BLM2012 Object Oriented Programming, Course Project: An Information System for Service Providers
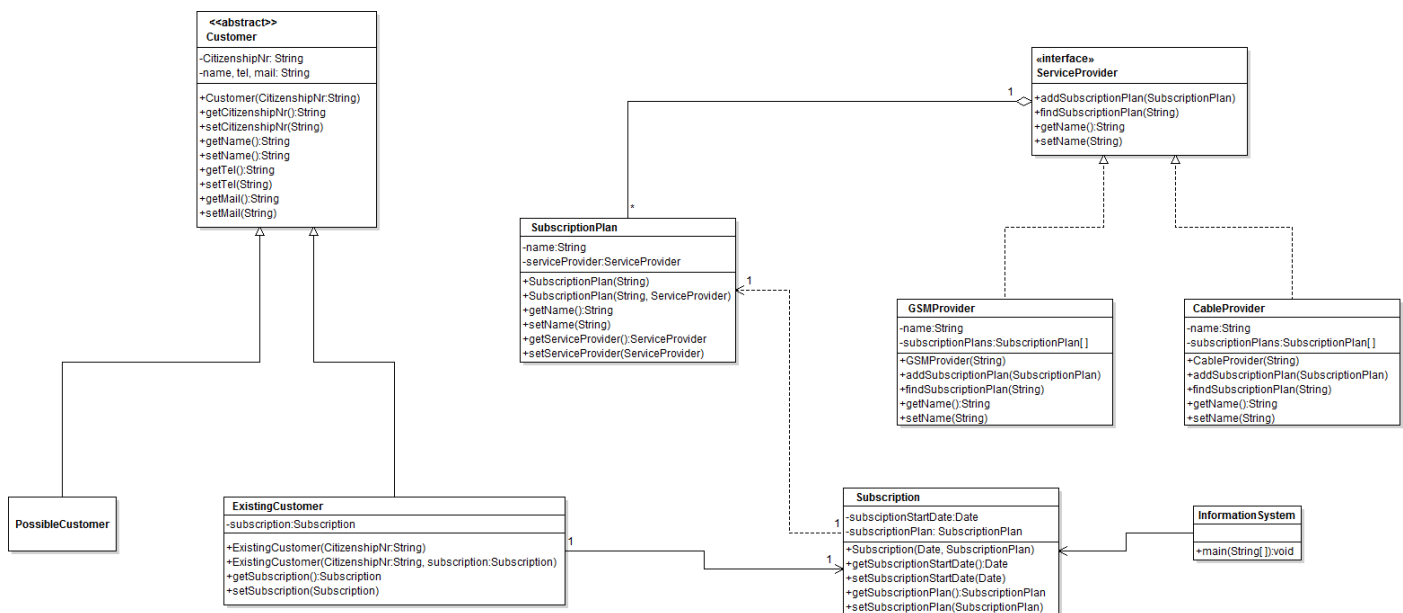## Due 07.06.2021, Monday, 23:59 PM

Description: You shall implement an Information System for a company, which is selling Subscriptions for internet connection from different Service Providers (GSM, Cable, etc...). Your program shall keep track of records of Existing Customers as well as records of Possible Customers who may have shown interests. At this point, you shall only accept customer applications for Subscription Plans of two different Service Providers: GSM and Cable. However, your program needs to be flexible. That is, your program should give you flexibility so that you may also sell Subscription Plans for different Service Providers in the future. To make it easier for you, we provide you the main UML class diagram of the program below.

