Home Work 6

July 18, 2024

Dylan Liesenfelt

1 Instructions:

Select ***ONLY ONE*** of the following exercises and solve it.

Submit your code along with explanations and test results that demonstrate the implementation of the required features.

Submit a single .pdf document in Canvas with both code and results.

Always remember, "Practice makes perfect!"

2 Exercise 4: Simple Online Store Inventory

Background: Develop a system to manage inventory for an online store.

Tasks: * Define a Product class with attributes for the product ID, name, and price. * Define an Inventory class that maintains a list of Product objects. * Implement methods in the Inventory class to add, remove, and search for products within the inventory. * Test the system by creating multiple products and managing them within the inventory system.

[ ]:

```python
class Product:
    # Creating Product class w/ attributes ID, name, and price
    def __init__(self, ID, name, price):
        self.ID = ID
        self.name = name
        self.price = price

    # Using __repr__ method so object data will be readable instead of showing memory address of object
    def __repr__(self):
        return f"Product(ID: '{self.ID}', Name: '{self.name}', Price: {self.price})"

class Inventory:
    # Creating Inventory class with a list of 'products' as na attribute
    def __init__(self):
        self.products = []
    # Method to add a product to the Inventory
    def addProduct(self, product):
        self.products.append(product)
        print(f'Added {product} to the Inventory')

    # Method to remove a product from the Inventory
    def removeProduct(self, ID):
        # Searches the products list (Inventory), by product ID
        for product in self.products:
            if product.ID == ID:
                self.products.remove(product)
                print(f'Removed {product} from Inventory')
                return
        print(f'Product ID: {ID} does not exist')

    # method to search Inventory tby product ID again but this time returns the␣
 ↪intended object
    def searchInventory(self, ID):
        for product in self.products:
            if product.ID == ID:
                return product
            else:
                return print(f'Product ID: {ID} is not in Inventory')
```

[ ]:

```
# Creating the Product objects
product1 = Product('001', 'Computer', 1500)
product2 = Product('002', 'Laptop', 800)
product3 = Product('003', 'TV', 500)
product4 = Product('004', 'Headphones', 50)
product5 = Product('005', 'Phone', 400)

electronicsStoreInventory = Inventory() # Call the inventory and assign it to a␣
 ↪var
print('The Inventory should be empty ->', electronicsStoreInventory.products,␣
 ↪'\n')

# Demonstrating add method
electronicsStoreInventory.addProduct(product1)
electronicsStoreInventory.addProduct(product2)
electronicsStoreInventory.addProduct(product3)
electronicsStoreInventory.addProduct(product4)
electronicsStoreInventory.addProduct(product5)

print('\nThe Inventory should have stuff in it ->', electronicsStoreInventory.
 ↪products, '\n')
```

```python
# Demonstrating remove method
electronicsStoreInventory.removeProduct('003') # Removing product w/ the ID of␣
 ↪003
print('The Inventory should have stuff in it ->', electronicsStoreInventory.
 ↪products, '\n')

# Demonstrating search method
print(electronicsStoreInventory.searchInventory('001'))
print(electronicsStoreInventory.searchInventory('003'))
```

```
The Inventory should be empty -> []

Added Product(ID: '001', Name: 'Computer', Price: 1500) to the Inventory
Added Product(ID: '002', Name: 'Laptop', Price: 800) to the Inventory
Added Product(ID: '003', Name: 'TV', Price: 500) to the Inventory
Added Product(ID: '004', Name: 'Headphones', Price: 50) to the Inventory
Added Product(ID: '005', Name: 'Phone', Price: 400) to the Inventory

The Inventory should have stuff in it -> [Product(ID: '001', Name: 'Computer',
Price: 1500), Product(ID: '002', Name: 'Laptop', Price: 800), Product(ID: '003',
Name: 'TV', Price: 500), Product(ID: '004', Name: 'Headphones', Price: 50),
Product(ID: '005', Name: 'Phone', Price: 400)]

Removed Product(ID: '003', Name: 'TV', Price: 500) from Inventory
The Inventory should have stuff in it -> [Product(ID: '001', Name: 'Computer',
Price: 1500), Product(ID: '002', Name: 'Laptop', Price: 800), Product(ID: '004',
Name: 'Headphones', Price: 50), Product(ID: '005', Name: 'Phone', Price: 400)]

Product(ID: '001', Name: 'Computer', Price: 1500)
Product ID: 003 is not in Inventory
None
```