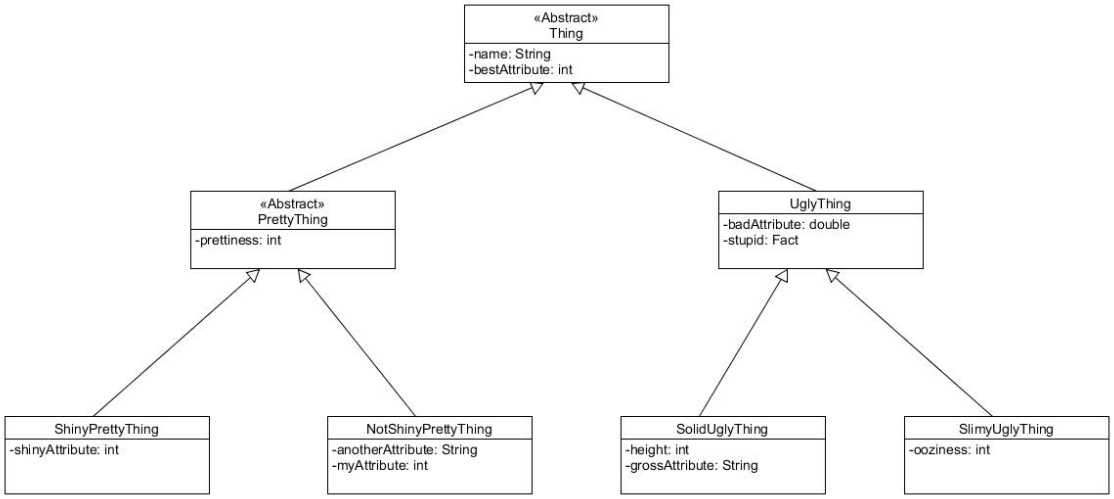

Figure 2: A Very Silly Example

1. Suppose we decided that all pretty things needed the anotherAttribute instance variable, so it moved from the NotShinyPrettyThing class to the PrettyThing class.
2. Suppose we decided that all ugly things needed the grossAttribute instance variable, so it moved from the SolidUglyThing class to the UglyThing class.
3. Suppose we decided that only pretty things needed the bestAttribute, so we moved it down from Thing to PrettyThing.
4. Suppose we decided that only ugly things needed the bestAttribute, so we moved it down from Thing to UglyThing.
5. Suppose we wanted to add an integer instance variable named anyAttribute to all Things.
6. Supposed we wanted to add an integer instance variable named anyAttribute to all UglyThings.
7. Suppose we wanted to delete bestAttribute from Thing
8. Suppose we wanted to delete the stupid Fact from UglyThing
9. Supposed we wanted to add a double instance variable named shinyAttribute to SlimyUglyThing.

3 Other Things to Think About

1. Suppose you knew that there were going to be a LOT more instances of one class than the others. How would that affect the choice of pattern? Does it matter how many instance variables are in that class?

2. Do you have to choose the same pattern for all of your system?

3. Do you have to choose the same pattern for all of one inheritance hierarchy?

4. If you chose to mix patterns, how would you document that choice so people who came after you could easily figure out what you did?

5. If your database is going to be small (not enterprise), which pattern would you choose? What criteria would you use for picking?

6. If we don't expect many changes to the hierarchy, how would each of these characteristics would affect your choice of pattern? (It is OK to say it is irrelevant)

   (a) The number of variables in the entire hierarchy.

   (b) The ratio of the number of variables high in the pattern versus the number of variables towards the leaves of the pattern.

   (c) The width of the inheritance tree (maximum classes in one level)

   (d) The depth of the inheritance tree (maximum length of a path from a root to a leaf)

   (e) The number and position of abstract classes in the hierarchy