Freightos Qualification Assignment

–

Vending Machine

Waleed Titi

Waleed.titi8@gmail.com

Introduction

The assignment was basically to develop a vending machine, where the user puts the input and then based on the current state of the vending machine (if items are available or not) and based on the input of the user (if he has enough money, if his card is accepted etc) an item is dispensed to the user.

Classes

Product Class

The product class contains the name, id, stock and price of the product and a method to purchase a product. The method checks if the product is out of stock or not, if the product was in stock, it approves the purchase and subtracts 1 from its stock.

Payment Class

The payment class contains the method of payment (cash or card), the amount you want to pay, the total amount of money you put in and the remaining amount.

The total amount and the remaining amount are set to 0 initially, because technically nothing happened so they can't be anything but 0.

In the payment class there are 3 methods:

- **Validate_Card() :** This method returns True and approves the card if the card is approved by the bank (enough balance, acceptable card, etc) but I assumed the card is always accepted and did not check for bank`s approval since I don't have access to it.
- **Validate_Cash(amount: float, machine: SnackMachine):** This method checks if the cash entered by the user is approved or not, if the entered amount by the user is USD, and if the coins or notes entered by the user, the machine accepts. It also checks if the user wants to continue with the payment or not, at any time the user can press "N" and the machine immediately stops and dispenses the total amount entered by the user. It basically keeps running until the user has entered more or equal to the amount of the snack and returns True.
- **dispesningCoins(amount:float, coins:dict):** This method is a greedy method to dispense remaining coins after a purchase or after the user has declined the purchase and has entered money in the machine. The method basically starts from the max possible coin that is less or equal to the remaining amount to dispense and adds it to an array of floats called output. It also checks if the coin is available or not before adding it.

SnackMachine Class

This class is the vending machine class and has a capacity, slots for the items and coins which is a dict.

The slots are 5 x 5 array filled with Products objects, and the coins is a dictionary where the keys are the accepted coins and the values are the amount of those available in the machine.

This class has a method called checkAvailableSnack(prod:Product) that checks if the product in the current slot is available or not, if it has enough stock or not, and if the capacity of the machine is above 0 or not, if everything is available it returns True, if not it returns False.

Customer Class

The customer is basically the main class and has to do all the functionalities for the vending machine.

I started by creating the vending machine, I initialized the slots by creating a 5 x 5 Products array and instead of filling 25 items I used random number generator, the ID is a string that contains the product row and column which is for example: (“13”).

The name is a string that is “Item at row {} and column {}” based on the for loop.

I came up with a simple strategy for the prices of the products, basically it’s a random number between 0 and 10 and then I subtract from it 1 or 0.5.

The 0 and 10 was a method called randint(0,10) which always gives a number between 0 and 10, and the 1 or 0.5 was by using the (randint(0,2) / 2) which will give (0,0.5,1).

However, it sometimes (because its RNG) can result in a price that is 0 or less, so I iterated over the array and checked for any price equal or less than 0 and changed it to 0.5.

The stocks is the same thing, it’s a randint(0,200) between 0 and 200.

The vending_coins is self-explanatory, a dictionary with keys as accepted coins and their values is the amount of those coins available in the machine.

I did check for the user input and made sure he/she always puts a 2 digit in range number

Then I check if the item is available or not. If it`s available the vending machine asks the user to choose payment type, card or cash.

I also validated the input of the user if its neither it prints wrong input.

Every time the user enters an approved amount, it prints the total amount the user has entered and when its equal to or above the price of the item, it dispenses the item and the remainder if there is any.

Scenarios

I checked for possible scenarios for the vending machine, here they are.

- **If Item is available**
- **If user`s input is not valid**
- **If card is valid**
- **If cash is valid and okay**
- **If the user decided to cancel in the middle of the operation**
- **If machine has enough change**
- **How to dispense change**
- **If the user inserted less amount**

test_vending.py

This file tests the functionalities of the entire classes whether the outcome is what I wanted or not. You can change it based on the input you want and it should give u the results you desire.

Whenever you want to run the program, run the Customer.py file.