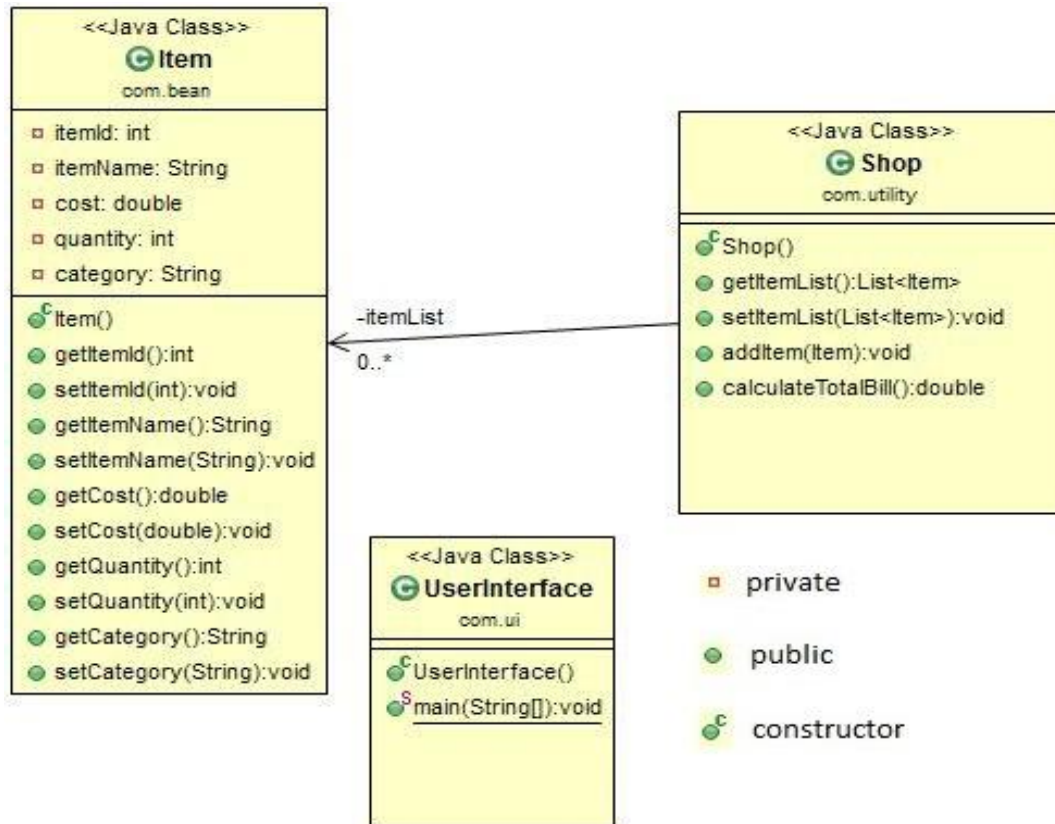


SET 2 Questions

1. Item Bill Calculation

ZEE Shop wants to provide calculate the delivery charges for item based on the category of the item purchased by their customers and then generate the bill. Help ZEE Shop to automate the above process with the help of java.



In the Item 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 **setCategory** method should set the category based on the total price. The total price needs to be calculated for each item based cost and quantity. The category should be set to the category attribute.

$\text{totalPrice} = \text{unitPrice} * \text{quantity}$

category should be set based on the below conditions

TotalPrice	Category
>10000	A
>=5000 && <=10000	B
<5000	C

The **addItem** method of Shop class accepts Item object as argument and adds it to the itemList. The item object passed should hold all the values including the category.

The **calculateTotalBill** method should return the total amount as double based on the cost , quantity and category for each item in the list. If the list is empty the method should return

-1.

Delivery charges should be calculated based on the category. For those items whose category is “A” then delivery charges would be 5% of the total Price(cost * quantity of the item) . If the item category is “B”, then delivery charges would be 10% of the total Price(cost * quantity of the item) and If the item category is “C”, then delivery charges would be 15% of the total Price(cost * quantity of the item)

[Bill Amount= For all the item’s ,sum of [cost* quantity + deliverycharges of each item] .

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

Sample Input and Output:

Enter the number of items

3

Enter the item details

Enter the item id:

1

Enter the item name:

Furniture

Enter the cost:

500

Enter the quantity:

50

Enter the item details

Enter the item id:

2

Enter the item name:

Notebook

Enter the cost:

20

Enter the quantity:

500

Enter the item details

Enter the item id:

3

Enter the item name:

Bags

Enter the cost:

100

Enter the quantity:

9

Calculated Bill Amount:47600.0

2. Vehicle List

Insurance Bazaar is an third party insurance company wants to categorize the vehicles based on their type (TwoWheeler or FourWheeler). Help them by writing a Java program to store the details of vehicles as key-value pair with the type of vehicle as key and the list of vehicle ids as values.

Consider the class given below,



private

public

constructor

The **public void addVehicleDetails (String vehicleType, int vehicleId)** method should add vehicle type as key and add the vehicle id in the list as values.

