## Problem Statement

You need to design a Vending Machine which

1. Accepts coins of 1,5,10,25 Cents i.e. penny, nickel, dime, and quarter.
2. Allow user to select products Coke(25), Pepsi(35), Soda(45)
3. Allow user to take refund by canceling the request.
4. Return selected product and remaining change if any
5. Allow reset operation for vending machine supplier.

 Some of the requirement are also implicit but it's better to make it explicit in your list e.g. In this problem, vending machine should not accept a request if it doesn't have sufficient change to return.

**Solution and Coding**

My implementation of Java Vending Machine has following classes and interfaces :  
  
VendingMachine  
It defines the public API of vending machine, usually all high-level functionality should go in this class  
  
VendingMachineImpl  
Sample implementation of Vending Machine  
  
VendingMachineFactory  
A Factory class to create different kinds of Vending Machine  
  
Item  
Java Enum to represent Item served by Vending Machine  
  
Inventory  
Java class to represent an Inventory, used for creating case and item inventory inside Vending Machine  
  
Coin  
Another Java Enum to represent Coins supported by Vending Machine  
  
Bucket  
A parameterized class to hold two objects. It's kind of Pair class.  
  
NotFullPaidException  
An Exception thrown by Vending Machine when a user tries to collect an item, without paying the full amount.  
  
NotSufficientChangeException  
Vending Machine throws this exception to indicate that it doesn't have sufficient change to complete this request.  
  
SoldOutExcepiton  
Vending Machine throws this exception if the user request for a product which is sold out.  
  
Read more: <https://javarevisited.blogspot.com/2016/06/design-vending-machine-in-java.html#ixzz65Jj8Cg8t>