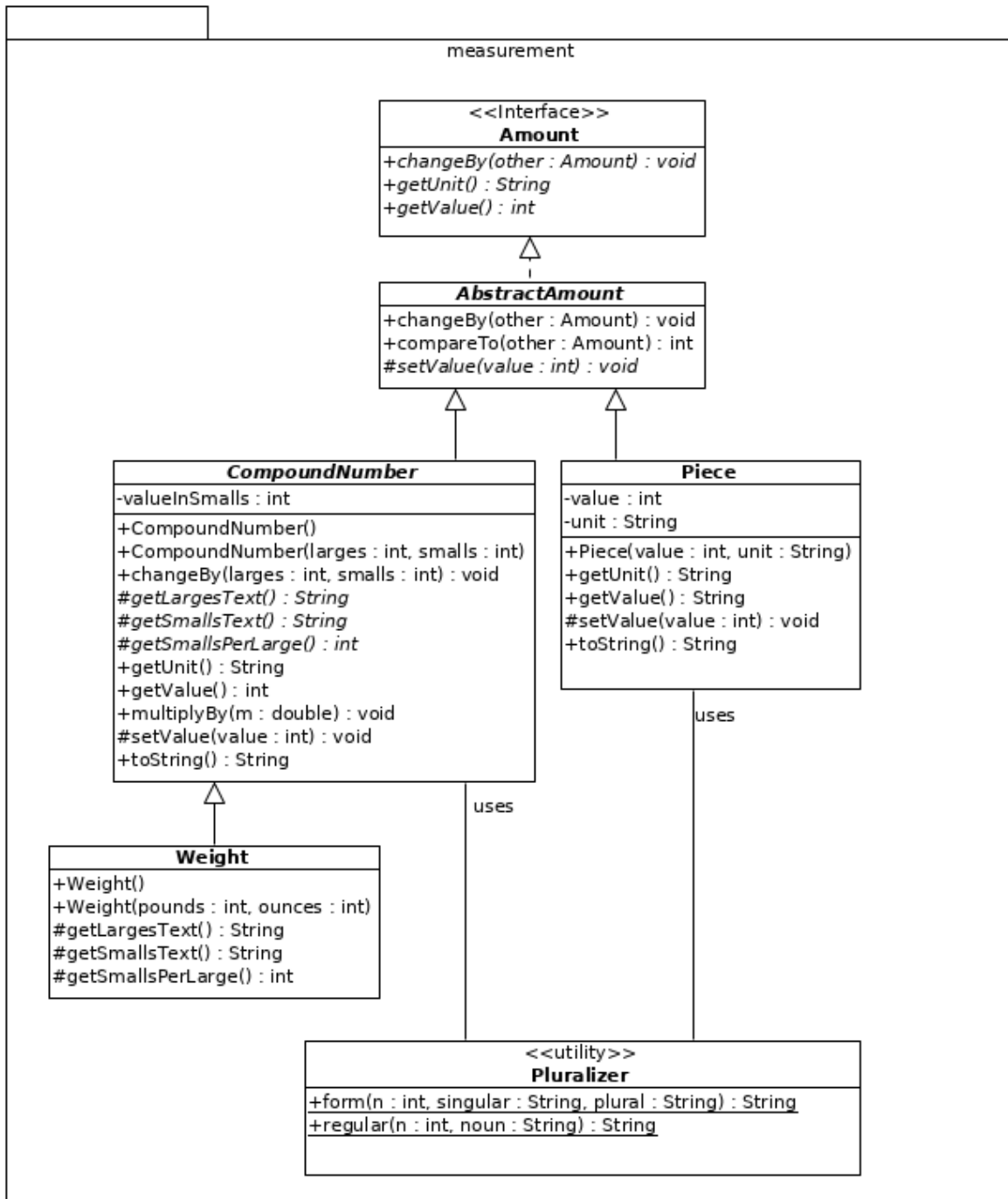




A Summary of the Commitments I made at the Planning Meeting for Sprint 1

Class Diagram

For this sprint, I committed to implementing the interfaces and classes illustrated in the following UML class diagram:



Specifications

In addition to the specifications that are obvious from the UML class diagram, the team agreed to the following.

Pluralizer

The `Pluralizer` class is a utility class that can be used to create plural forms of words from singular forms. The `form()` method is passed both the number, the singular form and the plural form, and returns the singular form when the number is 1 and the plural form otherwise. One might use it as follows:

```
System.out.println(length + " " + Pluralizer.form(length, "inch", "inches");
```

The `regular()` method is used for regular pluralizations (i.e., those that just involve appending an “s”). One might use it as follows:

```
System.out.println(weight + " " + Pluralizer.regular(weight, "pound");
```

Amount

The `Amount` interface describes the capabilities of the amount of something that might get used in an intelligent kitchen. For example, the amount of butter to use in a recipe, the amount of bread in the pantry, or the amount of ground beef in the kitchen.

Piece

The `Piece` class is an encapsulation of an `Amount` that is measured by the individual piece. For example, in some contexts, bread is measured by the slice. So, a `Piece` object for the bread in a sandwich would have a `value` attribute of 2 and a `unit` attribute of "slice".

CompoundNumber

The `CompoundNumber` class is a partial encapsulation of a quantity that is expressed in terms of two units (e.g., 6 feet, 2 inches or 3 pounds, 12 ounces). This class may be generalized in the future to an arbitrary number of units, however, it is currently designed for only two units, called the “larges” (e.g., feet or pounds in the previous example) and the “smalls” (e.g., inches and ounces in the previous example). The team agreed that it must only store the `valueInSmalls` (e.g., 74 inches or 60 ounces in the previous example) and calculate the two units when needed (e.g., in the `toString()` method).

In addition to the obvious specifications, this methods in this class must satisfy the following:

`multiplyBy()` – Must truncate the result of the multiplication.

`toString()` - For the two examples above, must return the `String` objects:

```
"6 feet, 2 inches"
```

and

```
"3 pounds, 12 ounces"
```

(without the quotes) respectively. It must use the `getLargestText()`, `getSmallsText()`, and `getSmallsPerLarge()` methods. It must also use the `Pluralizer` class. The `getUnit()` method must return the value returned by `getSmallsText()` and the `getValue()` method must return the value in smalls .

Weight

The `Weight` class is a `CompoundNumber` that handles quantities measured in pounds and ounces. The text for pounds is "pound" and the text for ounces is "ounce".