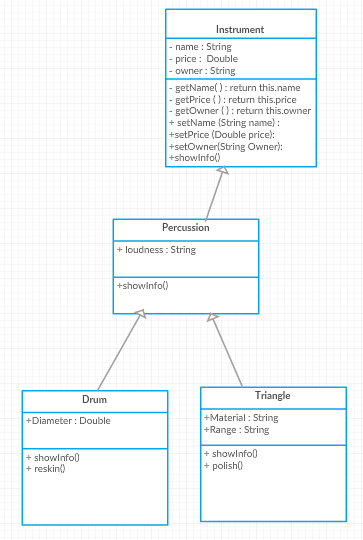
Design of the Instrument Inheritance for Question 3.



First step of designing the layout for this question was to create a UMl diagram showing the inheritance from the Instrument class downwards.

I only made a UML diagram for the percussion class as the other classes would follow this layout.

Having the attributes that were common to every instrument in the question in the Instrument class.

Then the more specialised attributes for each type of instrument inside that instrument types class.

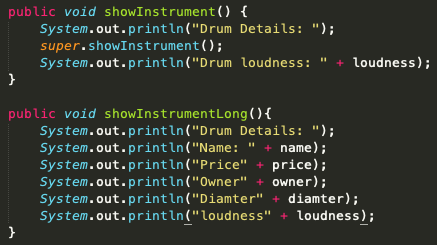
+Hit()

In the percussion example I used loudness as that is something that is is more specialised to the percussion type of instrument.

In this case the other instrument classes could also have a loudness attribute however as this is more an example then anything I hope it counts! In a real situation it would be something a bit more specific to that exact type of instrument.

Then the two individual instrument classes inherited from the percussion class where the Drum and Triangle classes. Both of these contained the final attributes that are completely individual.

This is along with methods that are unique to these instruments. Reskinning is unique to only drums so it can’t be inherited from the percussion class as you can’t reskin a triangle. However as you hit both the Drum and the Triangle the hit method can be inherited from the percussion class.

The individual classes also had their own showInfo methods as each layer down would call the upper levels. 

So you could use Super.showInfo in the Drum class to show the Percussion classes data. And same from the Percussion class calling super.showInfo() would show the data from the Instrument class.

This means that from the Drum class you wouldn't have to type out the showInfo() method as a print for every attribute instead you would call the showInfo() for the level above and then print out anything extra have added at the current layer.

In the code screenshot above the difference in the two methods is that use uses super.showInfo to have the higher level class print its own data then prints it’s after. This compared to the bottom method in which each attribute has to be printed out individually.

Also if using the second method then each of the individual instrument classes would need the same section of code. Especially with more attributes this would make each class have a massive print section. With each section having its own showInfo() it means that the lower classes don’t have to repeat printing code.