

Programming Element 2

Software Application

Programming Module Code: CP40076E

Computer Science(Foundation)

Student ID: 21359035

Module Leader: Fehmida Mohamedali

Contents Table

Table of Contents	2
Introduction.....	3
The User Guide for the Application	4
Class Diagram	19
Purpose of each class	20
Further Discussion and Future Plans for the Application	21
References	22
Appendix.....	23

Introduction

Aim or purpose of the application

In this assignment I develop a software application using programming language called Java to help a small company keep details of items such as cars stored in their business. This is by writing code which allows users to enter details of items as well the company displaying or printing details of items already stored.

What will the end user be able to do with this application?

I create the application which enables users to select options such as entering details of cars such as the registration number, make, model and mileage of the car, printing the details they have entered and quitting the application.

Describe main options implemented

The main options are a prompt which asks the end user to enter car data or details. These will be the Registration number, the make and model of the cars and their mileages. The prompt will appear again and again to allow the user to enter details or data again and again. A menu driven application allows options to be chosen by the end user such Enter, Print or Quit. So once they have entered information they can choose to print the information on screen or choose the quit option to end the application.

Summarise key Java features used

Key Java program features used are the methods, variables, constructors (to create an object) and properties. This type of programming involves the creation of objects so therefore I wrote constructors at the beginning of the code to create the objects. In fact, they create an instance of a class. An empty constructor creates objects with default values set to values such a zero and constructors with parameters create objects with values. Properties are the declared variables each with the type or return type such as integer(int), String or Char. Methods are used to perform operations, carry out tasks or just return a value.

Menu Diagram with menu options

The User Guide for the Application

How to start application

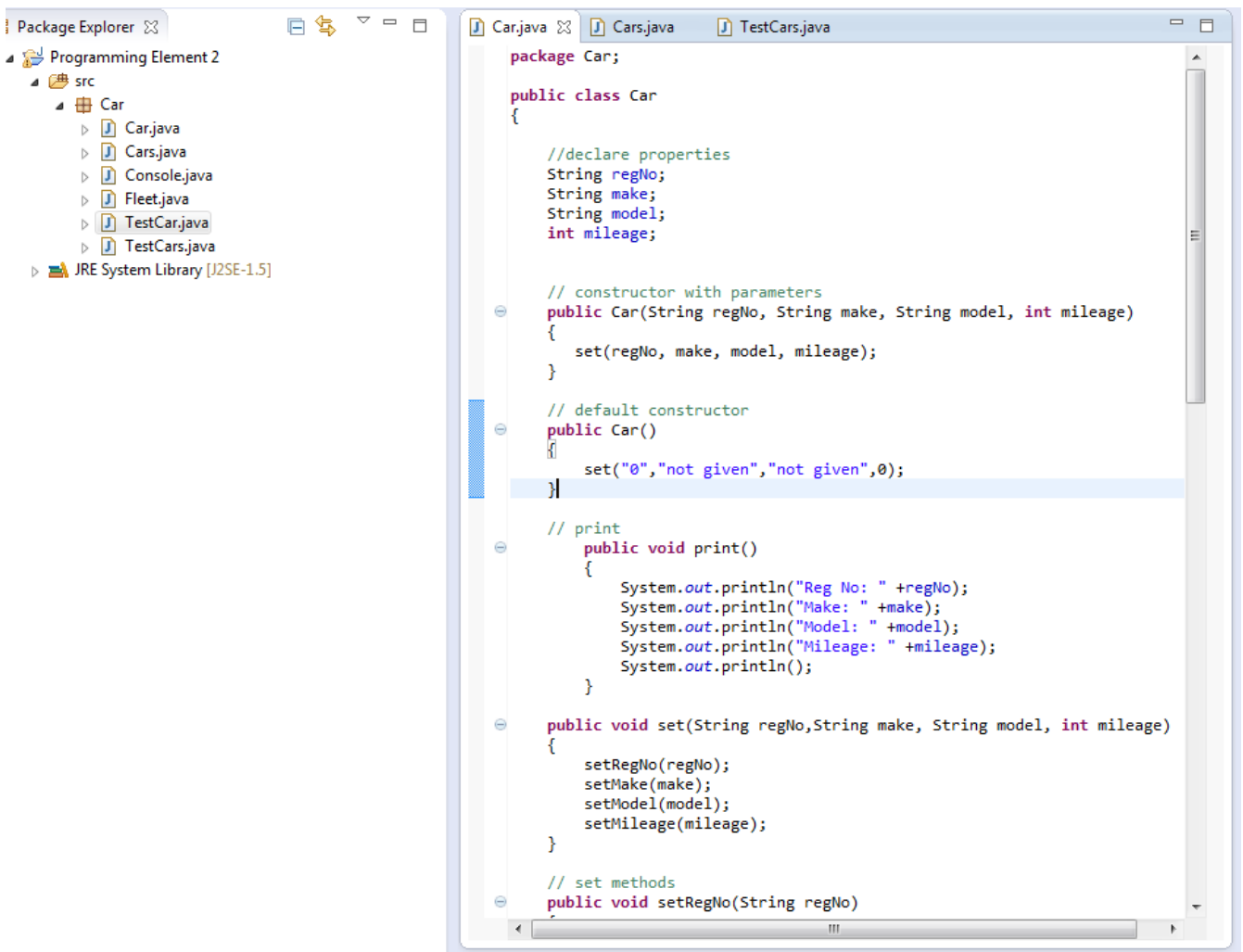
Starting the application begins by opening the PyDev Eclipse software program and going to the class named TestCars.java, selecting Run from the Run menu at the top of the screen or pressing Ctrl+F11. When the application runs a list of details of each car appears. At the bottom it asks the end user to enter car details. Do the same for all the classes except the console class Console.java.

