Call Details

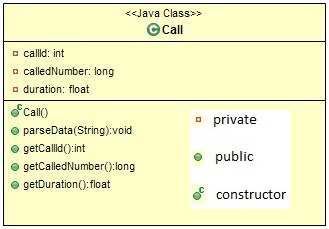
For the postpaid customers, the airvoice mobile shop maintains the calls made by the customer in the file in the given format below:

Callid:callednumber:noofminutes

**Example:** 123:9874561230:2.5

The shop wants to extract the information from the file and populate them in to the Call object. Develop a java application to do the above task.

Consider the class given below:



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

The getter methods are used to retrieve the value.

The **parseData** method takes the string as argument, the string value will be (123:9874561230:2.5). This method should extract the callid, called number and number of minutes from the string and set the callId, calledNumber and noOfMinutes.

In the **Main** class, create an object for the Call class; Get the details as shown in the sample input and invoke the parseData method.  Display the details as shown in the sample output using the getters method.

**Note:** The attribute/method/class name should be specified correctly as given in the class diagram.

**Sample Input:**

Enter the call details:

102:6547891230:2.15

**Sample output:**

  Call id:102

  Called number:6547891230

  Duration:2.15

Employee rating

DreamTek Company provides a rating to its employees based on the “Certification” they have completed.  For each certification completed by the employee the rating will be increased by one. Based on the rating the company planned to provide an increment for the employees. For the current year , the employees can take up any of the certification courses. They are:

a.       JAVA

b.      ORACLE

c.       GCUX

d.      CCNA

e.      AWS

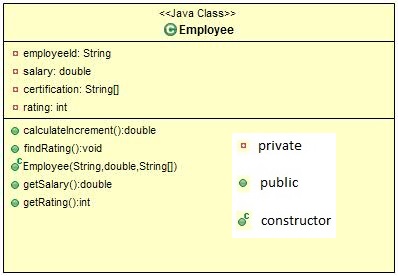
For each certification the rating will be 1. If an employee has completed three certifications, their rating will be 3. The employee is supposed to take up the certification suggested by the company. If the employee takes up some other certification, their rating will not be updated for that certification.

The increment for the employee will be given based on the below criteria:

|  |  |
| --- | --- |
| Rating | Increment Percentage |
| 1 | 2% |
| 2 | 3.5% |
| 3 | 5% |
| 4 | 7.5% |
| 5 | 10% |

Help the DreamTek Company to develop a java application to do the above task.

Consider the below class:



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

The constructors are used to initialize the value and the getter methods are used to retrieve the value.

The **findRating()** method should set the rating based on the certification completed by the employee.

The **calculateIncrement()** method should calculate the increment based on the rating and update the salary with the incremented amount. This method should return the increment amount. (For example: if the salary is 50000, and the incremented amount is 4000, then this method should return 4000, and  update the salary as 54000).

In the **Main** class, Get the details as shown in the sample input and create an object for the Employee class; invoke the appropriate methods to get the results as shown in the sample output.

**Sample Input1:**

Enter the employee id:

TEK163

Enter the salary:

50000

Enter the no of certification complete:

3

Enter the certification names:

JAVA

ORACLE

CCNA

**Sample Output1:**

Your rating is 3

Increment amount is 2500.00

Current salary 52500.00

**Sample Input2:**

Enter the employee id:

TEK163

Enter the salary:

50000

Enter the no of certification complete:

3

Enter the certification names:

JAVA

J2EE

CCNA

**Sample Output2:**

Your rating is 2

Increment amount is 1750.00

Current salary 51750.00

**Sample Input3:**

Enter the employee id:

TEK163

Enter the salary:

50000

Enter the no of certification complete:

3

Enter the certification names:

PHP

J2EE

MYSQL

**Sample Output3:**

Your rating is 0

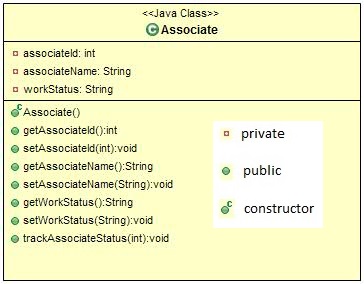
Increment amount is 0.00

Current salary 50000.00

DreamTek Company

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 Output 1:**

The associate john work status:Project phase

**Sample Input 2:**

Enter the associate id:

124

Enter the associate name:

ram

Enter the number of days:

70

**Sample Output 2:**

The associate ram work status:Deployed in project

ZeeZee bank

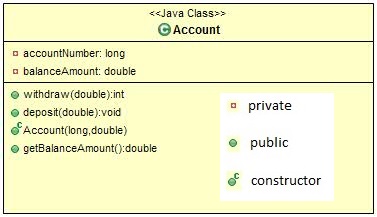
ZeeZee bank maintains the account details for each customer. The account details are:

a.       Account number

b.      Type of account

c.       Balance amount

Consider the below class:



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

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

The **withdraw** method takes the amount to be withdrawn as argument. This method should check the balance and detect the withdrawn amount from the balance amount and return 1. If there is insufficient balance then return -1.

