Code.Hub

The first Hub for Developers

Software Engineering Bootcamp
Java & Angular
powered by Pfizer

Pfizer

Java fundamentals
Assignment in Inheritance

Deadline : 2-9-2020, 23:59

at your GitHub account

Code.Hub

Exercise 1

Tasks

1. Implement a base class Account with the following fields owner:String, balance:Double, numberOfTransactions:int
2. Provide suitable constructor(s), the methods withdraw and deposit.
3. Write a tester method that uses all the methods of the class and displays suitable messages

Exercise (cont.)

4. Implement a subclass StoreAccount, using as base the Account class with the following extra fields storeName:String, accountCategory:string

5. Provide suitable constructor(s), the methods withdraw and deposit, changeCategory

6. Write a tester method that uses all the methods of the class and displays suitable messages

Note: categories are basic, mid, extra, premium.

Exercise 2

1. Define a POJO class Product with the following fields

id: int , name:string , priceWhenBuy:double, priceWhenSell:double

2. Create an interface for the Store called IStore signatures of desired methods
buy(Product), sell(Product), getRevenue():double

Homework exercise (cont.)

3. Create an implementation of IStore named **SimpleRetailStore** is a store that keeps no Inventory. Fields: totalBuyBalance:double, totalSellBalance :double

Create the suitable methods so that main method in the Main class will be the following:

```
Product x1 = new Product(1, "Painting Picasso. Guernica", 100, 1000);
Product x2 = new Product(2, "Painting Tsarouxis. Naftis A", 200, 2000);
Product x4 = new Product(3, "Chair. Luis XV", 100, 1000);

IStore simple= new SimpleRetailStore();
simple.buy(x1);      simple.buy(x4);      simple.sell(x1);
System.out.println(simple.getRevenue());
```

Homework exercise (cont.)

4. Create another implementations of IStore named **InventoryRetailStore** is a store that has detailed Inventory. The event buy adds to Inventory and sell removes from Inventory

Fields:
List<Product>
Methods: getInventory(), reset()

Create the suitable methods so that main method in the Main class will be the following:

```
Product x1 = new Product(1, "Painting Picasso. Guernica", 100, 1000);
Product x2 = new Product(2, "Painting Tsarouxis. Naftis A", 200, 2000);
Product x4 = new Product(3, "Chair. Luis XV", 100, 1000);

IStore invStore = new InventoryRetailStore();
invStore.buy(x1);   invStore.buy(x4);   invStore.sell(x1);
System.out.println(invStore.getRevenue());
invStore.sell(x2);  invStore.getInventory();
```