Explanation of Menu

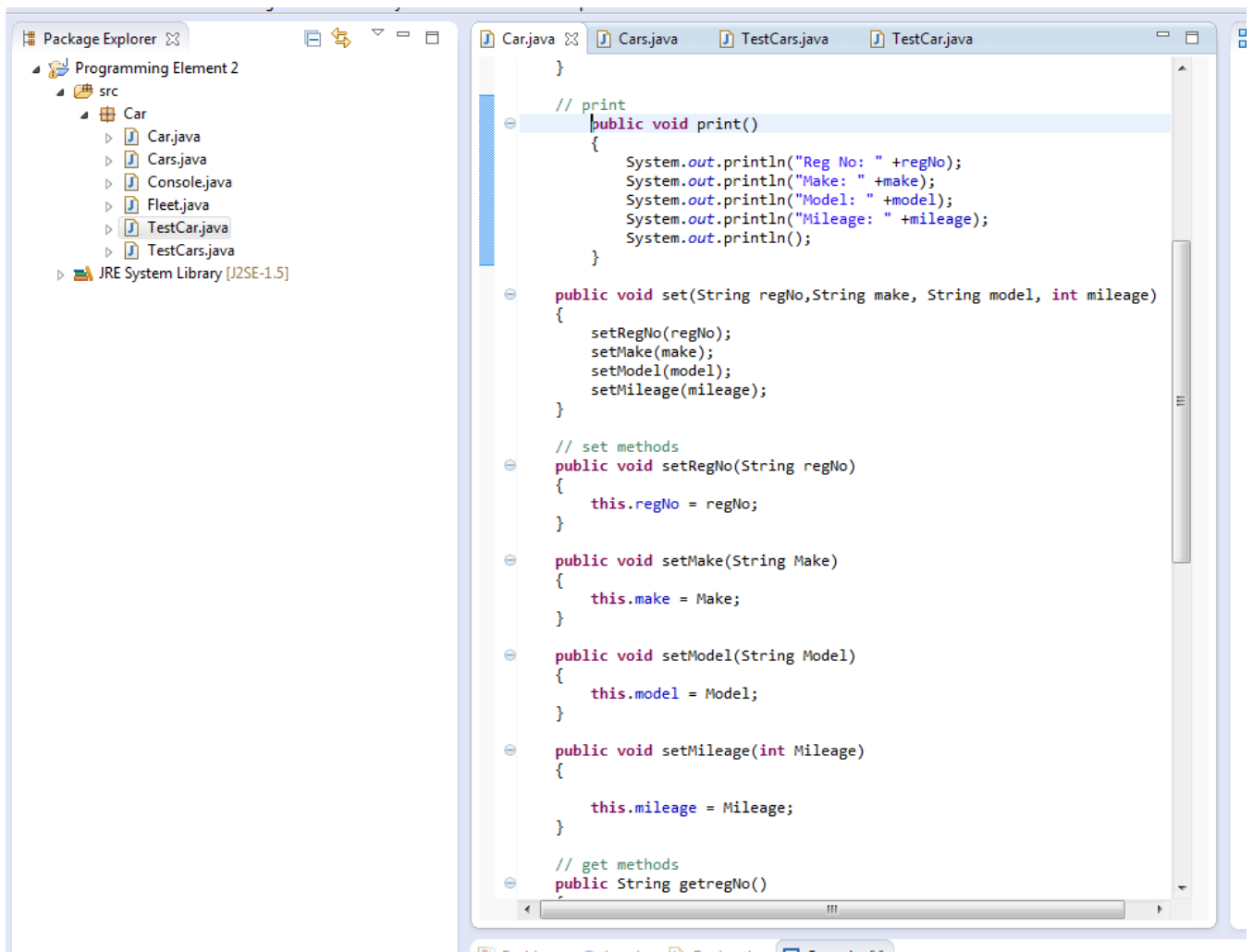
Menu

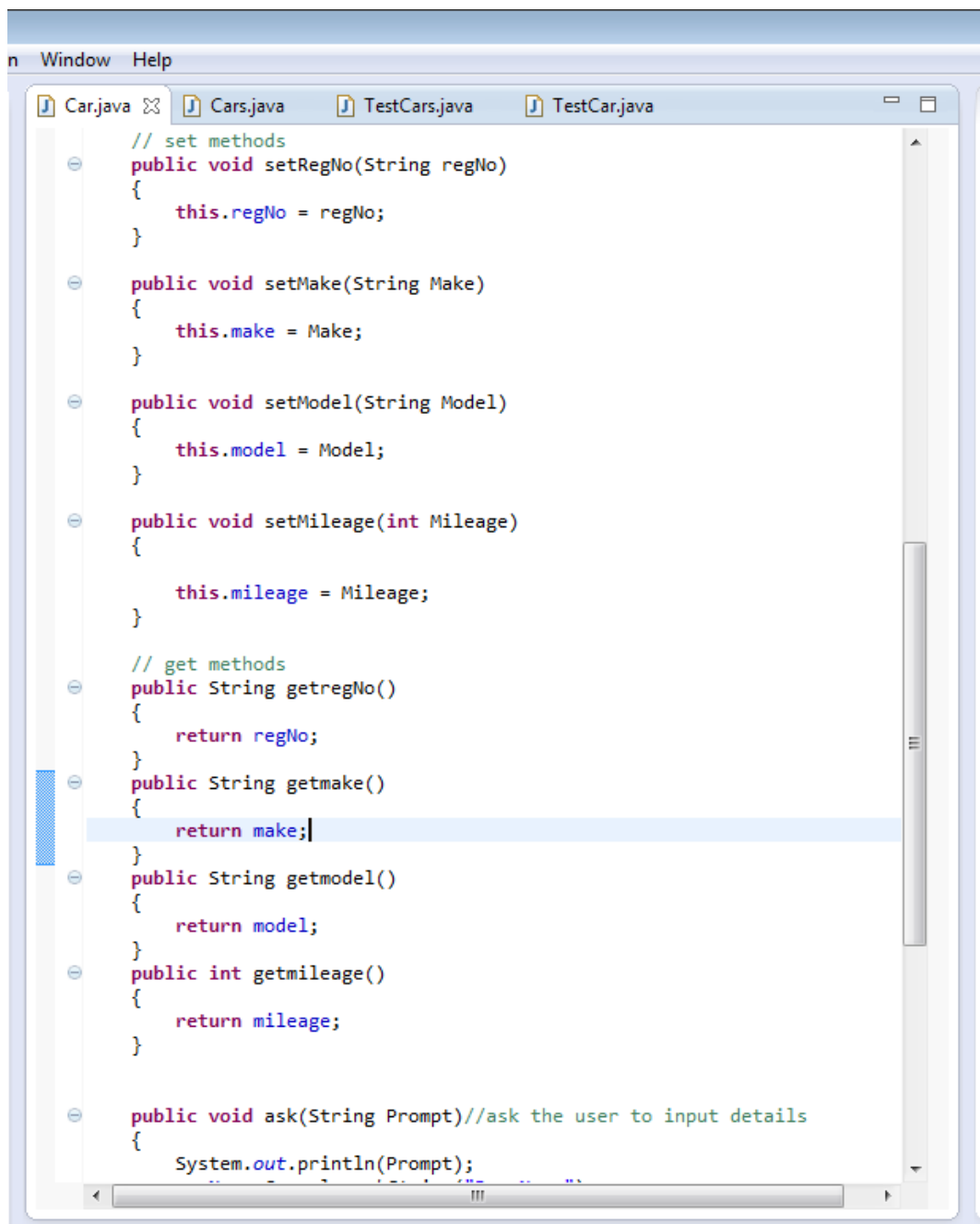
Explanation of each option

Screen shots for each option



I declared all the properties with their type here(Strings and one Integer a.k.a int). The constructors and methods codes were then created.





The screenshot shows an IDE window with four tabs: Car.java, Cars.java, TestCars.java, and TestCar.java. The Car.java tab is active, displaying the following Java code:

```
// set methods
public void setRegNo(String regNo)
{
    this.regNo = regNo;
}

public void setMake(String Make)
{
    this.make = Make;
}

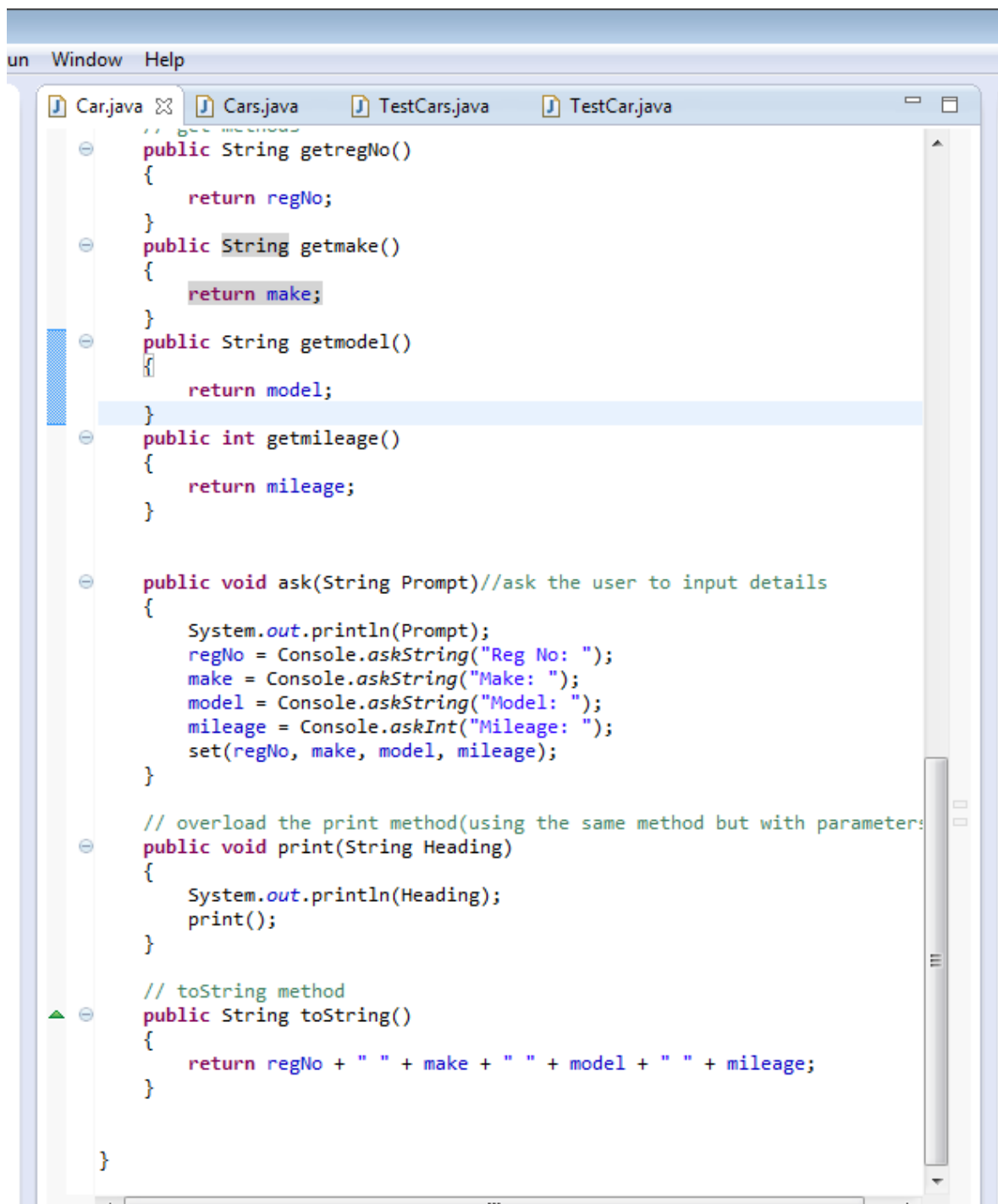
public void setModel(String Model)
{
    this.model = Model;
}

public void setMileage(int Mileage)
{
    this.mileage = Mileage;
}

// get methods
public String getregNo()
{
    return regNo;
}
public String getmake()
{
    return make;
}
public String getmodel()
{
    return model;
}
public int getmileage()
{
    return mileage;
}

public void ask(String Prompt)//ask the user to input details
{
    System.out.println(Prompt);
}
```

