

Coins Project

Sprint 4

Sprint Goal

The goal of the sprint is to add "dashboards" as observers. And, as always, our code meets our corporate requirements: checkstyle, unit testing.

Sprint tasks

1. Explore the CelsiusConverter codes.
2. Create a CoinCounts class that is composed inside of the Coin class.
3. Add a "total coins" dashboard observer that watches the CoinCounts object.
4. Add a "quarter counter" dashboard observe that watches the CoinCounts object.
5. Write a Demo class that adds to its demonstration of our code. We should be able to choose different coin creations and see the dashboards change automatically.

Domain details

The Treasury Department undersecretary is very happy with your progress so far! In fact, they want a "dashboard" to see how things are progressing. In particular, they want to see the total number of coins being manufactured and separately the number of quarters.

Implementation details

Play with the CelsiusConverter codes to understand how we can have a text UI "main" method create "windows" that display values. This code uses Java Swing to build a GUI window. Currently, the cs3667 server can not display a GUI window back to your local device. So, for now, you will need to work locally only. For Windows you can use the [Xming X11 server](#). For MacOS, it may work by default in the Terminal application. If not, you can use the [XQuartz X11 server](#).

We will add a new protected and static composition with CoinCounter to the Coin abstract class. The concrete coin constructors will call methods on this CoinCounter delegate (e.g., incrementQuarter()). Think about why this delegate needs to be static...

Using this knowledge, have your Demo create a TotalCoins "window" that employs the Observer pattern. First you will get the CoinCounter object reference from Coin. Then you will

pass that object to the TotalCoins window constructor. This constructor will register with CoinCounter as an observer.

Now when coins are made (by the Demo) they will increment the counts in the coin counter who will "update" the registered observers.

You can decide if you want the observers to work in a push or pull way.