If the vehicle type (key) is already available in the map then add the vehicle id into the list which is already available as value.

If the vehicle type(key) is not available in the map then add the vehicle type as key and create a new list as value corresponding to the key and add the vehicle id into that list.

The **public int findLastvehicleId(String vehicleType)** method should return the vehicle id available as the last element in the list corresponding to the vehicle type (key) passed as argument (String) to this method.

In the Map, if no key matches (**case-insensitive**) with the vehicleType which was passed as an argument, then the method has to return -1.

Note: vehicleType is **case-insensitive**. Use appropriate collection to maintain insertion order while adding elements into Map and List.

The signature of the above methods are given as part of code skeleton, do not change the method signature.

Sample keys and values of Vehicle Details:

Key(String)	Value (List<Integer>)
TwoWheeler	[101,104,103]
FourWheeler	[105,102]

In main class create a main method with the menu as described in sample Input and Output. When user select the **option 1(Add Vehicle Details)**, it should get the Vehicle Type (String) and Vehicle Id (int) from the user and invoke the method **addVehicleDetails (String vehicleType,int vehicleId)**.

When user select the **option 2(Last Vehicle Id)**, it should get the Vehicle Type (String) as input from the user and invoke the method **findLastvehicleId(String vehicleType)** and display the vehicle id returned from the method.

When user select the **option 3(Exit)**, display the message "**End**" and terminate the program.

Sample Input and Output 1:

1. Add Vehicle Details
2. Last Vehicle Id
3. Exit

Enter your choice

1

Enter the Vehicle Type

TwoWheeler

Enter the Vehicle Id

101

1. Add Vehicle Details
2. Last Vehicle Id
3. Exit

Enter your choice

1

Enter the Vehicle Type

FourWheeler

Enter the Vehicle Id

105

1. Add Vehicle Details
2. Last Vehicle Id
3. Exit

Enter your choice

1

Enter the Vehicle Type

TwoWheeler

Enter the Vehicle Id

104

1. Add Vehicle Details
2. Last Vehicle Id
3. Exit

Enter your choice

1

Enter the Vehicle Type

TwoWheeler

Enter the Vehicle Id

103

1. Add Vehicle Details
2. Last Vehicle Id
3. Exit

Enter your choice

1

Enter the Vehicle Type

FourWheeler

Enter the Vehicle Id

102

1. Add Vehicle Details

2. Last Vehicle Id

3. Exit

Enter your choice

2

Enter the Vehicle Type to search the last element

twowheeler // note vehicle Type is case-Insensitive

103

1. Add Vehicle Details

2. Last Vehicle Id

3. Exit

Enter your choice

2

Enter the Vehicle Type to search the last element

FourWheeler // note vehicle Type is case-Insensitive

102

1. Add Vehicle Details

2. Last Vehicle Id

3. Exit

Enter your choice

3

End

3.Memory Recall

Science teacher wants to teach about rainbow. She wanted the students to remember the colors in the rainbow. She used the acronym VIBGYOR, V-Violet, I-Indigo, B-Blue, G-Green, Y-Yellow, O-Orange, R-Red. Help her to test the students using java program, whether they remember the colors or not.

Note: Characters has to be in capital letters only, or else print "Invalid Input". Only VIBGYOR characters can be used, or else print "Invalid Input". Number of letters should not exceed 7, or else print "Invalid Input".

Sample Input1:

Enter the acronym

VIY

Sample output1:

Violet

Indigo

Yellow

Sample Input2:

Enter the acronym

OORRG

Sample output2:

Orange

Orange

Red

Red

Green

Sample Input3:

Enter the acronym

BRYL

Sample output3:

Invalid Input

Sample Input4:

Enter the acronym

bgo

Sample output4:

Invalid Input

4. Password Validator

Mary owns a departmental store. She is planning to create a website for her store. In order to create login to her customer, she collects informations of her customer such as customeName and Password. Customers wants to confirm the password to complete the registration.

If the password and confirm password is same the system should display "**Registered Sucessfully**". If the password and confirm password is different the system should alert the customer with the message "**Password entered does not match**". So help her to automate by writing a Java program which will throw a user defined exception "**PasswordMismatchException**".

Partial Code is given to do the above task. In the Validator class, you are provided with the method `public static boolean validatePassword(String password,String confirmpassWord)`. Logic for validating the password is partially given. Complete the code as follows

If the password matches (case-sensitive) exactly with confirmPassword this method should return true.

If the password doesnt matches (case-sensitive) with confirmPassword this method should throw a "`PasswordMismatchException`".

From the main method invoke the validatePassword method and handle the exception and print the appropriate message.

Sample Input and Output 1:

Enter the customer Name

Jaison

Enter the password

Jai@1234

Re-enter the password to confirm

Jai@1234

Registered Sucessfully

Sample Input and Output 2:

Enter the customer Name

Jaison

Enter the password

Jai@1234

Re-enter the password to confirm

JAi@1234

Password entered does not match