Here the set methods and get methods were created



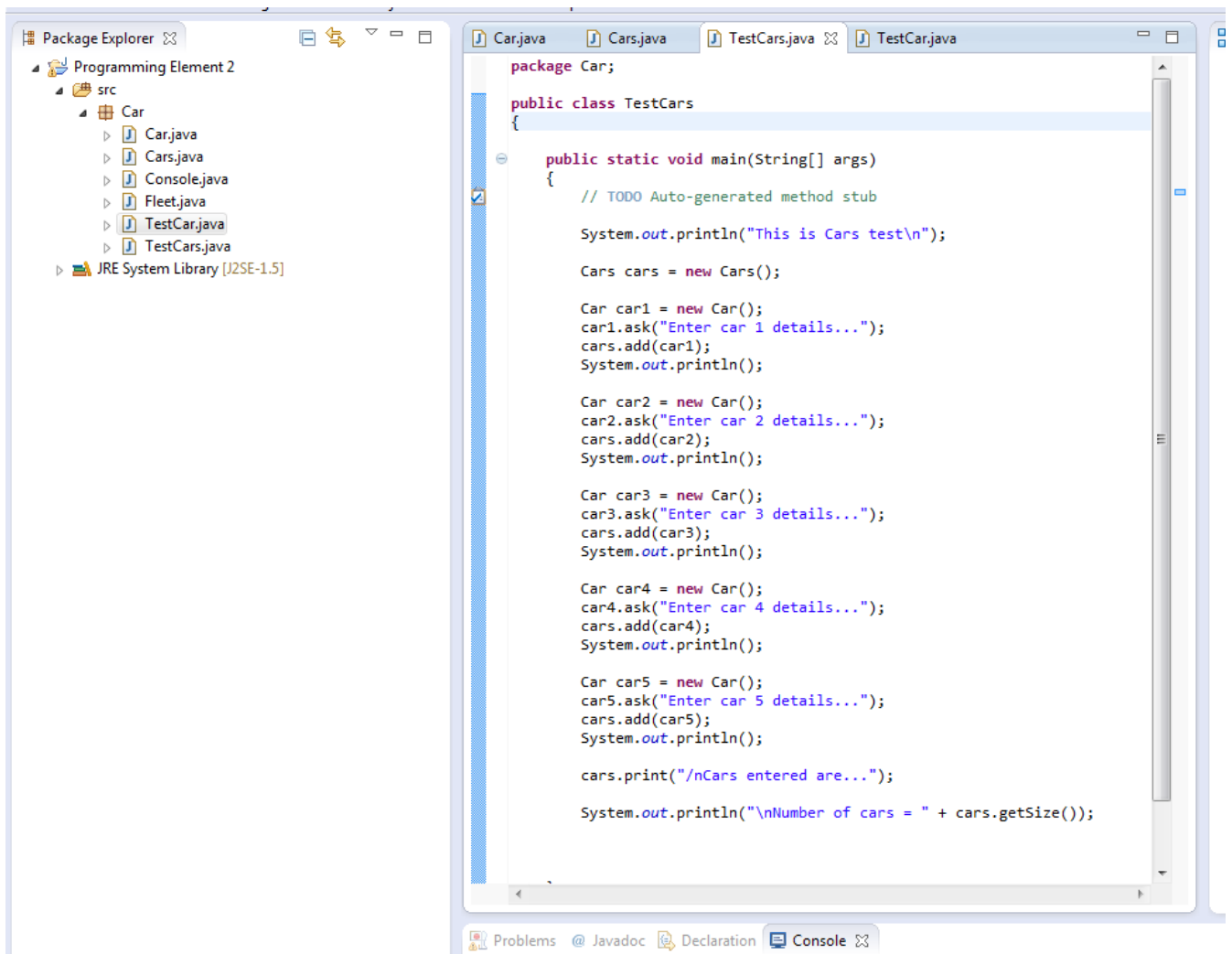
The screenshot shows an IDE window with four tabs: Car.java, Cars.java, TestCars.java, and TestCar.java. The Car.java tab is active, displaying the following Java code:

