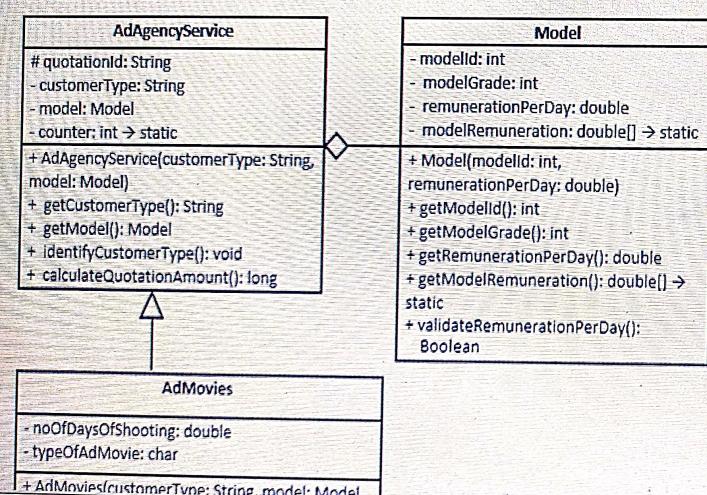


Problem Statement:**Description:**

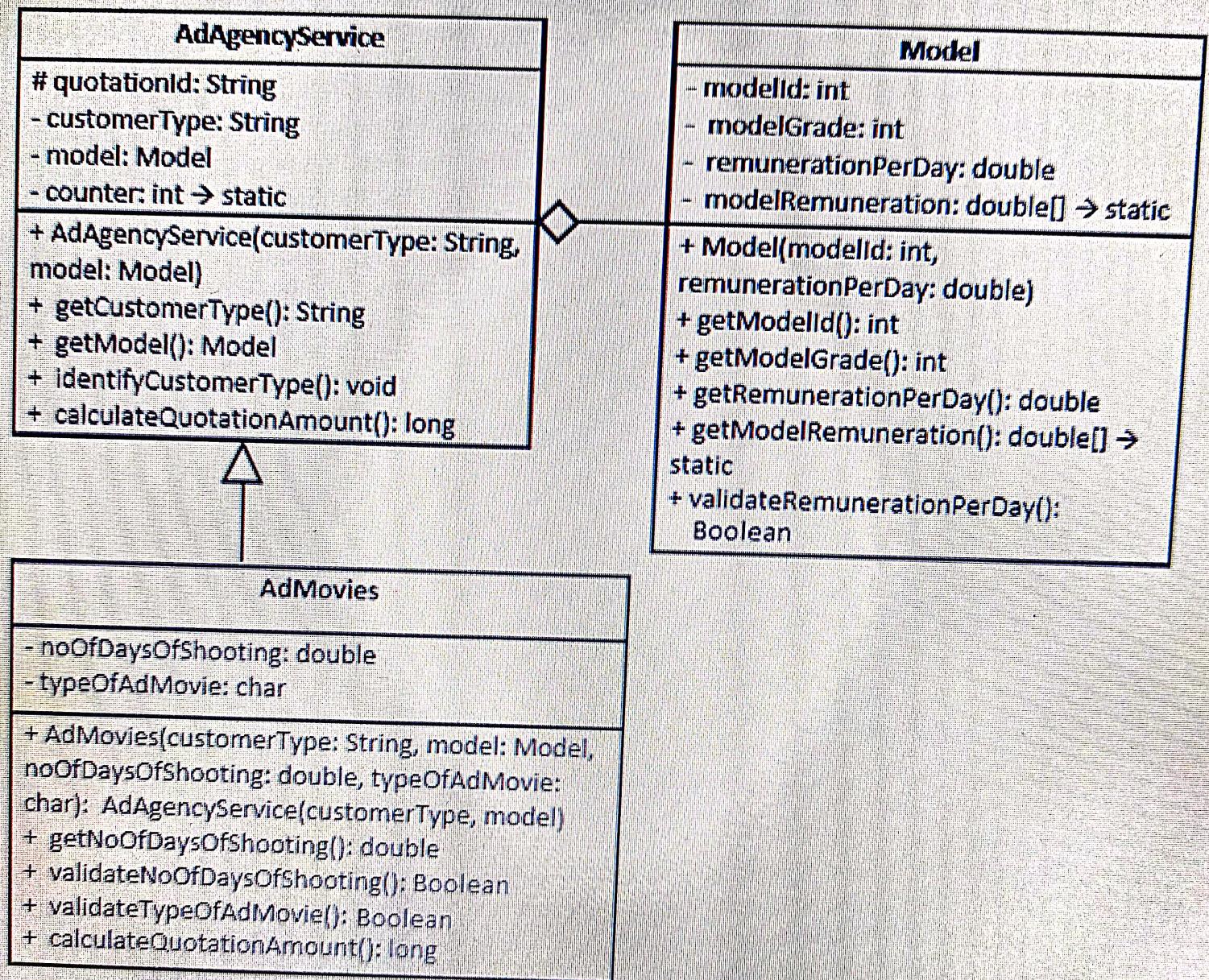
Advertise is an advertising movie shooting agency. It is popularly known for producing the creative advertisement movies. The automation of quotation generation system is depicted in the class diagram given below.

