Q.1. Vehicle-Loan-Insurance - Use Interface

Universal Loan and Insurance Providers is one of the fastest growing organization. It needs to automate the transactions performed in the organization. They need to automate the process of issuing loan and insurance coverage for vehicles.

Partial code is provided to do this. Donâ€™t change the skeleton. Do the additions wherever necessary.

You are provided with a public class Vehicle with private attributes :

- String vehicleNumber
- String modelName
- String vehicleType
- double price

Appropriate public getters and setters are already written.

You are also provided with a 4 argument constructor with arguments â€“vehicleNumber, modelName, vehicleType and price.

Note that the vehicleType can take the values as â€œ4 wheelerâ€□ or â€œ3 wheelerâ€□ or â€œ2 wheelerâ€□.

Write a public interface Loan with an abstract method â€œdouble issueLoan()â€□.

Write a public interface Insurance with an abstract method â€œdouble takeInsurance ()â€□.

The above class Vehicle should implement the Interfaces Loan and Insurance.

Provide the implementation for issueLoan method based on the type of Vehicle.

If the vehicleType is â€œ4 wheelerâ€□, the eligible loan amount is 80% of its price.

If the vehicleType is â€œ3 wheelerâ€□, the eligible loan amount is 75% of its price.

If the vehicleType is â€œ2 wheelerâ€□, the eligible loan amount is 50% of its price.

Provide the implementation for takeInsurance based on price of vehicle.

If the vehicle price is less than or equal to 150000 insurance amount is 3500.

If the vehicle price is greater than 150000 and less than or equal to 300000 insurance amount is 4000.

If the vehicle price is greater than 300000 insurance amount is 5000.

You are provided with a public class Main which has the main method.

Check the correctness of the methods written in these classes.

Q.2. # Book Manipulation

The district central library needs an application to store book details of their library. The clerk who has all the rights to add a new book,search for any book,display the book details and should update the count of total number of books.

You are provided with a Book with the following private attributes:

- int isbnno
- String bookName
- String author

Needed getters and setters are written.

Create a class Library with the following private attribute:

- ArrayList<Book> bookList = new ArrayList<Book>();

Also provide the necessary setter and getter methods.
Include the following public methods:

- `void addBook(Book bobj)` - This method should add the book object to the booklist.
- `boolean isEmpty()` - This method should return true if the booklist is empty else return false.
- `ArrayList<Book> viewAllBooks()` - This method should return the list of books maintained in the library.
- `ArrayList<Book> viewBooksByAuthor(String author)` - This method should return a list of books written by the author passed as argument. When you display an empty list it should print the message "The list is empty".
- `int countnoofbook(String bname)` - this method should return the count of books with the name passed as argument.

Write a Main class to test the above functionalities.

> Sample Input and Output:

```
    1.Add Book
    2.Display all book details
    3.Search Book by author
    4.Count number of books - by book name
    5.Exit
    Enter your choice:
```

```
1
Enter the isbn no:
123
Enter the book name:
Java
Enter the author name:
Bruce Eckel
1.Add Book
2.Display all book details
3.Search Book by author
4.Count number of books - by book name
5.Exit
Enter your choice:
1
Enter the isbn no:
124
Enter the book name:
C++
Enter the author name:
Eric Nagler
1.Add Book
2.Display all book details
3.Search Book by author
4.Count number of books - by book name
5.Exit
Enter your choice:
3
Enter the author name:
Henry
None of the book published by the author Henry
1.Add Book
2.Display all book details
3.Search Book by author
4.Count number of books - by book name
5.Exit
Enter your choice:
3
Enter the author name:
Eric Nagler
ISBN no: 124
Book name: C++
Author name: Eric Nagler
1.Add Book
2.Display all book details
3.Search Book by author
4.Count number of books - by book name
5.Exit
Enter your choice:
5
```

Q.3. # Count of Each Words

Miss.Jane, an experienced English professor, gives practice tests to her students to improve their written skills. Everyday students write an article and they submit it to Jane.

Jane is particular that the students use only special characters like , ; : . ? ! in the article.

**Note : Using the above mentioned special characters will help to split the words in a sentence.
All other special characters when used will be considered as a part of the word itself.**

She counts the total number of words used and the count of each word in the article. Based on this analysis she gives her feedback to the students.

Difficulty arises when the number of students increase. So she wanted to automate the process in the following format. Help her to write a java program to display the words and the number of times it has been used in the article and to display the words using lower case and in alphabetical order.

> Sample Input Output:

```
Enter Student's Article
hello Hello HEllo hi hi: hi! Welcome,   welcome
Number of words 8
Words with the count
hello: 3
hi: 3
welcome: 2
```

Q.4. # Insurance Bazaar

Insurance Bazaar is developing an online website for showcasing various types of Insurance policies to their customers based on their needs. There are various types of Insurances provided by different insurance agencies. The admin of Insurance Bazaar wants to add different insurance policy names like Max Bupa Health Insurance, SBI Health Insurance, IFFCO Tokio Two Wheeler Insurance and New India Assurance Two Wheeler Insurance to his database with Policy ID as the Tags.

Customers can view the names of all the polices available in Insurance Bazaar based on the type of insurance.