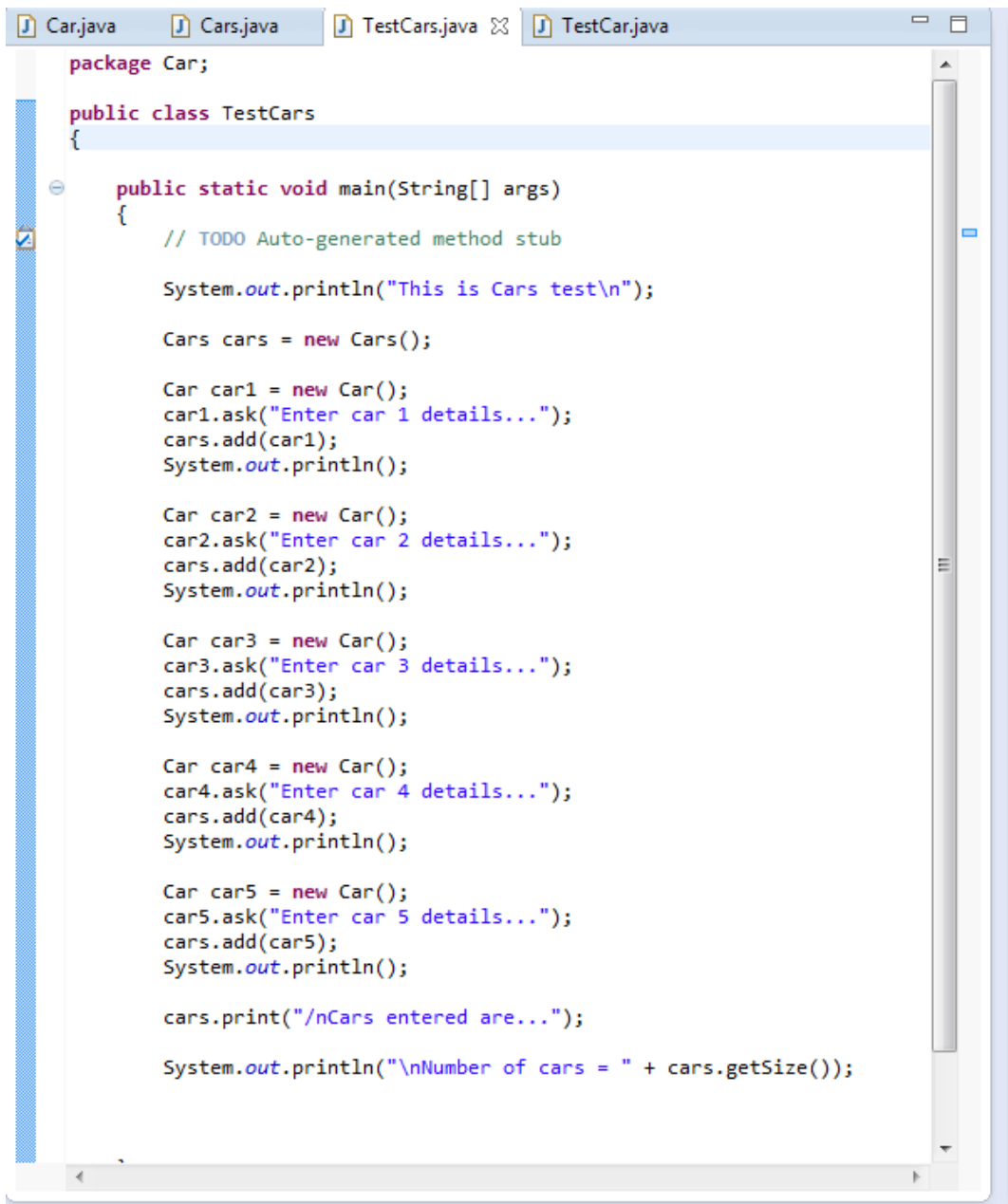
```
// get methods
public String getregNo()
{
    return regNo;
}
public String getmake()
{
    return make;
}
public String getmodel()
{
    return model;
}
public int getmileage()
{
    return mileage;
}

public void ask(String Prompt)//ask the user to input details
{
    System.out.println(Prompt);
    regNo = Console.askString("Reg No: ");
    make = Console.askString("Make: ");
    model = Console.askString("Model: ");
    mileage = Console.askInt("Mileage: ");
    set(regNo, make, model, mileage);
}

// overload the print method(using the same method but with parameters)
public void print(String Heading)
{
    System.out.println(Heading);
    print();
}

// toString method
public String toString()
{
    return regNo + " " + make + " " + model + " " + mileage;
}
}
```





```
package Car;

public class TestCars
{
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub

        System.out.println("This is Cars test\n");

        Cars cars = new Cars();

        Car car1 = new Car();
        car1.ask("Enter car 1 details...");
        cars.add(car1);
        System.out.println();

        Car car2 = new Car();
        car2.ask("Enter car 2 details...");
        cars.add(car2);
        System.out.println();

        Car car3 = new Car();
        car3.ask("Enter car 3 details...");
        cars.add(car3);
        System.out.println();

        Car car4 = new Car();
        car4.ask("Enter car 4 details...");
        cars.add(car4);
        System.out.println();

        Car car5 = new Car();
        car5.ask("Enter car 5 details...");
        cars.add(car5);
        System.out.println();

        cars.print("/nCars entered are...");

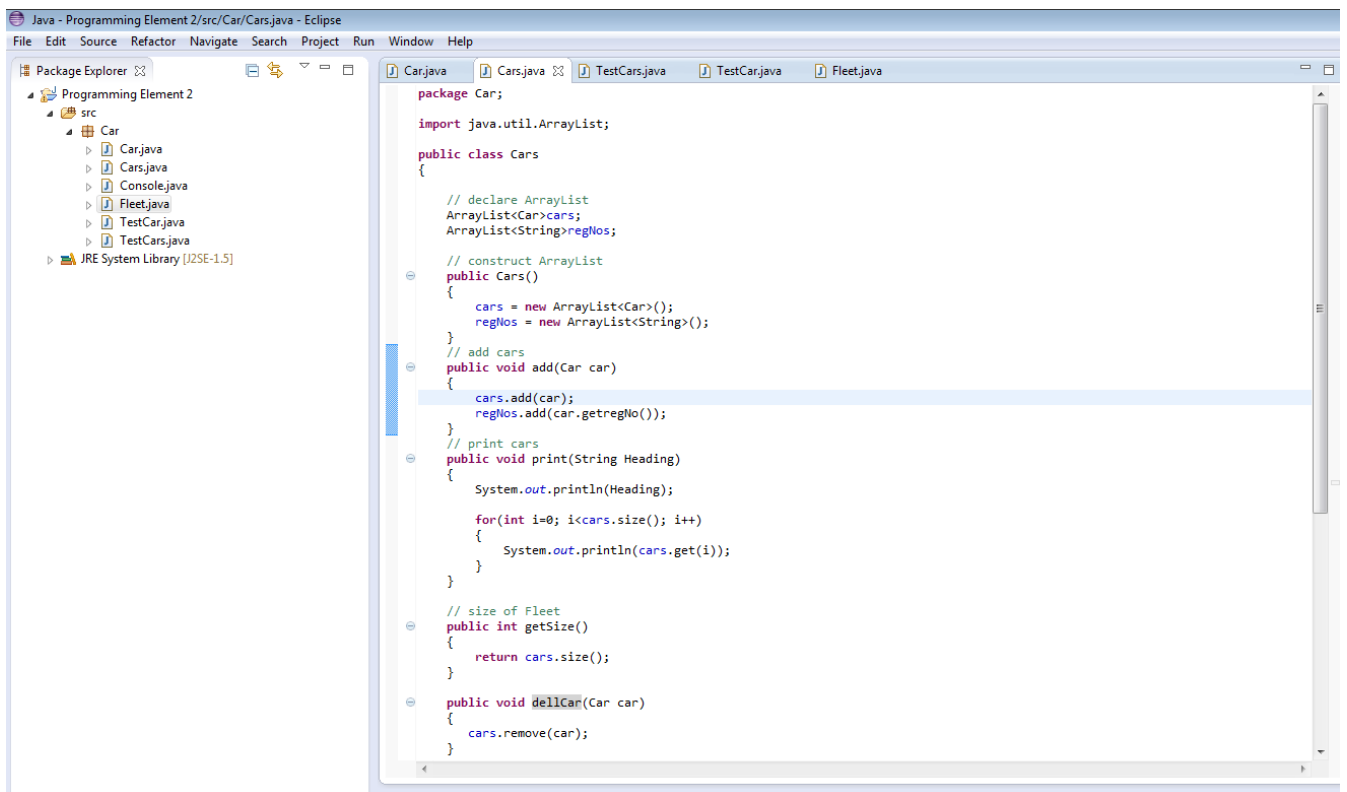
        System.out.println("\nNumber of cars = " + cars.getSize());
    }
}
```