Implement the class diagram below to achieve the same.

Class Diagram:

Implement the class diagram below to achieve the same.

Class Diagram:



Notes:

- Do not include any extra instance/static variables and instance/static methods in the given classes
- Case insensitive comparison is required to be done unless specified explicitly
- Do not change any value or case of the given variables.
- Read notes and examples for better understanding of the logic
- In the derived classes, the order of passing arguments to the constructor would be the base class variables followed by the derived class variables

Implementation Details:

Class Name	Implementation Details
AdAgencyService	Fully Implemented
AdMovies	Partially Implemented
Model	Partially Implemented

Model class:**modelRemuneration:**

- This is a static array containing the following values-

modelRemuneration	{25000.0, 30000.0, 45000.0, 57500.0, 100000.0, 150000.0}
-------------------	--

<p>Note: <code>modelRemuneration</code> is initialized and provided. Hence no need to define</p> <p>Constructor:</p> <ul style="list-style-type: none">• This method initializes <code>modelId</code>, <code>modelGrade</code> and <code>remunerationPerDay</code>• Extract the first digit of <code>modelId</code> and set that digit as <code>modelGrade</code> <p>For Example: if the <code>modelId</code> is 43216 then the first digit, i.e. 4 would be assigned as the <code>modelGrade</code></p> <p>Assumption: <code>modelId</code> is a 5digit positive nonzero integer number, not more than 59999</p> <p>Note: No need to validate assumption</p> <p>validateRemunerationPerDay():</p> <ul style="list-style-type: none">• This method validates the <code>remunerationPerDay(double)</code>• Identify the <code>minimumLimit</code> and <code>maximumLimit</code> to validate <code>remunerationPerDay</code> using <code>modelRemuneration</code> array and <code>modelGrade</code>• In <code>modelRemuneration</code> array, considering <code>modelGrade</code> as position of <code>maximumLimit</code> of <code>remunerationPerDay</code>, and the <code>minimumLimit</code> is the value available in the previous position of the same array. <p>For example: When the <code>modelGrade</code> is 2, <code>maximumLimit</code> is 45000.0 (the value at index 2 in <code>modelRemuneration</code> array) and <code>minimumLimit</code> is 30000.0 (the value at index 1 in <code>modelRemuneration</code> array)</p> <ul style="list-style-type: none">• If the value of <code>remunerationPerDay</code> is more than the <code>minimumLimit</code> and less than or equal to <code>maximumLimit</code>, then return true• Otherwise return false <p>Example: If the <code>modelId</code> is 12345, <code>modelRemuneration</code> is {25000.0, 30000.0, 45000.0, 57500.0, 100000.0, 150000.0} and <code>remunerationPerDay</code> is 28000.0, then this method would return true</p>	
---	--

AdMovies class:

validateNoDaysOfShooting():

- This method validates `noDaysOfShooting(double)`
- If the value of `noDaysOfShooting` is between 2.0 to 100.0 (inclusive of boundary values) return `true`
- Otherwise return `false`

For Example: If `noDaysOfShooting` is 10.0 days, this method would return `true`

validateTypeOfAdMovie():

- This method validates `typeOfAdMovie (char)`
- If the value of `typeOfAdMovie` is either 'H' or 'A' or 'L', return `true`
- Otherwise return `false`

Note: Perform case-sensitive comparison

For Example: If `typeOfAdMovie` is 'H', this method would return `true`

calculateQuotationAmount():

- This method calculates and returns the `quotationAmount` as per the logic given below
- Invoke `validateTypeOfAdMovie()`, `validateNoDaysOfShooting()` and `validateRemunerationPerDay()`

Note: `validateRemunerationPerDay()` needs to be called from Model class

Note: validateRemunerationPerDay() needs to be called from Model class

- If all the above methods return true,
 - Identify *budget(int)* and *serviceCharge(double)* in percentage based on *typeOfAdMovie* as provided in the below table:

<i>typeOfAdMovie</i>	<i>budget</i>	<i>serviceCharge(%)</i>
'H'	2500000	25.25
'A'	2000000	17.5
'L'	1500000	12.75

- Identify *tax(double)* based on *customerType* as provided in the below table:

<i>customerType</i>	<i>tax(%)</i>
"Government"	0.0
"Public"	15.75
"Private"	20.25

Note: Perform case insensitive comparison

- Calculate *actualNumberOfShootingDays(long)* as mentioned below:

