FEU Institute of Technology
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

INFORMATION TECHNOLOGY EDUCATION DEPARTMENT

ITPROG3
(Object Oriented Programming)

EXERCISE 6

Declaring classes

Francisco, Elline Joy C.
W293
May 17, 2017

I. Objectives:

At the end of the experiment, students must be able to:

Cognitive
a) understand how to create classes
b) understand attributes and methods

Psychomotor:
a) construct a program using classes and objects
b) compile and debug the error of the program

Affective
a) appreciate the concept behind this experiment

II. BACKGROUND INFORMATION

To define a class, we write:

```
<modifier> class <name> {
        <attributeDeclaration>*
        <constructorDeclaration>*
        <methodDeclaration>*
}
```

– where

- <modifier> is an access modifier, which may be combined with other types of modifier.

```
public class StudentRecord {

//we'll add more code here later

}
```

– where,

- public - means that our class is accessible to other classes outside the package
- class - this is the keyword used to create a class in Java
- StudentRecord - a unique identifier that describes our class

III. EXPERIMENTAL PROCEDURE:

1. Write a program to create a room class, the attributes of this class is roomno, roomtype, roomarea and ACmachine. In this class the member functions are setdata and displaydata.

2. ***Address Book Entry.*** Your task is to create a class that contains an address book entry. The following table describes the information that an adressbook entry has.

Attributes/Properties	Description
Name	Name of the person in the addressbook
Address	Address of the person
Telephone Number	Telephone number of the person
Email Address	Person's Email address

   a. Provide the necessary accessor and mutator methods for all the attributes.
   b. Constructors

3. ***AddressBook.*** Create a class address book that can contain 100 entries of AddressBookEntry objects (use the class you created in the first exercise). You should provide the following methods for the address book.
   a. Add entry
   b. Delete entry
   c. View all entries
   d. Update an entry

2.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package addressbook;

/**
 *
 * @author 201510968
 */
class StudentRecord {
    private String name;
   private String address;
   private String email;
   private int telNo;

    public StudentRecord(){
```

```
    this.name="";
    this.address="";
    this.email="";
    this.telNo=0;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}

public String getEmail() {
    return email;
}

public void setEmail(String email) {
    this.email = email;
}

public int getTelNo() {
    return telNo;
}

public void setTelNo(int telNo) {
    this.telNo = telNo;
}
//custom methods here
public void deleteEntry(){
        this.name="";
        this.address="";
        this.email="";
        this.telNo=0;
}

public void viewEntry(){

        System.out.println("Name    : " + this.name);
        System.out.println("Address : " + this.address);
        System.out.println("Email   : " + this.email);
        System.out.println("Tel No. : " + this.telNo);
```

```
        System.out.println();