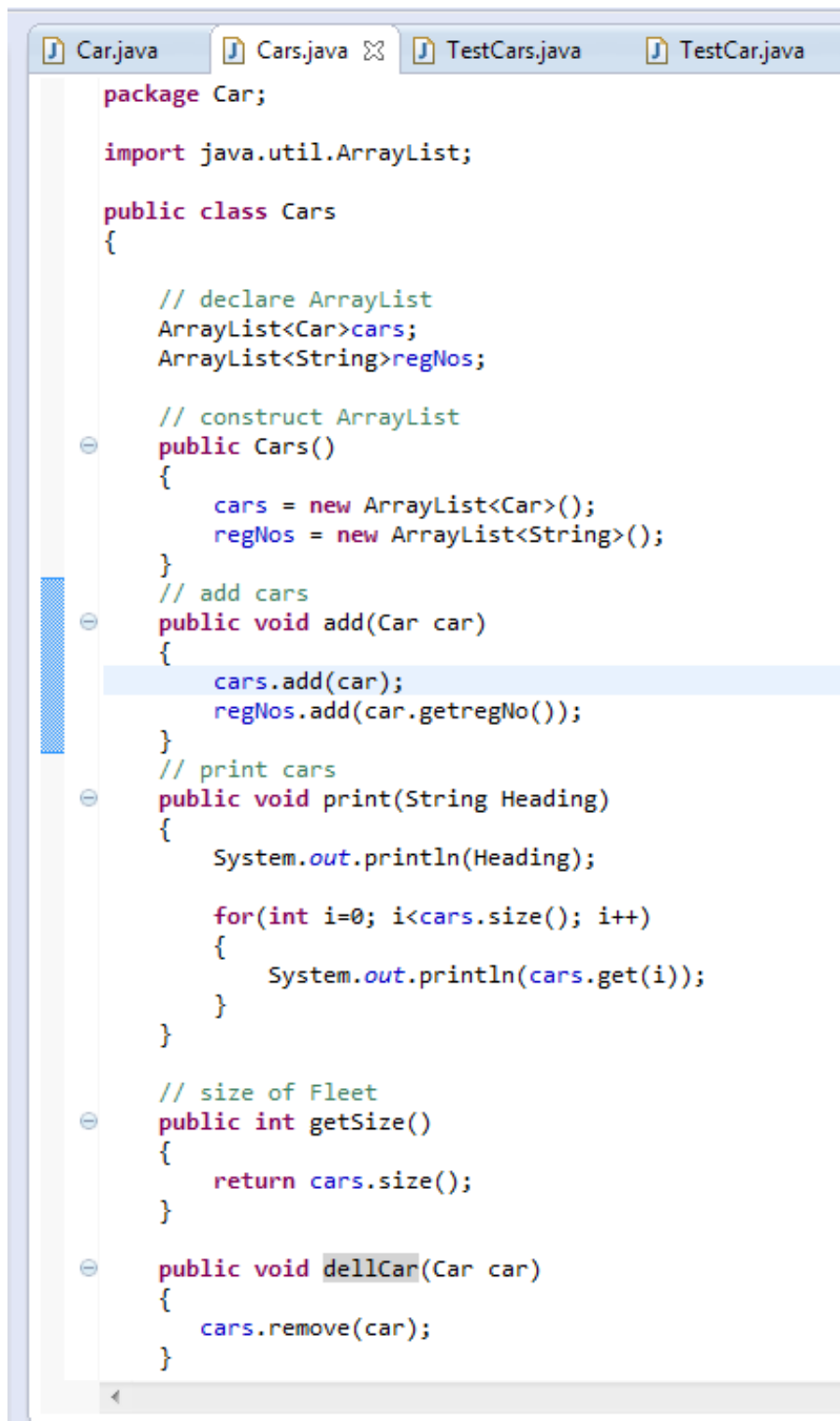


```
TestCar [Java Application] C:\Program Files (x86)\Java\jre1.8.0_91\bin\javaw.exe (14 Jan 2019 22:46:06)
Reg No: 0
Make: not given
Model: not given
Mileage: 0

Reg No: T322 WED
Make: Lancia
Model: Delta Integrale
Mileage: 64300

Reg No: EF34 UHB
```





```
package Car;

import java.util.ArrayList;

public class Cars
{
    // declare ArrayList
    ArrayList<Car>cars;
    ArrayList<String>regNos;

    // construct ArrayList
    public Cars()
    {
        cars = new ArrayList<Car>();
        regNos = new ArrayList<String>();
    }

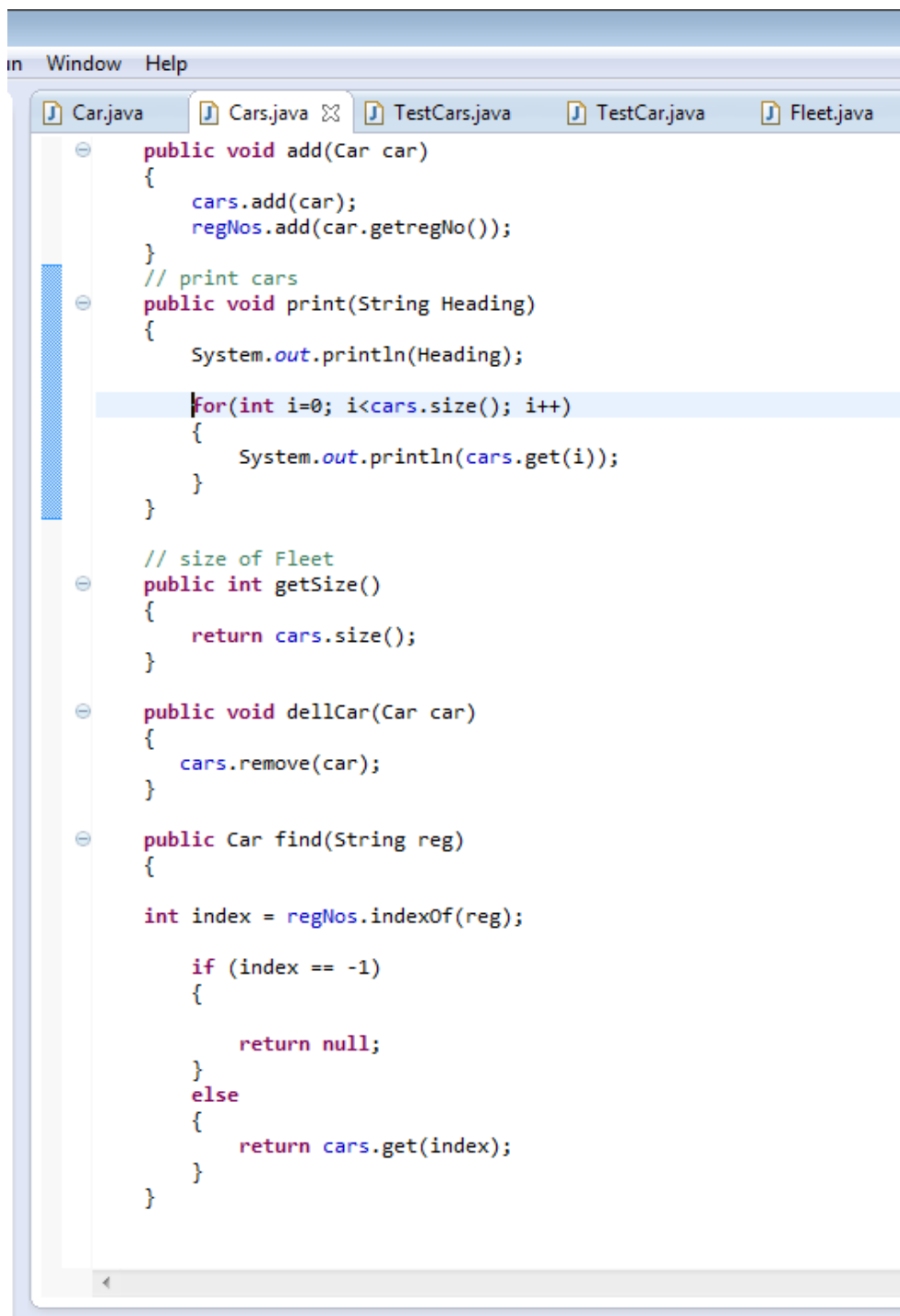
    // add cars
    public void add(Car car)
    {
        cars.add(car);
        regNos.add(car.getregNo());
    }

    // print cars
    public void print(String Heading)
    {
        System.out.println(Heading);

        for(int i=0; i<cars.size(); i++)
        {
            System.out.println(cars.get(i));
        }
    }

    // size of Fleet
    public int getSize()
    {
        return cars.size();
    }

    public void dellCar(Car car)
    {
        cars.remove(car);
    }
}
```



