GALWAY-MAYO INSTITUTE OF TECHNOLOGY

Department of Computer Science & Applied Physics

Advanced Object-Oriented Design Principles & Patterns (2020) ASSESSMENT I

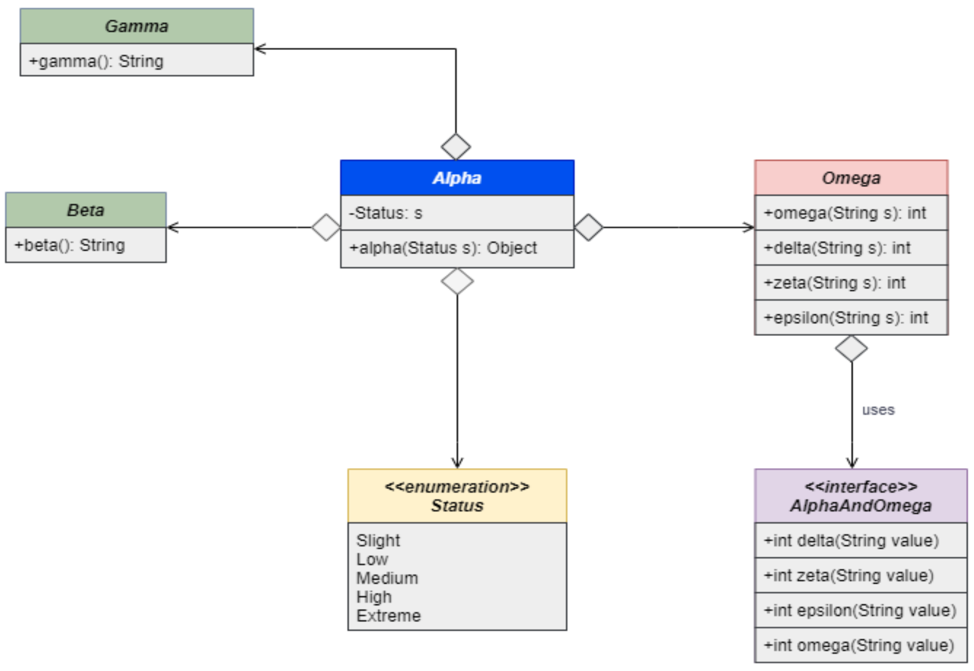
John Shields – G00348436

README

DESIGN RATIONAL

My redesigning of the application made me realize I first had to change the Alpha class meaning I had to change the other classes in order to redesign it completely. Starting with making a class called Omega, I took the methods from the classes Delta, Zeta and Epsilon and put them into the Omega class. These methods had similar functions therefore it was unnecessary to have more than one class managing them. Having this done this allowed me to alter the switch statement. I could now put all the days into one case. This makes it much more simplified. In order to make this all work I then had to create an Interface called AlphaAndOmega. This Interface allows to implement the methods into the Omega Class. With this redesigning in place I can now discard the classes Delta, Zeta and Epsilon as Omega and AlphaAndOmega are now doing all their work for them. I have reduced the classes from 6 to 4 with one interface and one enumeration. This makes for a much more defined design.

The following UML diagram shows a suite of classes my redesign of the application:



- The method alpha() returns an instance of Object based on a Status and DayOfWeek.
- Either of the classes Beta and Gamma can be used to handle a Status of Slight, Low or Medium and there are many other potential related types than can handle similar requests.
- Depending on the DayOfWeek the remaining enumerations of Status are handled by instances of Omega and an unbounded number of related types. Delta, Zeta and Epsilon are orchestrated together as Omega to create complex types.
- Omega uses the interface AlphaAndOmega to implement the methods of Delta, Zeta and Epsilon.

END OF DOCUMENT