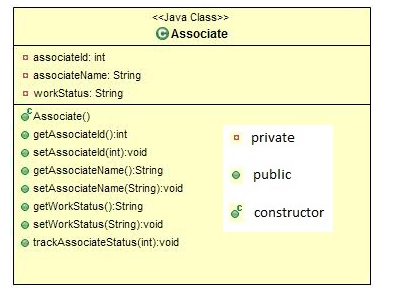
1. DreamTek Company provides an initial training for all its employees, once they join the company. During the training phase they call the employees as “Associate”. The initial training is conducted for 60 days for each Associate. In these 60 days they learn various technologies. The first 20 days they learn “Core skills”, the next 20 days they learn “Advanced modules” and the final 20 days they go to the “Project phase”.  Help the DreamTek Company to find in which phase the associates are in.

Consider the below class:



In the **Associate**class include the given attributes and methods with the access specifiers as specified in the class diagram.

The setter methods are used to set the value and the getter methods are used to retrieve the value.

The trackAssociateStatus method takes the number of days as argument and sets the work status of the associate based on the number of days. If the number of days is greater than 60 days then set the work status as “Deployed in project”.

 In the **Main** class, create an object for the Associate class; Get the details as shown in the sample input and assign the value for its attributes using the setters. Invoke the trackAssociateStatus method and find the work status and display the details as shown in the sample output.

**Sample Input1:**

Enter the associate id:

123

Enter the associate name:

john

Enter the number of days:

45

**Sample Output1:**

The associate john work status:Project phase

**Sample Input2:**

Enter the associate id:

124

Enter the associate name:

ram

Enter the number of days:

70

**Sample Output2:**

The associate ram work status: Deployed in project

2. In the first round of HR interview for a banking sector,  HR decides to make candidates design an application which provides only  information on transaction like amount withdrawn with respect to fields given. Develop a program to implement this scenario.

Create a class Account with the private attributes:

* accountId  int
* accountType String
* balance int

The method **public** **boolean withdraw(int)** used  to calculate the current balance of the respective account. Before that it should enough balance.  If there is enough balance, deduct the amount from the balance and print "Balance amount after withdraw: XXX" and return true.  If there is no enough balance, print "Sorry!!! No enough balance" and return false.

Create a class AccountDetails with main function and the below methods :

* public Account getAccountDetails() -  This methods gets the input related to Account from the user and returns the Account object with all values set.  If the input given for balance is less than or equal to zero, consider it as invalid and display "Balance should be positive". Continue this kind of evaluation till user enters a positive value.

* public int getWithdrawAmount() -  This methods gets the amount to be withdrawn as input from the user and returns the same.  If the input given for amount is less than or equal to zero, consider it as invalid and display "Amount should be positive". Continue this kind of evaluation till user enters a positive value.

**Use appropriate getters and setters.**

**Sample input 1:**

Enter account id:

100  
Enter account type:

Savings

Enter balance:

10000  
Enter amount to be withdrawn:

500

**Sample Output 1:**

Balance amount after withdraw: 9500

**Sample input 2:**  
Enter account id:

101  
Enter account type:

Savings  
Enter balance:  
1000  
Enter amount to be withdrawn:  
1500

**Sample Output 2:**

Sorry!!! No enough balance

**Sample input 3:**

Enter account id:

100

Enter account type:

Savings

Enter balance:

-100

Balance should be positive

Enter balance:

5000

Enter amount to be withdrawn:  
500

**Sample Output 1:**

Balance amount after withdraw: 4500