**Client Class**

import java.util.Scanner;

/\*\*

\*

\* @author Richelin.metellus

\* @version 01202017

\* The client class test the superclass Contact as well as the

\* subclasses Friend and Business. The getter method of each class

\* were explicitly tested. The setter r tested in a loop.

\*/

public class Client {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

System.out.println(" Entering main");

Scanner scan = new Scanner( System.in);

String userChoice;

System.out.print("How many conatacts would you like to create? ");

int num = scan.nextInt();

scan.nextLine();

Contact[] addressBook = new Contact[num];

for ( int i = 0; i < num ; )

{

String name, address, phone, title, bizName, birthday, favoriteMovie;

System.out.print("Enter B or b for business contact or F/f for friends: ");

userChoice = scan.nextLine();

char charChoice =userChoice.toLowerCase().charAt(0);

if(charChoice != 'b' && charChoice != 'f')

{

System.out.println(" Illegal Input. type B/b or F/f" );

continue;

}

// if input is legal execute statements below

System.out.print("Enter name: ");

name = scan.nextLine();

System.out.print("Enter address: ");

address = scan.nextLine();

System.out.print("Enter phone: ");

phone = scan.nextLine();

if (charChoice == 'b')

{

System.out.print("Enter title: ");

title = scan.nextLine();

System.out.print("Enter buisiness name: ");

bizName = scan.nextLine();

addressBook[i] = new Business(name, address, phone, title, bizName);

System.out.println("Creating business Contact = " + addressBook[i]);

i++;

}

else if(charChoice == 'f')

{

System.out.print("Enter birthday MM/DD/YYYY: ");

birthday = scan.nextLine();

System.out.print("What is your favorite movie? ");

favoriteMovie = scan.nextLine();

addressBook[i] = new Friend(name, address, phone, birthday, favoriteMovie);

System.out.println("Creating friend Contact = " + addressBook[i]); //invoke the toString of Friend.

i++;

}

}

for (int i = 1; i <= Contact.getContactCount(); i++)

{

System.out.println( "[" + i + "]" + addressBook[i-1]);

System.out.println(addressBook[i-1].getName());

// testing setter

addressBook[0].setName("Richelin Metellus");

addressBook[0].setAddress("1805 Universtiy Dr N");

addressBook[0].setPhone("754-551-4765");

if ( addressBook[0] instanceof Friend )

{

((Friend) addressBook[0]).setBirthday("05/10/94");

((Friend) addressBook[0]).setFavoriteMovie(" The Big Fish");

}

else if (addressBook[0] instanceof Business)

{

((Business) addressBook[0]).setBusinessName(" NDSU");

((Business) addressBook[0]).setTitle("Student");

}

System.out.println( "[" + i + "]" + addressBook[i-1]);

}

System.out.println( " Exiting main");

}

}

**CONTACT CLASS**

/\*\*

\*

\* @author richelin.metellus

\*/

public class Contact {

private static int contactCount = 0;

private String name;

private String address;

private String phone;

/\*\*

\*

\* @param name

\* @param newAddress

\* @param newPhone

\*/

public Contact(String name, String newAddress, String newPhone)

{

this.name = name;

address = newAddress;

phone = newPhone ;

contactCount++;

}

public String getName()

{

return name;

}

/\*\*

\*

\* @param newName

\*/

public void setName(String newName)

{

name = newName;

}

/\*\*

\*

\* @return

\*/

public String getAddress()

{

return address;

}

/\*\*

\*

\* @param address

\*/

public void setAddress( String address)

{

this.address = address;

}

/\*\*

\*

\* @return

\*/

public static int getContactCount()

{

return contactCount;

}

/\*\*

\*

\* @return

\*/

public String getPhone() // non-static because ii want to call it on instance.

{

return phone;

}

/\*\*

\*

\* @param phone

\*/

public void setPhone(String phone)

{

this.phone = phone;

}

/\*\*

\*

\* @return

\*/

public String toString()

{

return getClass().getSuperclass().getName() + "@" + contactCount +":"

+ getName() + ":" + getAddress() + ":" + getPhone() ;

}

/\*\*

\*

\* @param o

\* @return

\*/

@Override

public boolean equals(Object o)

{

if (!( o instanceof Contact))

return false;

Contact c = (Contact)o;

return name.equalsIgnoreCase(c.name)

&& address.equalsIgnoreCase(c.address) // not o.address. it has to be the cast identifier.

&& phone.equals(c.phone);

}

}

**FRIEND CLASS.**

/\*\*

\*

\* @author Richelin.metellus

\* @version 01/19/2017

\* Friend contact with 2+data fields & getter & setter;

\*/

public class Friend extends Contact {

private String birthday;

private String favoriteMovie;

private static int friendCount;

//Overload constructor

/\*\*

\*

\* @param name

\* @param address

\* @param phone

\* @param birthday

\* @param favoriteMovie

\*/

public Friend( String name, String address, String phone, String birthday, String favoriteMovie)

{

super(name, address,phone);

this.birthday = birthday;

this.favoriteMovie = favoriteMovie;

friendCount++ ;

}

/\*\*

\*

\* @return

\*/

public String getBirthday()

{

return birthday;

}

/\*\*

\*

\* @param birthday

\*/

public void setBirthday( String birthday)

{

this.birthday = birthday;

}

/\*\*

\*

\* @return

\*/

public String getFavoriteMovie()

{

return favoriteMovie;

}

/\*\*

\*

\* @param favoriteMovie

\*/

public void setFavoriteMovie( String favoriteMovie)

{

this.favoriteMovie = favoriteMovie;

}

/\*\*

\*

\* @return

\*/

@Override

public String toString()

{

return super.toString() + ":" +getClass().getName() + "@" + friendCount

+ " : " + getBirthday() + ":" + getFavoriteMovie();

}

/\*\*

\*

\* @param o

\* @return

\*/

@Override

public boolean equals( Object o)

{

if( !( o instanceof Object))

return false;

Friend f = (Friend) o;

return super.equals(f) && birthday.equalsIgnoreCase(f.birthday)

&& favoriteMovie.equalsIgnoreCase(f.favoriteMovie);

}

}

**Business Class**

/\*\*

\*

\* @author richelin.metellus

\* @version 01/19/2017

\* Business contact with 2extra data fields/getter/setter

\*/

public class Business extends Contact {

private static int businessCount = 0;

private String title;

private String businessName;

/\*\*

\*

\* @param name

\* @param address

\* @param phone

\* @param title

\* @param businessName

\*/

public Business( String name, String address, String phone,

String title, String businessName)

{

super(name, address, phone);

this.title = title;

this.businessName = businessName;

businessCount++;

}

/\*\*

\*

\* @return

\*/

public String getBusinessName()

{

return businessName;

}

/\*\*

\*

\* @param businessName

\*/

public void setBusinessName(String businessName)

{

this.businessName = businessName;

}

/\*\*

\*

\* @return

\*/

public String getTitle()

{

return title;

}

/\*\*

\*

\* @param title

\*/

public void setTitle(String title)

{

this.title = title;

}

/\*\*

\*

\* @return

\*/

@Override

public String toString()

{

return super.toString() + ":" + getClass().getName() + "@" + businessCount + ":"

+ title + ":" + businessName;

}

/\*\*

\*

\* @param o

\* @return

\*/

@Override

public boolean equals(Object o)

{

if (! ( o instanceof Business)) // is o an instance of the class Business. return true or false

return false; // if they are not the samme data type return false.

Business b = (Business) o;

return super.equals( b) && title.equalsIgnoreCase(b.title)

&& businessName.equalsIgnoreCase(b.businessName );

}

}