

## **Lab Exam 3**

Important instructions:

1. Lab exams are open book and open notes.
2. Please sign your name on the attendance sheet. Programs submitted without attendance will automatically get a zero.
3. Programs that do not compile will automatically get a zero.
4. During the exam, you are not allowed to use any E-mail account, instant messaging systems to send or receive messages, or surfing the web except the material of the course on Canvas.
5. Submit your files zipped under Assignments / Lab Exam 3 on Canvas:
  - NorthAmericaDealerships.java
  - Car.java
  - Dealership.java
6. (Hint) There is not much time available to complete the lab exam. Start with the Car class, then move on the dealership class. Try to break down your problem or task into smaller and easier to solve problems. And make sure you have something that compiles when you turn it in.

Tracking Car Dealerships sales performance.

A large network of car dealerships has asked that you develop a solution to keep track of the sales for multiple dealerships for a single time period.

Each dealership is described by the following pieces of data:

- Dealership Name
- Dealership ID (unique)
- total number of sold cars for single dealership
- most common model of car sold
- average cost of sold cars
- The list of all cars sold in the time period

Each dealership when created should be created with a custom constructor that creates a dealership with a specified name. The values for each dealership should be initialized in the constructor. The dealership should have five methods:

1. *sellACar* - this will create a random car and add it to the ArrayList *carsSold*
2. *getTotalSales* – this will return the total value of the sold cars.
3. *getMostCommonModel* – this determines the most common model and returns the number of the model.
4. *toStringSoldCars* – this returns a string representation of all the cars
5. *toString* – this returns a string representation of the object.

Each car can be described by the following pieces of data:

- model type (integer)
- cost – the cost should be within the range 5,000.00 to 30,000.00.

The car class should have the following methods:

1. Getter methods for each variable
2. Setter methods for each variable
3. toString method – This will return the string representation of the object.

The file Car.java contains an incomplete definition for the car class. Save it to your directory and complete the class definition as follows:

- Declare the instance data (model type and cost)
- Add the missing method headers (aka signatures)
- Add the missing method bodies

The file Dealership.java contains an incomplete definition for the dealership class. Save it to your directory and complete the class definition as follows:

- Declare the instance data (dealership name, dealership id, and average cost of cars sold)
- Add the missing method headers (aka signatures)
- Add the missing method bodies

The file NorthAmericanDealerships.java contains an driver class to test the methods implemented above. Add the code necessary to get the average totals of all the dealerships.

A sample trace of the driver class should look like so:

Dealership Totals Average \$54,067.10

```
ID: 0 Brandon Ford, Cars sold 4 Total Sales $72,568.34 Average Cost: $18,142.08 Most  
Common Model: 1  
type 1 cost: $21,461.48  
type 1 cost: $14,787.36  
type 0 cost: $8,612.44  
type 4 cost: $27,707.06
```

```
ID: 1 Tampa Honda, Cars sold 2 Total Sales $43,916.00 Average Cost: $21,958.00 Most Common  
Model: 1  
type 1 cost: $23,193.48  
type 1 cost: $20,722.52
```

```
ID: 2 Hillsboro Auto Mart, Inc., Cars sold 2 Total Sales $45,716.98 Average Cost:  
$22,858.49 Most Common Model: 1  
type 1 cost: $28,129.17  
type 1 cost: $17,587.81
```

```
import java.text.NumberFormat;
import java.util.Random;

public class Car {
    static Random gen = new Random(System.currentTimeMillis());

    //private variables

    //Constructor with a given type as a parameter

    //getters

    //setters

    //Override the toString method

} //end class
```

```

import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.Random;

public class Dealership {
    final int numModels = 5;
    int [] modelsCount = new int[numModels];
    ArrayList<Car> carsSold = new ArrayList<Car>();
    Random gen = new Random(System.currentTimeMillis());

    //private variables

    //Constructor with a name as a parameter

    /* Sell Car - This method will create a random car object and
    * add it to the ArrayList using the add method.
    */
    public void sellACar(){
        int type = gen.nextInt(numModels);//0 to 5
        carsSold.add(new Car(type));
        modelsCount[type]++;
    }

    //Get the total cost of all the cars by add the cost of each car
    //Add code here
    public double getTotalSales(){
        double total = 0.0;
        return total;
    }

    //Get the average cost of each car by getting the totalSales
    //and dividing by the size() of the arrayList carsSold
    //Add code here

    //Get the most common model
    //Add code here

```

```
//Print all the sold cars
public String toStringSoldCars(){
    StringBuilder result = new StringBuilder();
    for(int i = 0; i < this.carsSold.size(); i++){
        result.append(this.carsSold.get(i));
    }
    return result.toString();
}
```

```
//override toString method
//Add code here
```

```
}
```

```

import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.Random;

import labExam03.Car;

public class NorthAmericanDealerships {
    public static void main(String[] args) {
        int max = 7;
        int day = 0;

        Random gen = new Random(System.currentTimeMillis());
        double sellRate = .30;

        ArrayList<Dealership> dealers = new ArrayList<Dealership>();

        dealers.add(new Dealership("Brandon Ford"));
        dealers.add(new Dealership("Tampa Honda"));
        dealers.add(new Dealership("Hillsboro Auto Mart, Inc.));

        while(day < max){
            for(int i = 0; i < dealers.size(); i++){
                if(gen.nextDouble() < sellRate){
                    dealers.get(i).sellACar();
                }
            }
            day++;
        }

        double average = 0.0;

        //Add code to get the average totals of all the dealerships

        NumberFormat f = NumberFormat.getCurrencyInstance();
        System.out.println("Dealership Totals Average " + f.format(average)+"\n");

        //print each dealership
        for(int i = 0; i < dealers.size(); i++){
            System.out.println(dealers.get(i));
            System.out.println(dealers.get(i).toStringSoldCars() +"\n");
        }
    } //end main
} //end class

```