Write a Java program to simulate this scenario. Key (Policy ID) should be an Integer and Value (Insurance policy name) should be a String. The key-value should be sorted based on the key. Use the appropriate Collection to Store all these details and display.

The addPolicyDetails method should add the Policy ID and the Policy name into the appropriate map.

The searchBasedOnPolicyType method should return the list of Insurance policy ID depending on the input provided. This method takes the input as string (Input can be either Health or Two Wheeler).

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

Sample Policy ID and policy names:

Policy ID	Policy names
10654	Max Bupa Health Insurance
10321	SBI Health Insurance
20145	IFFCO Tokio Two Wheeler Insurance
20165	New India Assurance Two Wheeler Insurance
10110	Reliance Health Insurance

> Sample Input and Output1

```
Enter the no of Policy names you want to store
5
Enter the Policy ID
10654
Enter the Policy Name
Max Bupa Health Insurance
Enter the Policy ID
10321
Enter the Policy Name
SBI Health Insurance
Enter the Policy ID
20145
Enter the Policy Name
IFFCO Tokio Two Wheeler Insurance
Enter the Policy ID
20165
Enter the Policy Name
New India Assurance Two Wheeler Insurance
Enter the Policy ID
10110
Enter the Policy Name
Reliance Health Insurance
10110 Reliance Health Insurance
10321 SBI Health Insurance
10654 Max Bupa Health Insurance
20145 IFFCO Tokio Two Wheeler Insurance
20165 New India Assurance Two Wheeler Insurance
Enter the policy type to be searched
Two Wheeler
20145
20165
```

> Sample Input and Output2

```
Enter the no of Policy names you want to store
4
```

```
    Enter the Policy ID
    10654
    Enter the Policy Name
    Max Bupa Health Insurance
    Enter the Policy ID
    10321
    Enter the Policy Name
    SBI Health Insurance
    Enter the Policy ID
    20145
    Enter the Policy Name
    IFFCO Tokio Two Wheeler Insurance
    Enter the Policy ID
    20165
    Enter the Policy Name
    New India Assurance Two Wheeler Insurance
    10321 SBI Health Insurance
    10654 Max Bupa Health Insurance
    20145 IFFCO Tokio Two Wheeler Insurance
    20165 New India Assurance Two Wheeler Insurance
    Enter the policy type to be searched
    Health
    10321
    10654
```

Q.5. # PhoneBook Manipulation

Airone mobile services needs to store their customer details in the company portal. The details are customer's first and last name, phone number, and email address. Help the company develop an application to maintain the details of their customer systematically.

You are provided with a class Contact with the following attributes as private.

- String firstName

- String lastName

- long phoneNumber

- String emailId

A 4 argument constructor and appropriate setters and getters to store and retrieve the details are also provided.

Create a class PhoneBook with a private attribute

- List<Contact> phoneBook = new ArrayList<Contact>();

Write the getters and setters.

Write the following methods in the PhoneBook class.

- `public void addContact(Contact contactObj)` â€“ This method should add the contact object to the phoneBook list.
- `public List<Contact> viewAllContacts()` â€“ This method should return the list of all contacts available.
- `public Contact viewContactGivenPhone(long phoneNumber)` - This method should return the contact details which has the phoneNumber given as parameter.
- `public boolean removeContact(long phoneNumber)` - This method should remove the contact details which has the phoneNumber given as parameter. If removed return true. Else if the phone number is not available return false.

Write a class Main with the main method. Create the menu as shown in the Sample Input and Output and invoke the corresponding methods as per the choice provided.

> Sample Input and Output 1:

```
Menu
1.Add Contact
2.Display all contacts
3.Search contact by phone
4.Remove contact
5.Exit
Enter your choice: 1
Add a Contact:
Enter the First Name: John
Enter the Last Name: Michael
Enter the Phone No.: 9787878900
Enter the Email: John@gmail.com
Menu
1.Add Contact
2.Display all contacts
3.Search contact by phone
4.Remove contact
5.Exit
Enter your choice: 1
Add a Contact:
First Name: Jhonty
Last Name: Rhodes
Phone No.: 9787888900
Email: Jhonty@gmail.com
Menu
1.Add Contact
2.Display all contacts
3.Search contact by phone
4.Remove contact
5.Exit
Enter your choice: 2
The contacts in the List are:
First Name: John
Last Name: Michael
Phone No.: 9787878900
Email: John@gmail.com
First Name: Jhonty
Last Name: Rhodes
```

```
Phone No.: 9787888900
Email: Jhonty@gmail.com
Menu
1.Add Contact
2.Display all contacts
3.Search contact by phone
4.Remove contact
5.Exit
Enter your choice: 3
Enter the Phone number to search contact:9787888900
The contact is:
First Name: Jhonty
Last Name: Rhodes
Phone No.: 9787888900
Email: Jhonty@gmail.com
Menu
1.Add Contact
2.Display all contacts
3.Search contact by phone
4.Remove contact
5.Exit
Enter your choice: 4
Enter the Phone number to remove :9787888900
Do you want to remove the contact (Y/N): Y
The contact is successfully deleted.
Menu
1.Add Contact
2.Display all contacts
3.Search contact by phone
4.Remove contact
5.Exit
Enter your choice: 5
```