The screenshot shows an IDE window with a menu bar (File, Window, Help) and a tab bar containing five files: Car.java, Cars.java, TestCars.java, TestCar.java, and Fleet.java. The Cars.java file is active and displays the following Java code:

```
public void add(Car car)
{
    cars.add(car);
    regNos.add(car.getregNo());
}
// print cars
public void print(String Heading)
{
    System.out.println(Heading);

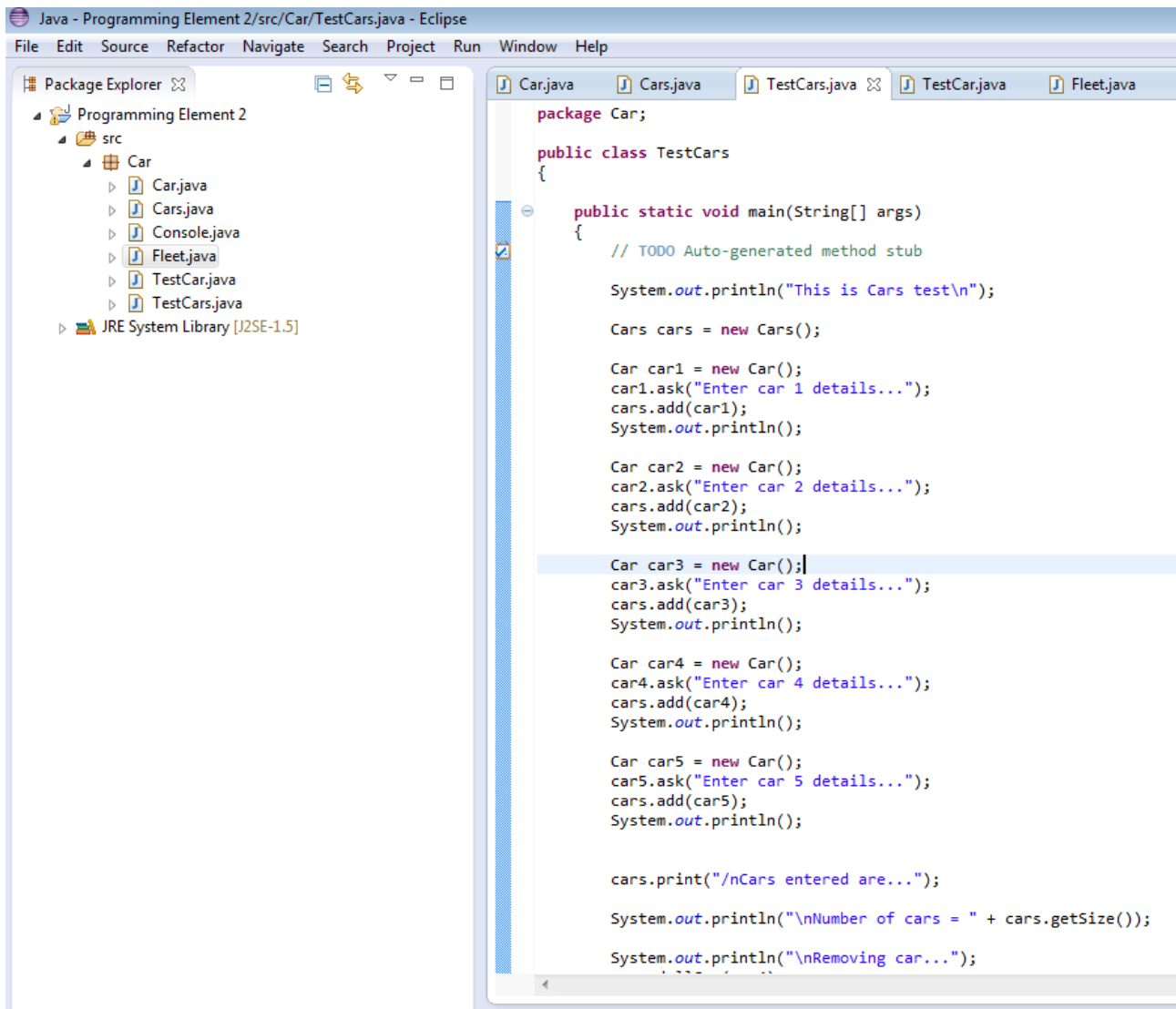
    for(int i=0; i<cars.size(); i++)
    {
        System.out.println(cars.get(i));
    }
}

// size of Fleet
public int getSize()
{
    return cars.size();
}

public void dellCar(Car car)
{
    cars.remove(car);
}

public Car find(String reg)
{
    int index = regNos.indexOf(reg);

    if (index == -1)
    {
        return null;
    }
    else
    {
        return cars.get(index);
    }
}
```



```

package Car;

public class TestCars
{
    public static void main(String[] args)
    {
        // TODO Auto-generated method stub

        System.out.println("This is Cars test\n");

        Cars cars = new Cars();

        Car car1 = new Car();
        car1.ask("Enter car 1 details...");
        cars.add(car1);
        System.out.println();

        Car car2 = new Car();
        car2.ask("Enter car 2 details...");
        cars.add(car2);
        System.out.println();

        Car car3 = new Car();
        car3.ask("Enter car 3 details...");
        cars.add(car3);
        System.out.println();

        Car car4 = new Car();
        car4.ask("Enter car 4 details...");
        cars.add(car4);
        System.out.println();

        Car car5 = new Car();
        car5.ask("Enter car 5 details...");
        cars.add(car5);
        System.out.println();

        cars.print("/nCars entered are...");

        System.out.println("\nNumber of cars = " + cars.getSize());

        System.out.println("\nRemoving car...");
    }
}

