R2S
Resource Software Solution

JAVA BASIC

Lab Guides

Java Collections

Objectives:

- Understand basic concept about Collections in Java and Generic.

Lab Specifications:

Create a class to represent a **DemoArrayList** type in java, which should have following:

- Add some element to a list.
- Print out all of the elements.
- Retrieve a special element.
- Remove an element at a given index
- Modify the element at a given index.
- Sort a list.

Create a new class named Test with a main() method.

Functional Requirements:

- Understand how to create a new ArrayList and use its methods.
- Explain the meaning of line 41 in class **ArrayListDemo** if we instead of use List<String> Collections.sort(listSubject); by another List<Object>?

Guidelines:

- Step1. Open Eclipse IDE
- Step2. Create a new project named **JAVA.S.L1002**
- Step3: Create package excercise.arraylist that contains ArrayListDemo class:

ArrayListDemo class source code:

```
1.    package excercise.arraylist;
2.    import java.util.ArrayList;
3.    import java.util.Collections;
4.    public class DemoArrayList {
5.    /**
6.    The main method
7.    @param args
8.    */
9.
10.
11.   public static void main(String[] args) {
12.   List<String> listSubject = new ArrayList<>();
13.
14.   listSubject.add("IOS");
```

```java
listSubject.add("Java");
listSubject.add("PHP");
listSubject.add("Android");
listSubject.add("Python");
listSubject.add("Java"); // list allow contains duplicate record
listSubject.add("C");

// display all subject
System.out.print("There are "+listSubject.size()+" subject: ");
display(listSubject);

//Retrieve the an element at a given index
String popularLanguage = listSubject.get(1);
System.out.println("This is a language program popular in the world:" + popularLanguage);

//Remove an element at a given index
listSubject.remove(listSubject.size()-1);
System.out.print("There are " + listSubject.size()+ "subject after remove a element : ");
display(listSubject);

// Modify the element at a given index
listSubject.set(listSubject.size()-1, "Ruby");
System.out.print("List after modify: ");
display(listSubject);

// sort a list string.
Collections.sort(listSubject);
System.out.print("List after sort: ");
display(listSubject);
}

/**
This method to show all element in list.
@param listSubject
*/
  public static void display(ArrayList<String> listSubject) {
    for(String subject : listSubject) {
                System.out.print(subject + ", ");
    }
    System.out.println();
    }
  }
```

- » Step4: How to run

In Eclipse window | select **Run Test** or right-click **Run as..**:

Outputs:

```
There are 7 subject: IOS, Java, PHP, Android, Python, Java, C,
This is a language program popular in the world: Java
There are 6 subject after remove a element: IOS, Java, PHP, Android, Python, Java,
List after modify: IOS, Java, PHP, Android, Python, Ruby,
List after sort: Android, IOS, Java, PHP, Python, Ruby,
```

Assignment: Project Guides

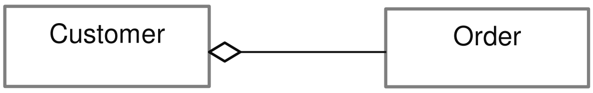
Objectives:

» To understand basic concept of ArrayList

» To declare and use common methods of ArrayList

Specifications:

For the class hierarchy is as follows, the trainee let's create the java classes install this class diagram to be able to relationship between it.



Create a class called Customer is designed as follows:

» Four private instance variables: name (String), phone number (String), address (String), and list of orders (of the class Order you have just created).

» Default constructor.

» One constructor to initialize the name, phone number, address and list of orders with the given values. Getter and setter methods.

And, a class called Order is designed as follows:

» Two private instance variables: number (String), date (Date).

» Default constructor.

One constructor which constructs an instance with the values given. Gettes and setter methods.

Business Rules:

» *phone number*: correct phone format

» *order number*: number string with length equals 10.

Functional Requirements:

a. Write a method to enter customer data with a her/his list order, users will be asked whether to continue or finish.

Finish typing data when the user selects 'n' or 'N' (method named: *public void createCustomer()).*

b. The program has a method to get all of customers (method named *public List<Customer> findAll()*).

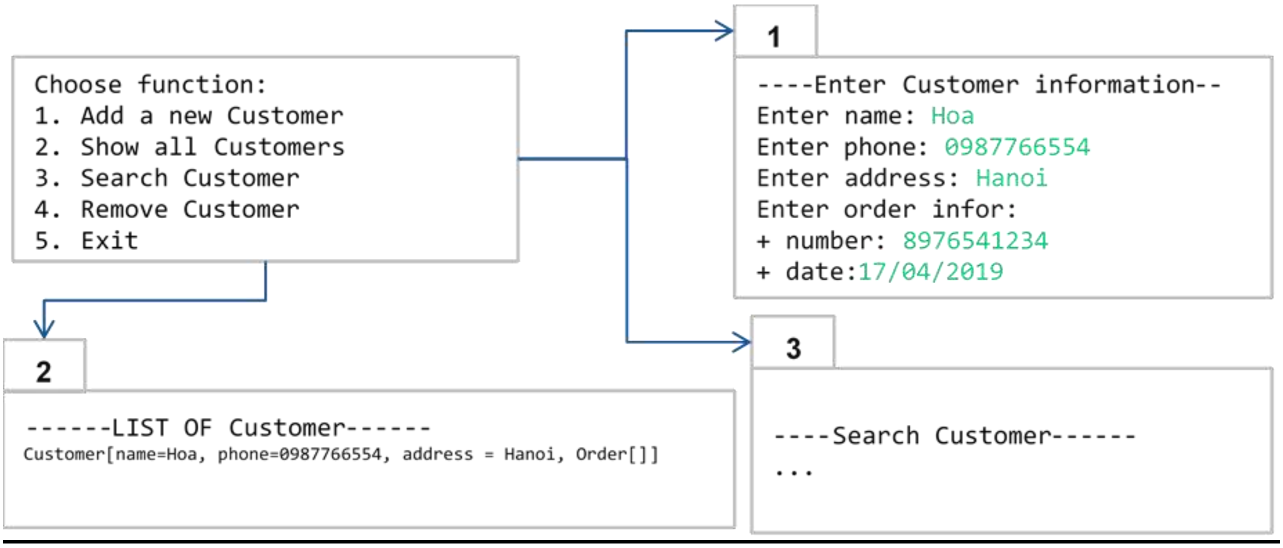
c. The program has a method to display data in the following format: (method named: *public void display(List<Customer>)*).

Customer Name	Address	Phone Number	OrderList

d. The program has a method to search order by customer (method named *public List<Order> search(String phone)*).

e. Write a method to remove a specific customer by phone number from customer file (method named *public boolean remove(String phone)*).

Screen Requirements:



```
Choose function:
1. Add a new Customer
2. Show all Customers
3. Search Customer
4. Remove Customer
5. Exit
```

```
----Enter Customer information--
Enter name: Hoa
Enter phone: 0987766554
Enter address: Hanoi
Enter order infor:
+ number: 8976541234
+ date:17/04/2019
```

```
------LIST OF Customer------
Customer[name=Hoa, phone=0987766554, address = Hanoi, Order[]]
```

```
----Search Customer------
...
```

Guidelines:

- Create a new project named **SaleManagement**
- Package **r2s.training.entities** that contains two classes: Customer, Order.
- Package **r2s.training.services** that contains two classes: CustomerService, OrderService.
- Package **r2s.training.utils** that contains two classes: Validator, Constants.
- Package **r2s.training.main** contains class: **Test** that contains main() method to run program.