



Specifications: Grade

In addition to the obvious specifications illustrated in the UML class diagram, the `Grade` class must satisfy the following specifications.

1. `Grade` objects must be immutable.
2. If a constructor is passed a `key` that is `null` or empty (i.e., `""`) then the constructor must throw an `IllegalArgumentException`.
3. The `Grade(String key)` constructor must construct a `Grade` object with a `value` attribute of `0.0`.
4. The `compareTo(Grade other)` method must return the result of comparing `this.value` and `other.value` accounting for missing (i.e., `null`) values appropriately.
 - 4.1. If `this.value` is `null` and `other.value` is non-`null` then it must return `-1`.
 - 4.2. If `this.value` is `null` and `other.value` is `null` then it must return `0`.
 - 4.3. If `this.value` is non-`null` and `other.value` is `null` then it must return `1`.
 - 4.4. If both `this.value` and `other.value` are non-`null` then it must return the result of calling `compareTo()` on `this.value` and passing it `other.value` (though it need not be implemented this way).
5. The `toString()` method must return a `String` representation of the `Grade` object.
 - 5.1. If the `value` attribute is not `null` then the `String` must contain the `key` attribute, followed by the `String` literal `:"`, followed by a single space, followed by the `value` attribute (in a field of width 5 with 1 digit to the right of the decimal point).
 - 5.2. If the `value` attribute is `null` then the `String` must contain the `key` attribute, followed by the `String` literal `:"`, followed by a single space, followed by the `String` literal `" NA"` (which is also a field of width 5).

Note that, while `null` `key` attributes are invalid (i.e., every `Grade` object must have a non-`null`, non-empty `key` attribute), `null` `value` attributes are valid (and are used to indicate that the `Grade` is missing).