```

```

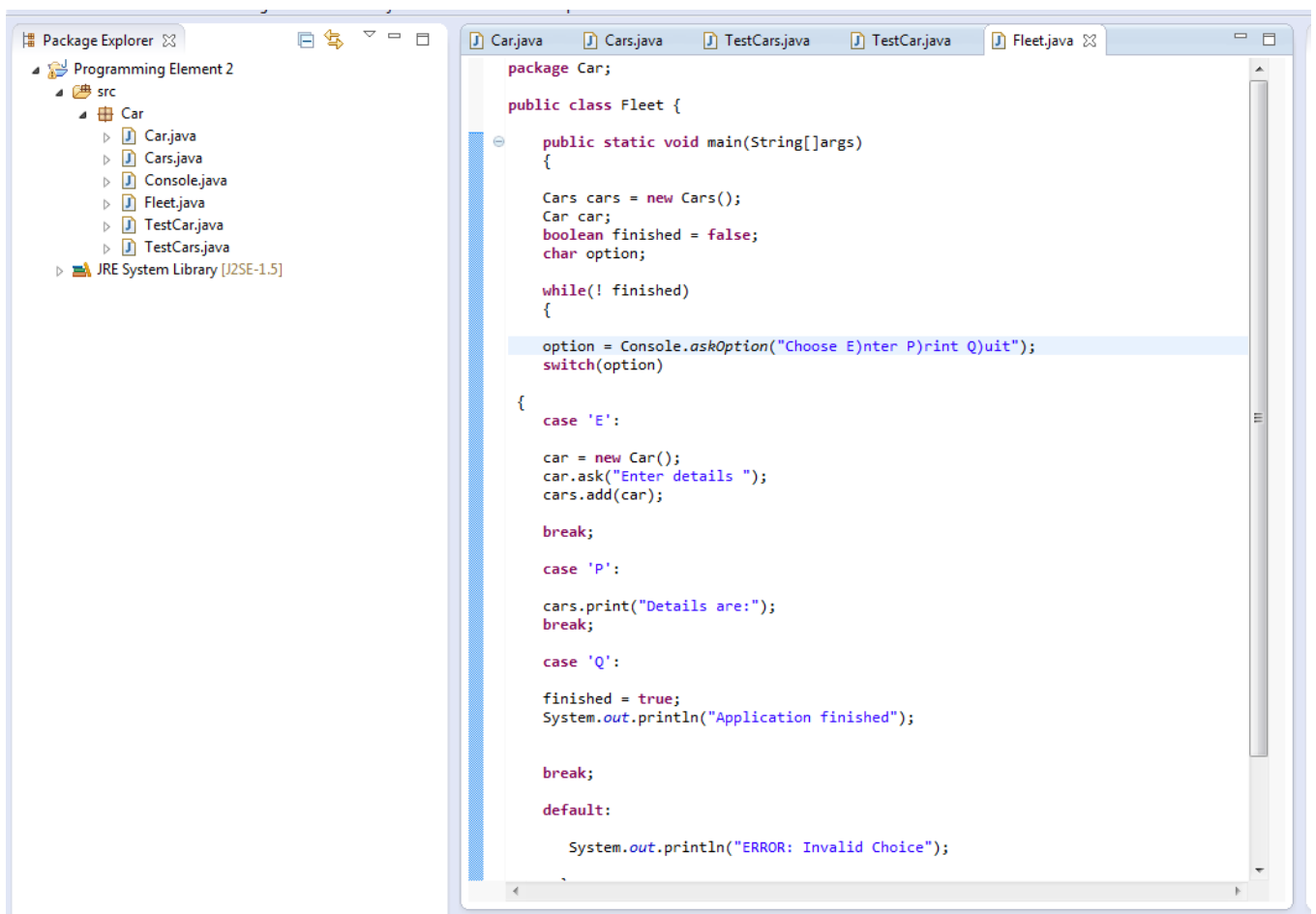
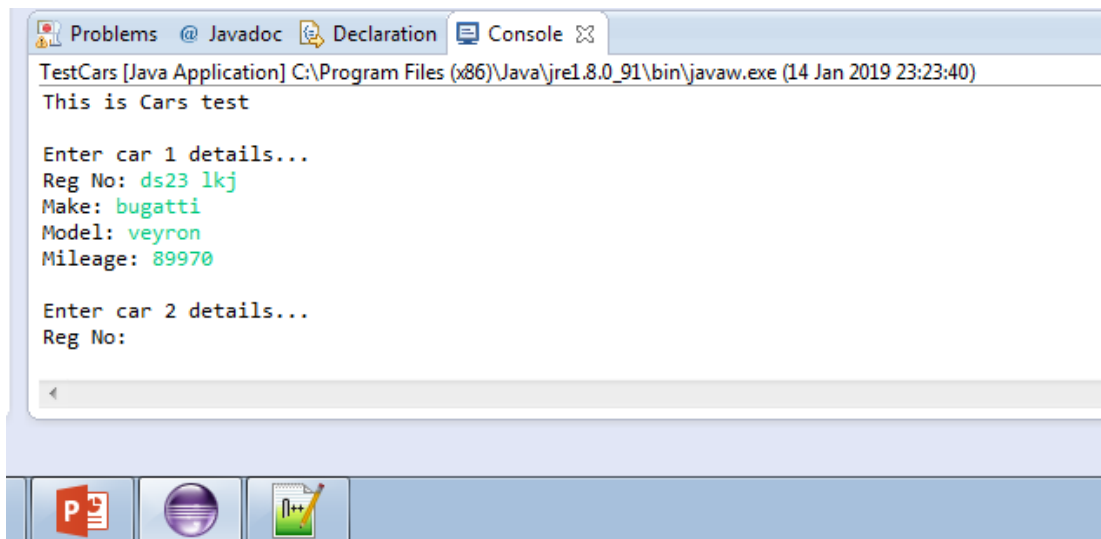
        car5.ask("Enter car 5 details...");
        cars.add(car5);
        System.out.println();

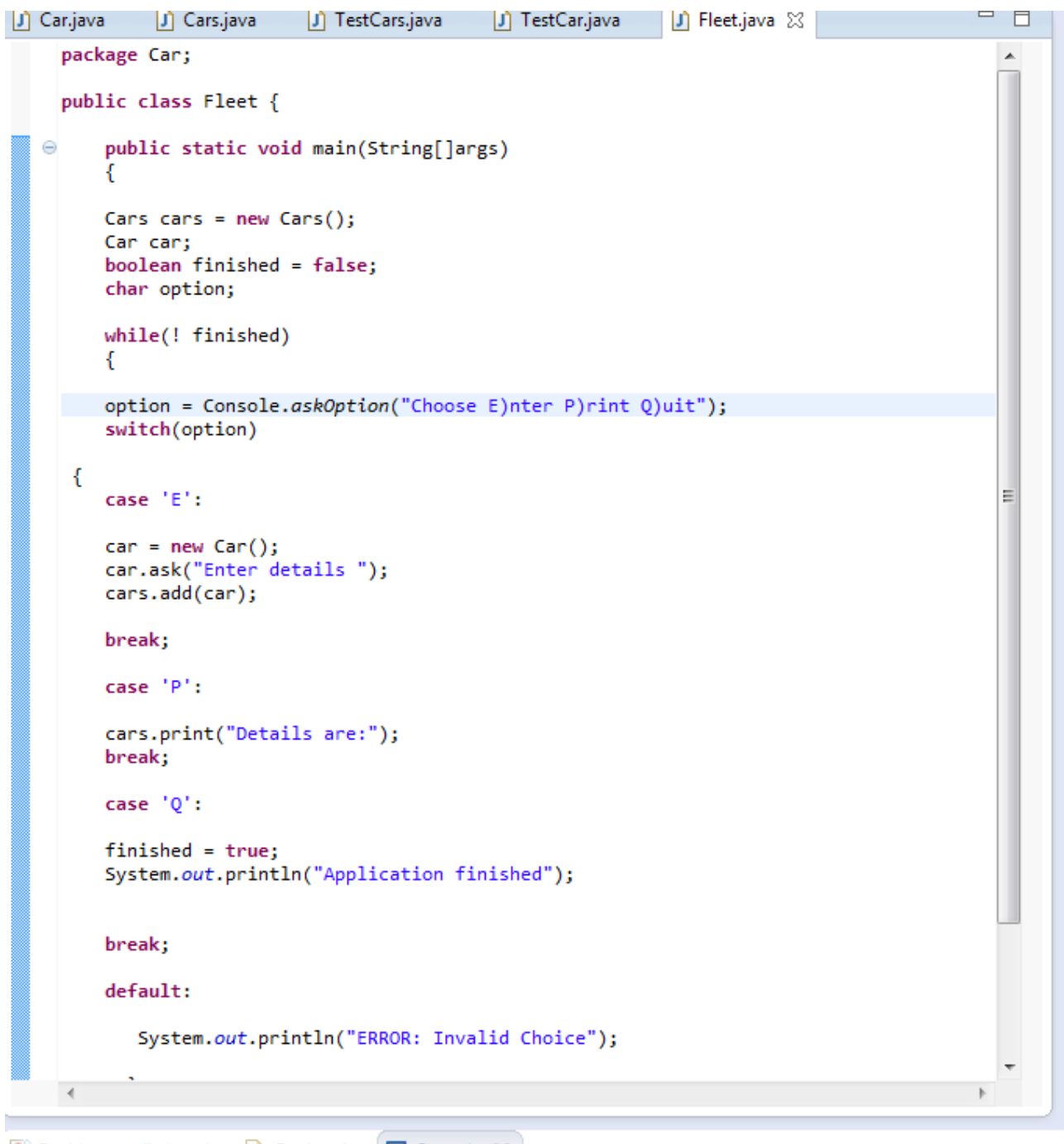
        cars.print("/nCars entered are...");

        System.out.println("\nNumber of cars = " + cars.getSize());

        System.out.println("\nRemoving car...");
        cars.dellCar(car4);
        System.out.println("\nWith one car removed number of cars is now " + cars.getSize());
    }
}

