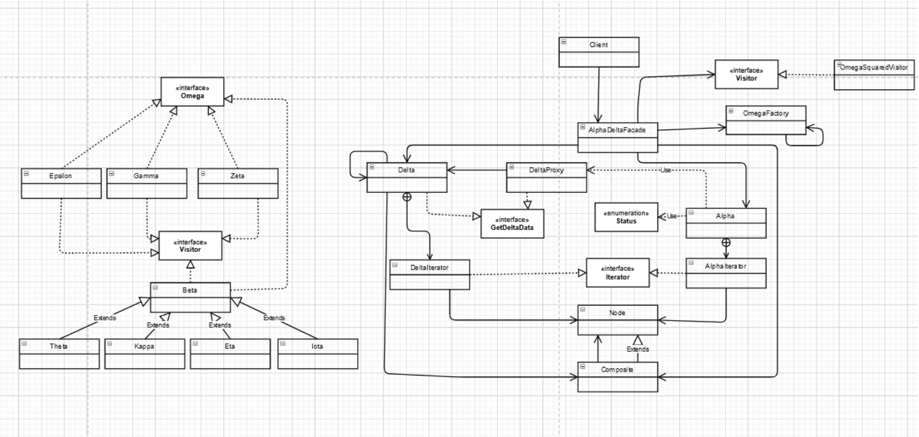
AOOSD - Assessment II

Eoin Wilkie

December 18, 2020



My understanding of this design is that Alpha is used to manipulate Gamma objects and Delta is used to manipulate Beta, and other objects. Although Epsilon and Zeta are mentioned, it is my understanding that they are to be stored in Delta by means unrelated to this assignment.

Beta and Gamma need a commonality and as I am unsure what Epsilon and Zeta require I decided to create the interface, Omega, for the four. This hierarchy would be expanded on if more information was known. Beta has sub-classes Eta, Iota, Kappa and Theta in order to satisfy an interpreted requirement.

Both Delta and Alpha have similar functionality. Delta is a singleton and as such it stores its data, alpha does not, rather it manipulates a collection passed to it. I should have implement synchronised access to Delta, as my understanding is that it could be used elsewhere to store the unused Epsilon and Zeta classes. Both Alpha and Delta use a composite iteration pattern to implement their respective collection.

The client class has access to a facade which manipulates Alpha and Delta, the facade also includes a visitor to update values stored in Alpha and Delta. The client can add to either Alpha or Delta through an overloaded add method. The facade creates objects with an factory pattern. If the user adds a Beta, it will be added to Alpha, which will then pass it to Delta through a proxy.