**Class – A\_Element**

The A\_Element class is an abstract class that is extended by the Mesh class and the Slice class. The A\_Element class has a random object that is used to store the random object its constructor receives when one of its inheriting classes is constructed. A\_Element also has an AxisDescriptor (AxisDescriptor class will be discussed later in the papr) object to store the AxisDescriptor that is passed in at construction time as well. A\_Element has a constructor and an override of toString(). The constructor checks the objects being passed in as parameter with a method from Assert class (will be discussed later in the paper) to make sure they are not null. It then assigns the parameter objects to its own objects to store them. The toString() method returns the randoms hashcode and the AxisDescriptor.

**Class – Assert**

Class Assert is an abstract class in order to take on a singleton role. Assert class is used throughout the program and defines all the assertion functionality. Assert does not store any variables or objects. Assert has isTrue method, nonnull method, nonnullempty method, optional method, positive method and range method.

The isTrue method deals with Boolean values and is passed in Boolean lists. Its first calls nonnull (will be covering next) to make sure it’s values are not null. After checking to see if the values are null or not isTrue than checks each value in the list and if a value is not true throw’s an AssertionError exception stating which value out of all the values is false. Otherwise it does nothing and drops out of the method.

The nonnull method receives object lists as its parameters and checks them to see if anything is null. First nonnull checks to see if the list itself is null and if so then it throw’s an exception stating it is null. If it passes that check then nonnull goes through and checks each value of the list to see if it is null or not. If it finds a null value it throws’ an exception AssertionError, stating which argument out of the total number of arguments, is null. If nonnull does not find any null values then it drops out of the method and continues on.

Nonnullempty checks to see if the list of strings it receives as a parameter is null or empty. The first thing it does is to check and see if the list itself is empty. If it is empty an exception is thrown stating it is null. If the list is not null than the method goes through and checks each value in the list to see if there is any that are null or empty. If it finds a value that is null then an exception is thrown stating which argument out of all the arguments is null. If it finds a value that is empty an AssertionError exception is thrown stating again which argument out of all the arguments is empty. If the method doesn’t find any null or empty values then it drops out of the method.

The next method is the optional method it check on optional arguments within the program. It takes an object array as its parameter. The first thing it does is to send that array to nonnull to make sure it’s not null. If it gets it back then it goes through and checks each value in that array to check and see if it is less than or equal to zero. If a value is less than or equal to zero then an AssertionError exception is thrown showing which value, out of all the values and stating its value. If not values are less than or equal to zero it drops out of the method.

The positive method checks to see if the values, in the list that is passed in to parameters, is positive or not. The method first sends the list to nonnull and then checks each value to see if it is less than or equal to zero. If a value is less than or equal to zero then an AssertionError exception is thrown stating which value and what its value is. If no values are found that are less than or equal to zero then it drops out of the method.

The range method checks to see if a value is within a range. It receives as its parameters, a double value, a double min and a double max. Then it checks to see if the value is less than the min or greater than the max. If the value is less than the min an AssertionError exception is thrown stating that the value is less than the min. If the value is greater than the max an AssertionError exception is thrown stating that the value is greater than the max. If the value is within the range it drops out of the method.