SE116 – LAB#10 2021-2022 SPRING

Aim: Understanding the use of interface; understanding the concepts of exception and exception handling.

1. To simulate an "**online shopping**" system, implement the design stated below:

- Write an interface `Advertisable` with the following `abstract` method: `void advertise();`

- Write a class `Computer` that `implements Advertisable`. The class `Computer` represents a computer to be advertised in our "online shopping" system. The class `Computer` has the following private data members: `int ramSize`, `String operatingSystem`, `String screenResolution`, and `int hardDriveSize`. The class `Computer` defines the method `advertise` to print out the values of the data members (information) of a computer.

- Write a class `AdvertOwner` that `implements Advertisable`. The class `AdvertOwner` represents an owner of an advert for a computer. The class `AdvertOwner` has the following private data members: `String fullName`, `int ADVERT_OWNER_ID`, and `String contactNumber`. The class `AdvertOwner` defines the method `advertise` to print out the corresponding information.

- Write a class `Advert` that `implements Advertisable`. The class `Advert` represents an advert for a computer. The class `Advert` has the following private data members: `int ADVERT_NO`, `AdvertOwner advertOwner`, `Computer computer`, and `int price`. The price value cannot be negative. The class `Advert` defines the method `advertise` to print out the corresponding information.

- Write a class `ComputerOwner` that `extends AdvertOwner`. The class `ComputerOwner` represents an individual person as the owner of an advert for a computer. The class `ComputerOwner` has the following extra private data member: `String occupation`. The class `ComputerOwner` overrides the method `advertise` to print out all information.

- Write a class `OnlineRetailer` that `extends AdvertOwner`. The class `OnlineRetailer` represents an online retail store as the owner of an advert for a computer. The class `OnlineRetailer` has the following extra private data member: `String webAddress`. The class `OnlineRetailer` overrides the method `advertise` to print out all information.

- Write a class `Test` that contains the following `public static` methods: `void menu()`, and `void main(String[] args)`. The definition of the method `menu` is given below:

```
public static void menu(){
    System.out.println("<<<<<<<ONLINE SHOPPING SEARCH PROGRAM>>>>>>>" );
    System.out.println("PRESS 1 to search for a computer by the properties you seek" );
    System.out.println("PRESS 2 to search for a computer by checking an advert owner" );
    System.out.println("PRESS 3 to quit" );
    System.out.println("\nPlease enter your choice to continue..." );
}
```

Define the method `main` in the following manner:

In `main`, at first, construct the following `ArrayList` instances: an `ArrayList` of `AdvertOwner`, an `ArrayList` of `Computer`, and an `ArrayList` of `Advert`. Create many test instances from `ComputerOwner`, `OnlineRetailer`, `Computer` and `Advert`. Add the references of these objects into the related `ArrayList` in the program. Then, display the menu options.

Considering the menu options, the response of the program to the option 1 will simulate the following scenario: At first, the program reads from the user some related "`ramSize`" and "`screenResolution`" criteria for a computer to be searched among the adverts. Then, the program prints out all information of the related adverts available. Among these possible results, the user will decide on the computer to buy and finally will filter the results by entering the corresponding "`ADVERT_NO`" to be searched in the system.

The response of the program to the option 2 of the menu will simulate the following scenario: At first, the program reads from the user the full name of an advert owner to be searched among the adverts. Then, the program prints out all information of the related adverts available. Among these possible results, the user will decide on the computer to buy and finally will filter the results by entering the corresponding "`ADVERT_NO`" to be searched in the system.

Regarding the option 1 of the menu, the user will enter "`screenResolution`" property in the following format: x*y (such as 3840*2160). The user can sequentially enter different properties (criteria) in each line; -1 will be used to stop adding new properties to search. The user will enter an integer value for the "`ramSize`" property. The user can sequentially enter many values (criteria) each separated by a space character; -1 will be used to stop adding new criteria to search. The program will search the list of adverts for all possible different combinations of "`screenResolution`" and "`ramSize`" properties. The results of the search operation must satisfy both of the "`screenResolution`" and "`ramSize`" criteria for a computer.

- To illustrate an exception and a corresponding handling mechanism, write the program code in such a manner that the program reads an integer value as the choice of the user among the option numbers in menu. It is expected that the user would enter a number; but, what if the user enters a different character or a string? Use an appropriate exception handling mechanism in case of that the user can press any key as the choice among the menu options. To illustrate such an exception, consider the following scenario: The user enters the key 'Z' instead of the key '2'. In such cases, let the program handle the exception and make the user re-enter a new choice until the value is valid.

TO DO @ HOME:

2. Modify your project implemented in Question#**1** in the following manner:

In this part, you are expected to use object serialization and files.
To write a serialized object into a file, use the method **writeObject** of class **ObjectOutputStream**. To read corresponding data from a file in order to deserialize the corresponding object in program memory, use the method **readObject** of class **ObjectInputStream**.

a. According to the corresponding I/O operations mentioned, modify the required classes of Question#1 to **implement** the interface **Serializable**.

b. To test the modified project, in main, at first, create 5 instances of class `Computer` and after serializing these objects write their data into a file (computers.ser) in your workspace. Then, create 5 instances of class `OnlineRetailer` and after serializing these objects write their data into a file (onlineRetailers.ser) in your workspace. Then, create 5 instances of class `ComputerOwner` and after serializing these objects write their data into a file (computerOwners.ser) in your workspace. At this moment, if everything is successful, you will have three files that store the data of corresponding objects. Therefore, now, you can read the required data from each file into your program and deserialize the corresponding objects. Using these deserialized objects, create sample instances of class `Advert` to be added into an instance of an `ArrayList` of `Advert`. At this moment, if everything is successful, you can display the menu and search for a computer in the list.