- o Calculate *actualNumberOfShootingDays*(long) as mentioned below:
 - If *noOfDaysOfShooting* has a fractional part set the value of *actualNumberOfShootingDays* to lower integral part

Example: if *noOfDaysOfShooting* is 10.50 then *actualNumberOfShootingDays* would be 10.0

Note: You may use **Math.ceil()**, **Math.floor()** and **Math.round()** methods of **java.lang.Math** package to implement the mathematical functions

- o Calculate *totalBudget*(double) using the below logic,

totalBudget = *budget* + *remunerationPerDay* * *actualNumberOfShootingDays*

Where *remunerationPerDay* is retrieved from the **Model** class

- o Calculate *serviceCharge* and *taxAmount* on *totalBudget* using the identified *serviceCharge* percentage, and *tax* percentage.
 - o Add them to *totalBudget*
 - o Invoke the **calculateQuotationAmount()** method of **AdAgencyService** to get the *baseQuotationAmount*
 - o Add this *baseQuotationAmount* to *totalBudget*
 - o Set the *quotationAmount* with the value of *totalBudget*
 - o Return *quotationAmount*
- Otherwise, return -1L

For Example: If *customerType* is "public", *typeOfAdMovie* is 'H' *noOfDaysOfShooting* is 10.0, *remunerationPerDay* is 35000.0 and *modelId* is 23534 then, *quotationAmount* would be 4019000 L

```
1 package mock;
2 //DO NOT MODIFY THE CODE PROVIDED TO YOU
3 public class AdAgencyService {
4     protected String quotationId;
5     private String customerType;
6     private Model model;
7     private static int counter = 1001;
8
9     public AdAgencyService(String customerType,Model model) {
10         this.customerType = customerType;
11         this.model = model;
12         identifyCustomerType();
13         quotationId = getCustomerType().substring(0, 3).toUpperCase()+counter;
14     }
15
16     public String getCustomerType() {
17         return customerType;
18     }
19     public Model getModel() {
20         return model;
21     }
22
23     public void identifyCustomerType() {
24         if(!(getCustomerType().equalsIgnoreCase("government") || getCustomerType().equalsIgnoreCase("public")
25             || getCustomerType().equalsIgnoreCase("private")))
26             customerType = "Private";
27     }
28 }
29
30     public long calculateQuotationAmount()
31     {
32         return 500L;
33     }
34 }
```

*AdAgencyService.java *AdMovies.java Model.java Tester.java

```
1 package mock;
2 //DO NOT MODIFY THE CODE PROVIDED TO YOU
3 public class AdMovies extends AdAgencyService{
4     private double noOfDaysOfShooting;
5     private char typeOfAdMovie;
6     public AdMovies(String customerType, Model model,double noOfDaysOfShooting,char typeOfAdMovie) {
7         super(customerType, model);
8         this.noOfDaysOfShooting = noOfDaysOfShooting;
9         this.typeOfAdMovie = typeOfAdMovie;
10    }
11    public double getNoOfDaysOfShooting() {
12        return noOfDaysOfShooting;
13    }
14    //To Trainees
15    public Boolean validateNoOfDaysOfShooting() {
16        //Implement your logic here
17        //Change the return statement accordingly
18        return null;
19    }
20    //To Trainees
21    public Boolean validateTypeOfAdMovie() {
22        //Implement your logic here
23        //Change the return statement accordingly
24        return null;
25    }
26    //To Trainees
27    @Override
28    public long calculateQuotationAmount(){
29        //Implement your code and change the return statement accordingly
30        return 0;
31    }
32 }
33 }
```

```
1 package mock;
2 //DO NOT MODIFY THE CODE PROVIDED TO YOU
3 public class Model {
4     private int modelId;
5     private int modelGrade;
6     private double remunerationPerDay;
7     private static double[] modelRemuneration = {25000.0,30000.0,45000.0,57500.0,100000.0,150000.0};
8     //To Trainees
9     public Model(int modelId,double remunerationPerDay) {
10         //Implement your code here
11     }
12     public int getModelId() {
13         return modelId;
14     }
15     public int getModelGrade() {
16         return modelGrade;
17     }
18     public double getRemunerationPerDay() {
19         return remunerationPerDay;
20     }
21     public static double[] getModelRemuneration() {
22         return Model.modelRemuneration;
23     }
24     //To Trainees
25     public Boolean validateRemunerationPerDay() {
26         //Implement your logic here
27         //Change the return statement accordingly
28         return null;
29     }
30 }
31 }
32 }
33 }
```

```
1 package mock;
2 import mock.AdMovies;
4
5 public class Tester
6 {
7
8     public static void main(String[] args)
9     {
10         Model modelobj = new Model(23534,35000.0);
11         AdMovies admoviesobj = new AdMovies("pubLIC",modelobj,10.0,'H');
12         System.out.println("Amount: "+admoviesobj.calculateQuotationAmount());
13     }
14
15 }
```