**Explanation of Merchant's-Guide Code Design**

I have below three main functionalities Merchant's-Guide problem statements:

InputFileReader:

* This class reads input file line by line and store it in **DataStore** class of **repository** package which is **singleton** in nature and which is also work as third party **database**.

MetalProcessing:

* MetalProcessing read data containing metals only from DataStore and processEachMetal.
* During processing it calculate the Credits value for each metal and Store **Metal** object in DataStore for future use.

QuestionReading:

* QuestionReading read data containing question only from DataStore and calculate the answer for each question.
* During answering of questions it checks if questions constants “how much” or “how many” based on that it calculate RomanValues and CreditsValues.
* During calculation of creditsValues its uses DataStore for accessing each metal object.
* Invalid inputs are handle during calculation only and if found custom exception **InvalidInputException** is thrown from **exception** package
* All Calculations are done from **Calculation** class of **utility** package.
* All the string literals are handling from single point of interface **Constants** from **utility** package