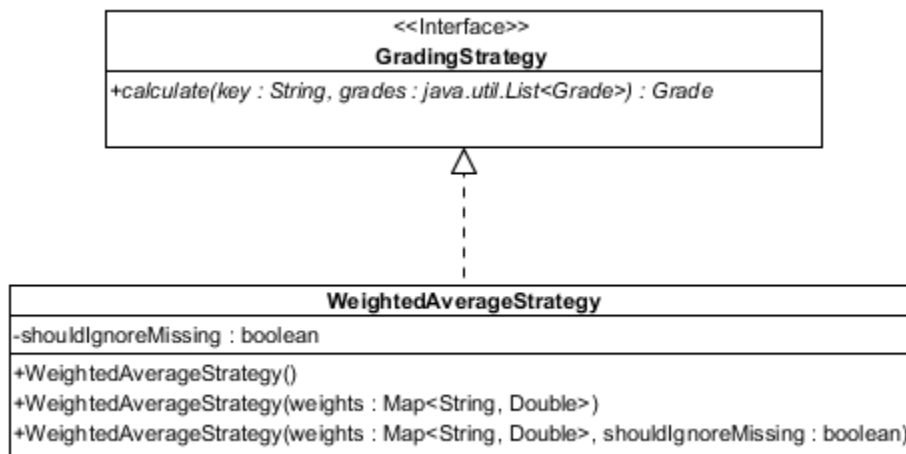




Specifications: WeightedAverageStrategy

In addition to the obvious specifications illustrated in the following UML class diagram



the `WeightedAverageStrategy` class must satisfy the following specifications.

1. The `calculate()` method must not have any side effects. That is, it must not change the `List` that it is passed or any of the values in the `List`.
2. You may assume that the `calculate()` method is passed a `List` that does not contain any null elements. Note: This does not mean that the `List` does not contain **missing** grades
3. You may assume that all of the weights are non-negative.
4. The `calculate()` method must handle null weights.
 - 4.1. If the weights `Map` is null then each `Grade` must be weighted equally.
 - 4.2. If the weight for a particular `Grade` is null then the weight for that `Grade` is said to be unspecified and a value of `0.0` must be used.
5. The `calculate()` method must calculate the weighted average of the `List` of `Grade` objects it is passed.
 - 5.1. It must account for missing values in one of two ways, depending on the value of the `shouldIgnoreMissing` attribute.
 - 5.1.1. If `shouldIgnoreMissing` is true then missing values should be ignored. In other words, if there are 10 elements and 2 are missing, the calculation should be performed as if there are only the 8 non-missing elements.
 - 5.1.2. If `shouldIgnoreMissing` is false then missing values should be treated as `0.0`.

- 5.2.If the List is null then the weighted average must be 0.0.
- 5.3.If the List is empty (i.e., has no elements after appropriately accounting for missing values as described above) then the weighted average must be 0.0.
- 5.4.If the List is not null and not empty then the weighted average must be calculated as described in the Glossary.
- 5.5.If all of the weights are 0.0 (i.e., the denominator in the above expression is 0.0) then the weighted average must be 0.0.
6. The default constructor must construct a WeightedAverageStrategy object with a null weights Map that ignores missing values.
7. The one-parameter constructor must construct a WeightedAverageStrategy object that ignores missing values

An Example with No Weights that Illustrates the Handling of Missing Values

Suppose there are no weights and the List contains Grade objects with the following value attributes: 5.0, 8.0, null, 2.0, and null. If the WeightedAverageStrategy is ignoring missing values, then the calculate() method must return a LeafGrade object with a value attribute of:

$$(5.0 + 8.0 + 2.0) / 3 = 15.0 / 3 = 5.0$$

On the other hand, if the WeightedAverageStrategy is not ignoring missing values, then the calculate() method must return a LeafGrade object with a value attribute of:

$$(5.0 + 8.0 + 0.0 + 2.0 + 0.0) / 5 = 15.0 / 5 = 3.0$$

An Example with Weights

Suppose the List contains Grade objects with the following value attributes: 5.0, 8.0, 2.0, and the corresponding weights are 30.0, 30.0, 40.0. Then the calculate() method must return a LeafGrade object with a value attribute of:

$$\begin{aligned} & (5.0 \cdot 30.0 + 8.0 \cdot 30.0 + 2.0 \cdot 40.0) / (30.0 + 30.0 + 40.0) \\ & = (150.0 + 240.0 + 80.0) / (30.0 + 30.0 + 40.0) \\ & = 470.0 / 100.0 = 4.7 \end{aligned}$$