    }

}
```

3.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package addressbook;

import java.io.BufferedReader;
import java.io.InputStreamReader;

/**
 *
 * @author 201510968
 */
public class AddressBook {

   public static int displayMenu()throws Exception{
      BufferedReader br=new BufferedReader (new InputStreamReader(System.in));

      System.out.print("\nADDRESS BOOK MAIN MENU" +
                  "\n[1] Add Entry" +
                  "\n[2] Delete Entry" +
                  "\n[3] View Entry" +
                  "\n[4] Update Entry" +
                  "\n[5] Exit" +
                  "\n    Enter an option --> ");
      int option=Integer.parseInt(br.readLine());
      return option;
   }
//-------------------------------------------------------------------------------
   public static int addEntry(StudentRecord[] sr, int i)throws Exception{
      BufferedReader br=new BufferedReader (new InputStreamReader(System.in));
      char answer;

      do{
      sr[i]=new StudentRecord();
      System.out.println("\n\n\tADD ENTRY");
      System.out.print("Name       --> ");
      String name=br.readLine();
      System.out.print("Address    --> ");
      String address=br.readLine();
      System.out.print("Email      --> ");
      String email=br.readLine();
```

```
        System.out.print("Contact No. --> ");
        int telNo=Integer.parseInt(br.readLine());
        //set the values
        sr[i].setName(name);
        sr[i].setAddress(address);
        sr[i].setEmail(email);
        sr[i].setTelNo(telNo);
        System.out.print("\nInput again? [y/n] --> ");
        answer=(char)System.in.read();
        System.in.read();
        //if(answer=='y' || answer=='Y') i++;
        i++;
        }while(answer=='y' || answer=='Y');
        return i;

    }
//the main method----------------------------------------------------------------------
    public static void main(String[] args) throws Exception{
        BufferedReader br=new BufferedReader (new InputStreamReader(System.in));
        StudentRecord[] sr=new StudentRecord[5];
        String nameToCompare;
        int opt,updateOption,i=0;
        char answer;
        boolean found;

        do{
        opt=displayMenu();
        switch(opt){
            case 1: i=addEntry(sr,i);
                    break;

            case 2: System.out.println("\n\n\tDELETE ENTRY");
                    System.out.print("Enter name to delete --> ");
                    nameToCompare=br.readLine();
                    found=false;
                    for(int index=0;index<i;index++){
                        if(nameToCompare.equalsIgnoreCase(sr[index].getName())){
                            sr[index].deleteEntry();
                            System.out.println("Student " + nameToCompare + " is deleted!");
                            found=true;
                        }
                    }
                    if (found==false){
                        System.out.println("No match found!");
                    }

                    break;

            case 3: System.out.println("\n\n\tVIEW ENTRY");
                    System.out.print("Enter name to view --> ");
                    nameToCompare=br.readLine();
```

```
                found=false;
                for(int index=0;index<i;index++){
                   if(nameToCompare.equalsIgnoreCase(sr[index].getName())){
                      sr[index].viewEntry();
                      found=true;
                   }
                }
                if (found==false){
                   System.out.println("No match found!");
                }
                break;

            case 4: System.out.println("\n\n\tUPDATE ENTRY");
                System.out.print("Enter name to update --> ");
                nameToCompare=br.readLine();
                found=false;
                for(int index=0;index<i;index++){
                   if(nameToCompare.equalsIgnoreCase(sr[index].getName())){
                      sr[index].viewEntry();
                      found=true;
                      do{
                      System.out.println("[1.] Name    : " + sr[index].getName());
                      System.out.println("[2.] Address : " + sr[index].getAddress());
                      System.out.println("[3.] Email   : " + sr[index].getEmail());
                      System.out.println("[4.] Tel No. : " + sr[index].getTelNo());
                      System.out.println("[5.] Back to Main Menu");
                      System.out.print("Please enter an option to update --> ");
                      updateOption=Integer.parseInt(br.readLine());
                      switch(updateOption){
                         case 1: System.out.print("Enter new name --> ");
                               sr[index].setName(br.readLine()); break;
                         case 2: System.out.print("Enter new address --> ");
                               sr[index].setAddress(br.readLine()); break;
                         case 3: System.out.print("Enter new email --> ");
                               sr[index].setEmail(br.readLine()); break;
                         case 4: System.out.print("Enter new telephone number --> ");
                               sr[index].setTelNo(Integer.parseInt(br.readLine())); break;
                      }
                      }while(updateOption!=5);
                   }
                }
                if (found==false)
                   System.out.println("No match found!");
                break;

            case 5: System.out.println("\n\tGoodbye!");
         }


         }while(opt!=5);
      }
```

```
}
```

```
        ADD ENTRY
Name         --> Renalyn Joy Francisco
Address      --> 30 Blk. 17 Brgy. Bungad QC
Email        --> renalynjoyfrancisco@gmail.com
Contact No. --> 9558253

Input again? [y/n] --> n

ADDRESS BOOK MAIN MENU
[1] Add Entry
[2] Delete Entry
[3] View Entry
[4] Update Entry
[5] Exit
    Enter an option --> 3


        VIEW ENTRY
Enter name to view --> Elline Joy Francisco
Name    : Elline Joy Francisco
Address : 30 Blk 17 Brgy. Bungad QC
Email   : ellinejoyfrancisco@gmail.com
Tel No. : 9558253


ADDRESS BOOK MAIN MENU
[1] Add Entry
[2] Delete Entry
[3] View Entry
[4] Update Entry
[5] Exit
    Enter an option -->
```

V. QUESTION AND ANSWER:

1. Differentiate setters and getters.

   Setter is a method that is used to change the data. While Getter is a method that is used to access or retrieve data.

2. What is the importance of constructor?

   Constructor is a method that is automatically executed when an object is created. This method is used to initialize the attributes.

VI. CONCLUSION:

I conclude that main is so sensitive. And every declaring classes must use in the codes.