



A Summary of the Discussion at the Planning Meeting for Sprint 2

Design Defects

The `AbstractAmount` class contains a `compareTo()` method, but does not implement the `Comparable` interface. Hence, some other members of the team were unable to use the sorting methods (e.g., in the `java.util` package) that are passed an array or list of `Comparable` objects.

In addition, as the classes are currently designed and implemented, the `changeBy()` and `compareTo()` methods both take `Amount` objects as parameters. This means that, for example, a `Weight` object can be compared to and/or changed by a `Piece` object.

The Obvious Refactoring

The `Amount` interface must now extend the `Comparable` interface. The team originally considered defining the `Amount` interface as follows:

```
public interface Amount<T> extends Comparable<T>
{
    public abstract void changeBy(T other);
    public abstract String getUnit();
    public abstract int getValue();
}
```

However, this definition would enable someone to create a class that implements this interface in which the `changeBy()` method could be passed an object that was not an `Amount`. Hence, the team instead decided on the following definition:

```
public interface Amount<T extends Amount<T>> extends Comparable<T>
{
    public abstract void changeBy(T other);
    public abstract String getUnit();
    public abstract int getValue();
}
```

Other Refactorings

The declaration of the `AbstractAmount` class must be refactored to comply with the above change to the declaration of the `Amount` interface.

It seems like the `CompoundNumber`, `Piece`, and `Weight` classes will also need to be refactored.