The **deposit** method takes the amount to be deposited as argument and adds the deposited amount to the balance amount.

 In the **Main** class, Get the details as shown in the sample input and create an object for the Account class; invoke the deposit method to deposit the amount and withdraw method to withdraw the amount from the account.

**Sample Input1 & Output1:**

Enter the account number:

1234567890

Enter the available amount in the account:

15000

Enter the amount to be deposited:

1500

Available balance is:16500.00

Enter the amount to be withdrawn:

500

Available balance is:16000.00

**Sample Input2 & Output2:**

Enter the account number:

1234567890

Enter the available amount in the account:

15000

Enter the amount to be deposited:

1500

Available balance is:16500.00

Enter the amount to be withdrawn:

18500

Insufficient balance

Available balance is:16500.00

Book Detail

Create a class Book with the following private member variables

* String bookName
* int bookPrice
* String authorName

Include appropriate getters and setters method.

Create a class TestBook which has the main method.  Get the details as shown in the sample input. Create an object for book class and assign the value for its attrbutes using the setters.  Print the output as shown in the sample output using the getters method.

Note: Use the same attribute names as given in the question and camel case notation for methods.  Name of book and author can have space in between.

**Sample Input 1:**

Enter the Book name:  
Java  
Enter the price:  
500  
Enter the Author name:  
Einstein

**Sample Output 1:**

Book Details  
Book Name :Java  
Book Price :500  
Author Name :Einstein

Ticket Price Calculation - Static

**Ticket Calculation**

Create a class Ticket with the following private variables  
int ticketid;  
int price;  
static int availableTickets;

Include getters and setters methods in the Ticket class.

AvailableTickets should hold only positive value. Zero and negative values are not allowed.(This logic should be checked inside the corresponding setter method)

Write the following method in the Ticket class:

public int calculateTicketCost(int nooftickets) —this method should check the ticket availability, If the tickets are available, reduce the nooftickets from availableTickets and calculate the total amount as nooftickets\*price  and return the total amount.  If the tickets are not available, this method should return -1.

Write a main method in the Main class to test the application.

**Sample input and output 1:**

Enter no of bookings:  
2  
Enter the available tickets:  
25  
Enter the ticketid:  
123  
Enter the price:  
100  
Enter the no of tickets:  
5  
Available tickets: 25

Total amount:500

Available ticket after booking:20

Enter the ticketid:  
124  
Enter the price:  
100  
Enter the no of tickets:  
2  
Available tickets: 20

Total amount:200

Available ticket after booking:18

**Sample input and output 2:**

Enter no of bookings:  
1  
Enter the available tickets:  
25  
Enter the ticketid:  
123  
Enter the price:  
100  
Enter the no of tickets:  
26  
Tickets not sufficient / available

Student Details - Constructor

Create a class Student with the private attributes

int studentId

String studentName, studentAddress, collegeName.

Include appropriate getter methods.

Write 2 constructors for the Student class based on the below assumptions.

Assume most of the students are from “NIT” college. So user has to give input whether the student is from NIT or not.

1. If student belongs to NIT, give input as 'yes/YES' and  skip input for the attribute collegeName  and create student object with 3-argument constructor to initilze the values for studentId, studentName and studentAddress and  collegeName as “NIT”.
2. If student belongs to other college, give input as 'no/NO' and get college name from the user and create student object with 4-argument constructor to initialize all the values.
3. Instead of Yes / No, if user enters different input then display 'Wrong Input' and get the input again.

Based on the above assumptions write the necessary constructors in the Student class.

Write a class StudentMain with the main method and test the application.

Get all the input needed from the main method.

**Sample Input 1:**

Enter Student's Id:

12

Enter Student's Name:

John

Enter Student's address:

Chennai

Whether the student is from NIT(Yes/No):

NO

Enter the college name:

SVS

**Sample Output 1:**

Student id:12

Student name:John

Address:Chennai

College name:SVS

**Sample Input 2:**

Enter Student's Id:

43

Enter Student's Name:

Tom

Enter Student's address:

Coimbatore

Whether the student is from NIT(Yes/No):

y

Wrong Input

Whether the student is from NIT(Yes/No):

yes

**Sample Output 2:**

Student id:43

Student name:Tom

Address:Coimbatore

College name:NIT

## BankAccountDetails

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

Student and Department Detail

Create a class Department with the following private member variables

* int did
* String dname

Include appropriate getters and setters method in Department class.

Create a class Student with the following private member variables

* int sid
* String sname
* Department department

Include appropriate getters and setters method in Student class.

Create a TestMain class which has main method.

In addition to main method, create a method

           public static Student createStudent() -  All input as shown in the sameple input should be got in this method.  Set the values to the Student object and return that object

Note : In main method, invoke the createStudent method and print the details of the object returned by that method.

**Sample Input 1:**

Enter the Department id:  
100  
Enter the Department name:  
Computerscience

Enter the Student id:  
123  
Enter the Student name:  
sudha

**Sample Output 1:**

Department id:100  
Department name:Computerscience  
Student id:123  
Student name:sudha