<> Customer
- CitizenshipNr: String
- name, tel, mail: String

- +Customer(CitizenshipNr:String)
- +getCitizenshipNr():String
- +setCitizenshipNr(String)
- +getName():String
- +setName():String
- +getTel():String
- +setTel(String)
- +getMail():String
- +setMail(String)

«interface» ServiceProvider
- +addSubscriptionPlan(SubscriptionPlan)
- +findSubscriptionPlan(String)
- +getName():String
- +setName(String)

SubscriptionPlan
- name:String
- serviceProvider:ServiceProvider

- +SubscriptionPlan(String)
- +SubscriptionPlan(String, ServiceProvider)
- +getName():String
- +setName(String)
- +getServiceProvider():ServiceProvider
- +setServiceProvider(ServiceProvider)

GSMProvider
- name:String
- subscriptionPlans:SubscriptionPlan[ ]

- +GSMProvider(String)
- +addSubscriptionPlan(SubscriptionPlan)
- +findSubscriptionPlan(String)
- +getName():String
- +setName(String)

CableProvider
- name:String
- subscriptionPlans:SubscriptionPlan[ ]

- +CableProvider(String)
- +addSubscriptionPlan(SubscriptionPlan)
- +findSubscriptionPlan(String)
- +getName():String
- +setName(String)

PossibleCustomer

ExistingCustomer
- subscription:Subscription

- +ExistingCustomer(CitizenshipNr:String)
- +ExistingCustomer(CitizenshipNr:String, subscription:Subscription)
- +getSubscription():Subscription
- +setSubscription(Subscription)

Subscription
- subsciptionStartDate:Date
- subscriptionPlan: SubscriptionPlan

- +Subscription(Date, SubscriptionPlan)
- +getSubscriptionStartDate():Date
- +setSubscriptionStartDate(Date)
- +getSubscriptionPlan():SubscriptionPlan
- +setSubscriptionPlan(SubscriptionPlan)

InformationSystem
- +main(String[ ]):void

1- You shall implement all the classes shown in the UML diagram above.
2- You shall implement the main method in the "InformationSystem" class.
3- In your main method you shall ask user the following inputs:

    a. GSM SERVICE PROVIDER ? : (Example input: TURKCELL)
    b. NUMBER OF SUBSCRIPTION PLANS ? : (Example input: 1)
    c. NAME OF THE Plan #0 ?: (Example input: 4GB)
    d. CUSTOMER CITIZENSHIP NUMBER ? : (Example input: 32643509892)
    e. CUSTOMER NAME ? : (Example input: AHMET OZTURK)
    f. NAME OF THE PLAN TO BE SUBSCRIBED ? : (Example input: 4GB)
    g. START DATE FOR SUBSCRIPTION (MM/DD/YYYY)? : (Example input: 11/12/2013)

Note that after getting the inputs a, b and c; you shall create a Service Provider and Service Plan(s) that this Service Provider company provides. After getting the inputs d and e; you shall create an object for Customer. After getting the f; you shall find the Subscription Plan that this customer wants to subscribe. After getting the start date for the subscription, you shall create a Subscription with the Subscription Plan and start data for this Customer.

4- You shall print out the output. The following is a sample output:

```
-------------------------------------------------------
CUSTOMER CITIZENSHIP NUMBER = 32643509892
CUSTOMER NAME = AHMET OZTURK
SERVICE PROVIDER : TURKCELL
SUBSCRIPTION START DATE : 11/12/2013
SUBSCRIPTION PLAN NAME : 4GB
-------------------------------------------------------
```

5- In addition to text based console I/O, you may choose to add GUI functionality to your project as it will be graded with up to 15 bonus points.

6- Your program shall be tested during grading by executing the InformationSystem class on the command line. Make sure that your programs compiles and run OK on the command line. Make sure to include instructions how to compile and run your program in the README.txt file. Additional grading will be given according to some undisclosed unit tests.

7- You are required to add unit tests to your project, too. There must be at least two test cases for each class in the main UML class diagram.

8- This is an individual project. You are required to do your project on your own. Your code is subject to be checked against other student's code. Cheating is not allowed and will not be tolerated. You should submit your projects in a zip file which contains your well commented source code and README.txt file as a zip file to online.yildiz.edu.tr.

9- Projects are due at 11:59pm on the date specified. A grading penalty will be applied to late assignments. (10% penalty up to the first 24 hours, 20% for 24 to 48 hours, with no credit received after that)

10- Submitting a project is mandatory. (Submitted projects will be demonstrated according to a schedule that will be announced later.)