```





```
package Car;

public class Fleet {

    public static void main(String[] args)
    {

        Cars cars = new Cars();
        Car car;
        boolean finished = false;
        char option;

        while(! finished)
        {

            option = Console.askOption("Choose E)nter P)rint Q)uit");
            switch(option)

            {

                case 'E':

                    car = new Car();
                    car.ask("Enter details ");
                    cars.add(car);

                    break;

                case 'P':

                    cars.print("Details are:");
                    break;

                case 'Q':

                    finished = true;
                    System.out.println("Application finished");

                    break;

                default:

                    System.out.println("ERROR: Invalid Choice");

            }

        }

    }

}
```

Code here for each option the user can choose.


```

        car = new Car();
        car.ask("Enter details ");
        cars.add(car);

        break;

        case 'P':

            cars.print("Details are:");
            break;

        case 'Q':

            finished = true;
            System.out.println("Application finished");

            break;

        default:

            System.out.println("ERROR: Invalid Choice");

    }
}
}

```

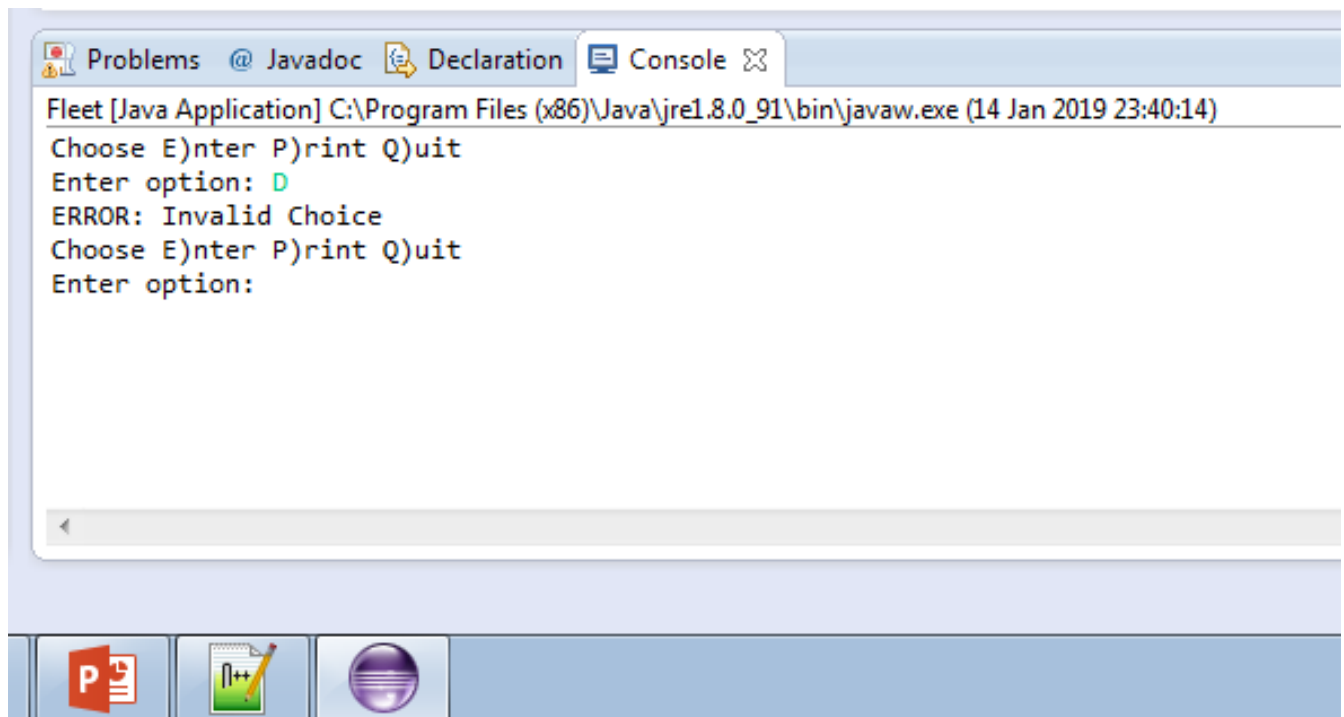
Problems @ Javadoc Declaration Console

<terminated> Fleet [Java Application] C:\Program Files (x86)\Java\jre1.8.0_91\bin\javaw.exe (14 Jan 2019 23:38:30)

```

Choose E)nter P)rint Q)uit
Enter option: E
Enter details
Reg No: LK45 BNH
Make: Vauxhall
Model: Insignia
Mileage: 34569
Choose E)nter P)rint Q)uit
Enter option: P
Details are:
LK45 BNH Vauxhall Insignia 34569

```

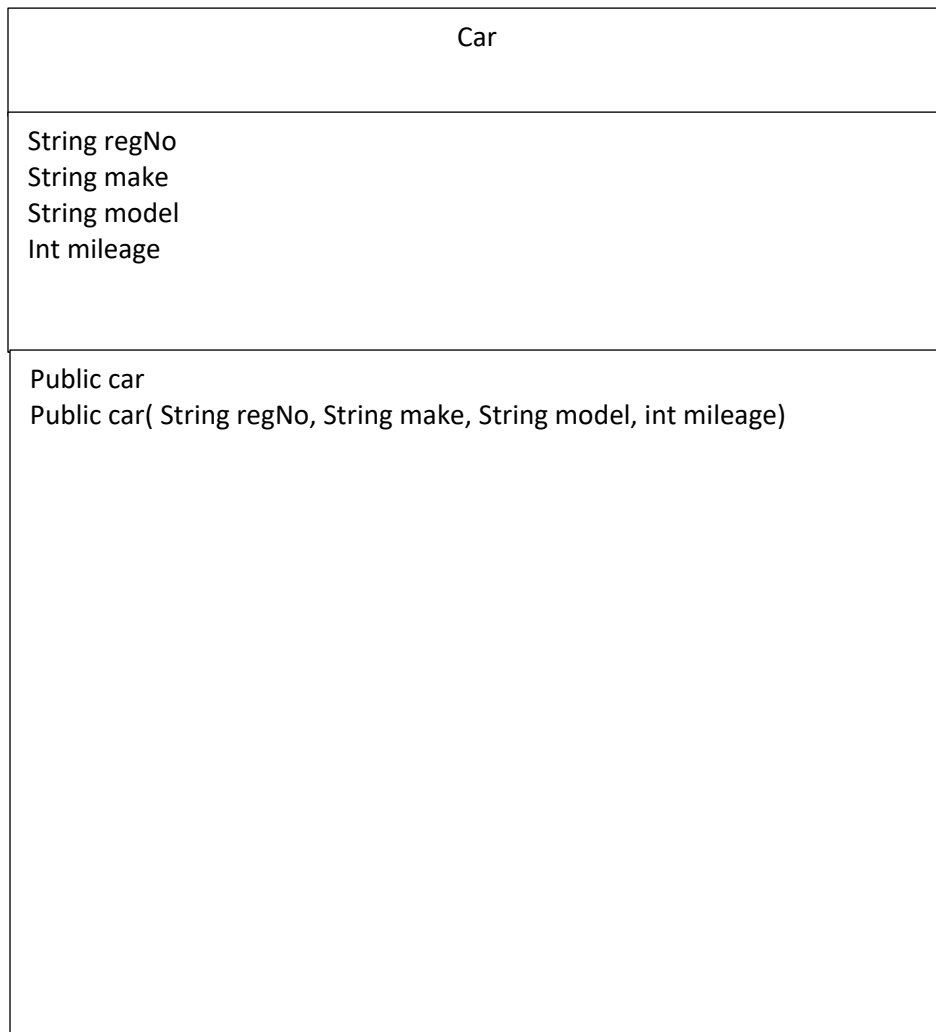


Screen shots of expected output for each option

Class Diagram

Diagram for each class to show Class name, Data properties and Method signatures

Example:



Purpose of each class

Explanation of Java features used

Clear screen shot of code with comments and/or annotations

Further Discussion and Future Plans for the Application

References

Appendix

Full copy of code listings for all classes.