Specifications: `WeightedTotalStrategy`

In addition to the obvious specifications illustrated in the UML class diagram and the specifications for the parent class and interface, the `WeightedTotalStrategy` 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. The `calculate()` method must calculate the weighted total of the `List` of `Grade` objects it is passed.
   2.1. You may assume that the `calculate()` method is passed a `List` that does not contain any `null` elements.
   2.2. If the `List` is `null` then it must throw a `SizeException`.
   2.3. If the `List` is empty then it must throw a `SizeException`.
   2.4. Otherwise, it must return a Grade object with the given key and a value equal to the weighted total of the `Grade` objects in the `List`.
      2.4.1. The weight for each element must be obtained from the `Map` using the key for that element.
         2.4.1.1. If the `weights Map` is `null` than a weight of `1.0` must be used.
         2.4.1.2. If the weight for a particular `Grade` is unspecified (i.e., `null`) then a weight of `1.0` must be used. Note: The `Missing` class has a method that can be used to accomplish this.
         2.4.1.3. If the weight for a particular `Grade` is less than `0.0` then a weight of `0.0` must be used.
      2.4.2. If the value of a particular `Grade` 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.
3. The default constructor must (directly or indirectly) initialize the `weights Map` to `null`.