

Specifications: WeightedTotalSystem

In addition to the obvious specifications illustrated in the UML class diagram and the specifications for the parent class and interface, the `WeightedTotalSystem` class must satisfy the following specifications.

1. public methods must not have any side effects. That is, they must not change the parameters that they are passed in any way (e.g., the `List` that is passed to the `calculate()` method must not be changed in any way) and they must not change attributes that are not “owned” (i.e., attributes that are aliases) in any way (e.g., the `Map` that is passed to the constructor must not be changed in any way).
2. You may assume that the `calculate()` method is passed a `List` that does not contain any `null` elements.
3. You may assume that all of the weights are non-negative.
4. If the weight for a particular `Score` is unspecified (i.e., `null`) then a weight of `1.0` must be used (even if the `weights` `Map` is `null`). Note: The `Missing` class has a method that can be used to accomplish this.
5. If the weight for a particular `Score` is less than `1.0` then a weight of `1.0` must be used. Note: The `Missing` class has a method that can be used to accomplish this.
6. If the value of a particular `Score` is missing (i.e., `null`) then a value of `0.0` must be used. Note: The `Missing` class has a method that can be used to accomplish this.
7. The `calculate()` method must calculate the weighted total of the `List` of `Score` objects it is passed.
 - 7.1. If the `List` is `null` then it must throw a `SizeException`.
 - 7.2. If the `List` is empty then it must throw a `SizeException`.
 - 7.3. Otherwise, it must return a `Score` object with the given key and a value equal to the weighted total of the `Score` objects in the `List`.
 - 7.3.1. The weight for each element must be obtained from the `Map` using the key for that element.
8. The default constructor must (directly or indirectly) initialize the `weights` `Map` to `null`.