You have to code a working coffee machine which can stock ingredients, prepare beverages, display the current quantity of stock available. One or more ingredients will be used to prepare beverages. Same ingredients can be used to prepare multiple beverages. Initial ingredients' stock and the ingredients required to prepare each beverage will be given. Coffee machine displays success messages for stocking ingredients and preparing beverages. It also displays proper error messages with all the missing ingredients in case of unavailability of stock.

The coffee machine has following ingredients

Ingredients ------------ Initial quantity of stock

hot_water ------- 500

hot_milk ------ 500

coffee_decoction ----- 100

sugar_syrup ------100

tea_leaves_syrup ------100

green_tea_syrup ------- 100

below is the list of Ingredients required to prepare beverages.

hot_tea

hot_water - 200

hot_milk - 100

sugar_syrup - 30

tea_leaves_syrup - 30

hot_coffee

hot_water - 100

hot_milk - 400

coffee_decoction - 50

sugar_syrup - 30

black_tea

hot_water - 300

sugar_syrup - 20

tea_leaves_syrup - 30

green_tea

hot_water - 200

sugar_syrup - 20

green_tea_syrup - 30

Sample test case

add_stock(“hot_water”,100)

add_stock(“sugar_syrup”, 200)

get_stock()

prepare(“black_tea”)

we were supposed to implement above APIs keeping extensibility, maintainablility etc of code.

My Approch: I cretaed two interfaces IIngredients, Ibeverages. IIngredients interface has some method like getName(), getIngredientQuantity(), setIngredientQuantity(). Ibeverages

interface has methods like setIngredients(), getIngredients() etc. all Ingredients classes will implement IIngredients interface and all Beverages subclasses will implement this Ibeverages interface. Then I have a facade which is single point of entry to this coffee machine. this facade is implemented with Singelton pattern. since every object either ingredient or beverages needs to be created only one time. i kept all objects in a map, once they are created.

object creation logic was residing in facade class, but i was thinking of moving it to factory method pattern.

Verdict:

I demostrated my running code to interviewer and he seemed satisfied. but recruiter told me that i was rejected. I dont know what happened.

Request from community:
If you have some better design in mind for this problem Or you are the one who cleared this round, please let us all know your design.