

## PHP Classes 2

Simulate the working process of detective agency with the existence of clients who need detective service and detectives who offer services for clients.

Create class Service. This class contains following private properties: description, service duration(hours) and price. Class constructor needs to receive and set all of the properties. You need to create appropriate setters and getters. Additionally create information() method that will print the description, service duration and the price.

Create abstract class Person that contains following private properties: first name, last name and address and define the abstract method identity().

Create class Client that extends the class Person. Additionally this class contains following private properties: personal budget and his list of services(objects from class Service). Class constructor needs to set the properties and call the parent constructor.

Identity() method should return the following text: "First name: " . \$this->firstName . " Last name: " . \$this->lastName . " Address: " . \$this->address . " Services: (result of every service information() method)

Create class Detective that extends the Person class. Additionally this class contains following private properties: available working time(hours) and list of services assigned to him(objects from class Service). Class constructor needs to set the properties and call the parent constructor.

Identity() method should return the following text: "First name: " . \$this->firstName . " Last name: " . \$this->lastName . " Address: " . \$this->address . " Services: ": (result of every service information() method if detective doesn't have any service return "Services are not assigned yet")

Create interface Actions that contains the contains method addService() and implement this interface in Client and Detective classes.

When this method is calling from Client class check if the price of all services that belongs to client are lower than his budget otherwise print: "Service: " . \$service->information() . " is too expensive."

When this method is called from Detective class check if the sum of all services duration time that are assigned to detective are lower than his available working time otherwise print: "Detective is too busy for service: " . \$service->information().

Create class DetectiveAgency with following private properties: name, list of clients and list of detectives. Class constructor needs to set all the properties.

Create method addClient() and check if the client have more then 3 services assigned otherwise print: "Customer" . \$client->identity() . " cannot be added to this agency"

Create method addDetective() that add new detective to the agency.

Create method revenue() that calculate the revenue of the agency. For every client that is added to the agency the agency receive 90% of his services budget. For every detective that is added to the agency the agency receive 15% of his services budged.

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. For example method in Detective class that will calculate fee for the agency.