

PHP Classes 3

Simulate the working process of car dealer shop with employees who sell cars to clients.

Create class Car. This class will contain following private properties: manufacturer, model, year, price. Class constructor needs to receive all the properties and before setting these values check if manufacturer and model are valid values. For validating create

manufacturer.txt file and insert example data in following format:

Example manufacturer.txt:

Toyota:corolla,aygo

Kia:ceed,i35

If the specified model and manufacturer are present in the file constructor will set the values otherwise print: Manufacturer or model is not valid.

Create class Person. This class contain following protected properties: firstName, lastName. Create class constructor and abstract method identity().

Create class Client that extends the class Person. This class contain following private properties: budget, list of cars(objects from Client class). Create appropriate setters and getters. The identity() method should return the following text: "\$firstName \$lastName".

Create class Manager that extends the class Person. This class contain following private properties: points. The identity() method should return the following text: "\$firstName \$lastName".

Create class CarDealer. This class contains following private properties: name, list of employees(object from Manager class), available cars for selling(in following format:

```
[  
    ['car'=> $car, 'quantity'=>$quantity]  
])
```

and create the appropriate class constructor.

Create method addCar(\$car, \$quantity).

Create method addEmployee(\$manager).

Create method sellCar(\$manager, \$client, \$manufacturer(string), \$model(string)). In this method manager is selling car to client. Check if CarDealer have available car(quantity > 0) from provided manufacturer and model. Every manager gets 11 points for every sold car and depending on manager points he can calculate discount. The rule for discount is:

1<= discount <=20 -> 5%

20< discount <=40 -> 10%

40< discount <=60 -> 15%

60< discount <=80 -> 20%

80< discount <=100 -> 25%

After calculating discount for current selling check if client have enough budget. If client have budget add specified car to customer, decrease the quantity by 1 and add points to manager. If some of these validations fails print message: "Selling process is not completed for customer: {\$client->identity()}"

Create method `topRatedEmployee()` where you will find the best selling employed manager and print the following text: `$topRatedEmployee->identity() . ' points:'` .
`$topRatedEmployee->getPoints();`
Otherwise print: "Car dealer does not have employees or cars were not sold"

Create separate file for every class/interface and include them in the main script. If you need additional method in some class feel free to add.