

Object Oriented Programs

1. JSON Inventory Data Management of Rice, Pulses and Wheats

- a. Desc -> Create a JSON file having Inventory Details for Rice, Pulses and Wheats with properties name, weight, price per kg.
- b. Use Library : [Java JSON Library](#), For IOS JSON Library use [NSJSONSerialization](#) for parsing the JSON.
- c. I/P -> read in JSON File
- d. Logic -> Get JSON Object in Java or NSDictionary in iOS. Create Inventory Object from JSON. Calculate the value for every Inventory.
- e. O/P -> Create the JSON from Inventory Object and output the JSON String

2. Inventory Management Program

- a. Desc -> Extend the above program to Create InventoryManager to manage the Inventory. The Inventory Manager will use InventoryFactory to create Inventory Object from JSON. The InventoryManager will call each Inventory Object in its list to calculate the Inventory Price and then call the Inventory Object to return the JSON String. The main program will be with InventoryManager
- b. I/P -> read in JSON File
- c. Logic -> Get JSON Object in Java or NSDictionary in iOS. Create Inventory Object from JSON. Calculate the value for every Inventory.
- d. O/P -> Create the JSON from Inventory Object and output the JSON String.

3. Stock Account Management

- a. Desc -> Write a program to read in Stock Names, Number of Share, Share Price. Print a Stock Report with total value of each Stock and the total value of Stock.
- b. I/P -> N number of Stocks, for Each Stock Read In the Share Name, Number of Share, and Share Price
- c. Logic -> Calculate the value of each stock and the total value
- d. O/P -> Print the Stock Report.
- e. Hint -> Create Stock and Stock Portfolio Class holding the list of Stocks read from the input file. Have functions in the Class to calculate the value of each stock and the value of total stocks

4. **Commercial data processing** - StockAccount.java implements a data type that might be used by a financial institution to keep track of customer information. The StockAccount class implements following methods

<code>public class StockAccount</code>	
<code> StockAccount(String filename)</code>	<i>create a new account from file</i>
<code>double valueOf()</code>	<i>total value of account dollars</i>
<code>void buy(int amount, String symbol)</code>	<i>add shares of stock to account</i>
<code>void sell(int amount, String symbol)</code>	<i>subtract shares of stock from account</i>
<code>void save(String filename)</code>	<i>save account to file</i>
<code>void printReport()</code>	<i>print a detailed report of stocks and values</i>

The StockAccount class also maintains a list of CompanyShares object which has Stock Symbol and Number of Shares as well as DateTime of the transaction. When buy or sell is initiated StockAccount checks if CompanyShares are available and accordingly update or create an Object.

5. Maintain the List of CompanyShares in a Linked List So new CompanyShares can be added or removed easily. Do not use any Collection Library to implement Linked List.
6. Further maintain the Stock Symbol Purchased or Sold in a Stack implemented using Linked List to indicate transactions done.
7. Further maintain DateTime of the transaction in a Queue implemented using Linked List to indicate when the transactions were done.

8. Write a Program ***DeckOfCards.java***, to initialize deck of cards having suit ("Clubs", "Diamonds", "Hearts", "Spades") & Rank ("2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King", "Ace"). Shuffle the cards using Random method and then distribute 9 Cards to 4 Players and Print the Cards received by the 4 Players using 2D Array...
9. Extend the above program to create a Player Object having Deck of Cards, and having ability to Sort by Rank and maintain the cards in a Queue implemented using Linked List. Do not use any Collection Library. Further the Players are also arranged in Queue. Finally Print the Player and the Cards received by each Player.