|  |
| --- |
| Start Program |
| -void startExecution()  -void main() |

|  |
| --- |
| DataStore singleton class |
| Map<String, String> metalMeasureMap;  List<String> metalValueCalculationStrings;  List<String> questionMarkLines;  Map<String, Metal> entityMap;  Map<Character, Integer> romanNumeralsValueMapping;  DataStore *dataStore*; |
| -getInstance ()  - addMetal(Metal entityObj)  - hasThisMetal(String entityName)  -getMetalNameValue(String entityName)  - addMetalMeasure(String name, String value)  -hasThisMetalMeasure(String entityMeasure)  -getMetalMeasureValue(String entityMeasure)  -getValueCalculationLines()  -addValueCalculationLine(String line)  -getQuestionMarkLines()  -addQuestionMarkLine(String line) |

|  |
| --- |
| InputFileReader |
| DataStore dataStoreObj |
| -readInputFile(String fileName)  -performActionOnEachLine(String line) |

|  |
| --- |
| MetalProcessing |
| - DataStore dataStore;  -Calculation calculation; |
| -processEachMetal()  - MetalValueCalculation(String eachLine)  - getMetalMeasureInRoman(String[] metalData) |

|  |
| --- |
| QuestionReading |
| - DataStore dataStore;  -Calculation calculation; |
| -processEachLineWithQuestionMark ()  - answerQuestions(String question)  - calculateRomanValues(String question)  -getParsedQuestionString(String question)  - calculateCreditsValues(String question)  - |

|  |
| --- |
| Calculation |
| - DataStore dataStore; |
| -getRomanNumerals(String romanNumerals)  - calculateMetalValueAndStore(String metalName,String entityMeasureInRoman, **int** creditsValue)  - calculateRomanToDecimal(String romanName)- |

|  |
| --- |
| InvalidInputException |
|  |

|  |
| --- |
| Metal |
| String metalName  **float** metalValue |

**Explanation of My code design**

I have below three main functionality during solving 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