CS1083 : FR03

Assignment 3

Student Name : Omar Sebri
Student Number : 3722350

```java
/**
   @Author Omar Sebri 3722350
*/
public interface Licensable{

     double calculateLicenseFee() ;
     String getDescription();
     String getLicenseID();
}
```

```java
/**
   @Author Omar Sebri 3722350
*/
public class Realtor implements Licensable{
     final String description;
     final String license;
    public Realtor(String description, String license){
        this.description=description;
        this.license=license;
    }
    public double calculateLicenseFee(){
        if (this.description.equals("manager"))
            return(75);
         else if (this.description.equals("broker"))
            return(125);
         else if (this.description.equals("salesperson"))
            return(50);
        return 0;
    }
    public String getDescription(){
        return this.description;
    }
    public String getLicenseID(){
        return this.license ;
    }
}
```

```java
public abstract class Vessel implements Licensable{
     final String description;
     final String license;
     final double length;
public Vessel(String description, String license, double length){
    this.description=description;
    this.license=license;
    this.length=length;
}
public String getDescription(){
    return this.description;
}
public String getLicenseID(){
    return this.license;
}
public double getLength(){
    return this.length;
}
public abstract double calculateLicenseFee();


}
```

```
/*
@Author Omar Sebri 3722350
*/
public class Airboat extends Vessel{
    public final double propeller_diam ;
    public Airboat(String description, String license, double length, double
propeller_diam){
        super(description,license,length);
        this.propeller_diam=propeller_diam;
    }
    public double calculateLicenseFee(){
        if(this.propeller_diam > 66)
        return((this.length*2.75)+60);
        else
        return((this.length*2.75)+46);
    }
}
```

```
/**
   @Author Omar Sebri 3722350
*/
public class Sailboat extends Vessel{
 final boolean motor;
 final int masts;
public Sailboat(String description, String license, double length, boolean
motor, int masts){
    super(description,license,length);
    this.motor=motor;
    this.masts=masts;
}
public double calculateLicenseFee(){
    if(this.motor==true){
        return((this.masts*32)+50);
    }
    else
        return (this.masts*32) ;
}
}
```

```
/**
   @Author Omar Sebri 3722350
*/
public class LicenseRegistry{
     final String province;
     static Licensable [] list ;
    public LicenseRegistry(String province, Licensable [] arr){
        list = new Licensable [arr.length];
        this.province = province ;
        for(int i=0; i<arr.length;i++){
            list[i]=arr[i];
        }
    }
    public static String search(String license){
        for(int j=0;j<list.length;j++){
            if(list[j].getLicenseID().equals(license))
            return list[j].getLicenseID() ;
        }
         return null;
    }
    public String toString(){
        String info = (this.province+"\n");
        for(int j=0; j<list.length;j++){
            info += list[j].getLicenseID() + "   " + list[j].getDescription()
+"\n"  ;
        }
        info+="\n";
        for(int j=0; j<list.length;j++){
            info += list[j].getLicenseID() + "   " +
list[j].calculateLicenseFee() +"\n"  ;
        }
        return info ;
    }
}
```

```
/*
   @Author Omar Sebri 3722350
```

```java
*/

import java.util.Scanner;
import java.io.*;
public class Driver{
    public static void main(String[] args)
        throws IOException
    {
        Scanner sc = new Scanner(new File("in2.txt"));
        String province = sc.nextLine();
        int nlines = sc.nextInt();
        sc.nextLine();
        Licensable [] registry = new Licensable [nlines];
        System.out.println(registry.length);
        /* an object will be created based ont the first letter of the license
*/
        for(int i=0; i<nlines ; i++){
             Scanner scan = new Scanner(sc.nextLine());
             scan.useDelimiter(",");
             String license = scan.next();
             if(license.charAt(0)=='R'){
                 String description = scan.next();
                 registry[i] = new Realtor(description,license);
             }
             else if(license.charAt(0)=='A'){
                 String description= scan.next();
                 String strLength = scan.next();
                 double length = Double.valueOf(strLength);
                 String strPropeller_diam = scan.next();
                 double propeller_diam = Double.valueOf(strPropeller_diam);
                 registry[i]= new
Airboat(description,license,length,propeller_diam);
             }
             else if(license.charAt(0)=='S'){
                String description= scan.next();
                double length = scan.nextDouble();
                String strMast = scan.next();
                int masts = Integer.valueOf(strMast);
                //int masts = scan.nextInt();
                Boolean motor = scan.nextBoolean();
                registry[i]= new
Sailboat(description,license,length,motor,masts);
        }
        }
        /*we will use the licensable type array to create a registry */
         LicenseRegistry myReg = new LicenseRegistry(province, registry);

         System.out.println(myReg);
         String iter = sc.nextLine();
        while(!iter.equals("end")){
            if(LicenseRegistry.search(iter)!=null){
                System.out.println(iter + " found");
            }
            else System.out.println(iter + " not found");
            iter = sc.nextLine();
        }

    }
}
```

Test Case 1:

Input(in.txt):

```
New Brunswick

7

S12345,daysailer,68,2,false

R55551,manager

A12223,swampboat,22.5,73

S98763,schooner,120,4,true

R88880,salesperson

A22277,bayou,16.25,61

R13321,broker

S98763

A22271

R55551

End
```

Output:

```
7

New Brunswick

S12345  daysailer

R55551  manager

A12223  swampboat

S98763  schooner

R88880  salesperson

A22277  bayou
```

```
R13321  broker


S12345  64.0

R55551  75.0

A12223  121.875

S98763  178.0

R88880  50.0

A22277  90.6875

R13321  125.0


S98763 found

A22271 not found

R55551 found
```

Test case explanation:

A registry has been created and a search based on the license has been conducted and of them were found in the register

Test Case 2:

Input(in1.txt):

```
Ontario

0

S98763

A22271

R55551

End
```

Output:

```
0

Ontario



S98763 not found
```

```
A22271 not found

R55551 not found
```

Test explanation:

An empty register has been created for Ontario, the searches came back negative because the register were empty

Test case 3:

Input(in2.txt):

```
New Brunswick

7

S12345,daysailer,68,2,false

R55551,manager

A12223,swampboat,22.5,73

S98763,schooner,120,4,true

R88880,salesperson

A22277,bayou,16.25,61

R13321,broker

End
```

Output:

```
7

New Brunswick

S12345  daysailer

R55551  manager

A12223  swampboat

S98763  schooner

R88880  salesperson

A22277  bayou

R13321  broker


S12345  64.0

R55551  75.0
```

```
A12223  121.875

S98763  178.0

R88880  50.0

A22277  90.6875

R13321  125.0
```

Test case explanation:

A registry has been successfully created for New Brunswick but no search was conducted because we did not ask for it