Luis Gutierrez

Section: 01

Project: 1

Date: 09/06/2018

Car.java

```java
/**
 * Class representing a car and it's properties
 *
 * @author (Luis Gutierrez)
 * @version (08/30/2018)
 */
public class Car
{
    private String vin, make, model;
    private double cost;
    private int year;

    /**
     * Constructor for objects of class Car
     */
    public Car(
        String vin,
        String make,
        String model,
        double cost,
        int year)
    {
        this.vin = vin;
        this.make = make;
        this.model = model;
        this.cost = cost;
        this.year = year;
    }

    /**
     * Constructor for objects of class car
     *
     * Assumes that array items are in the correct order
     * @param tokens - A String array containing strings that correspond to car properties
     */
```

```
public Car(String[] tokens){
    this.vin = tokens[0];
    this.make = tokens[1];
    this.model = tokens[2];
    this.cost = Double.parseDouble(tokens[3]);
    this.year = Integer.parseInt(tokens[4]);
}

/**
 * Returns the car's cost
 *
 * @return this.cost
 */
public double getCost()
{
    return this.cost;
}

/**
 * Returns this car's make
 *
 * @return this.make
 */
public String getMake(){
    return this.make;
}

/**
 * Returns true if cost of car is greater than $30,000
 */
public boolean isExpensive(){
    return this.cost > (double) 30000;
}

/**
 * Returns true if car's model year is before 1968
 */
public boolean isAntique(){
    return this.year < 1968;
}

/**
 * Returns a string representation of this car
 *
 * @return String
 */
public String toString(){
    String template = "%s\t%s\t%s\t%s\t%s";

    return String.format(template,
        this.vin,
```

```
            this.make,
            this.model,
            this.cost,
            this.year
        );

    }
}
```

CarList.java

```
/**
 * Write a description of class CarList here.
 *
 * @author (Luis Gutierrez)
 * @version (08/29/2018)
 */

import java.util.ArrayList;
import java.util.Scanner;
import java.io.*;

public class CarList
{
    private ArrayList<Car> list;

    /**
     * Constructor for objects of class CarList
     */
    public CarList(File file) throws IOException
    {
        try{
            this.list = new ArrayList<Car>();
            /*BufferedReader br = new BufferedReader(new FileReader(file));
            String st;
            while((st = br.readLine()) != null){
                String[] tokens = st.split(" ");
                Car car = new Car(tokens);
                list.add(car);
            }*/
            Scanner sc = new Scanner(file);
            while(sc.hasNextLine()){
                String[] tokens = sc.nextLine().toString().split(" ");
                Car car = new Car(tokens);
                list.add(car);
            }
```

```
        }catch(FileNotFoundException e){
            System.out.println("File location specified cannot be read or does not exist");
            System.out.println(e);
        }
    }

    /**
     * Prints the heading shared by all
     * of CarList's print functions
     */
    private void printHeading(){
        System.out.println("Vin\t\tMake\tModel\tCost\tYear");
    }

    /**
     * Prints a list of all cars
     */
    public void printList()
    {
        System.out.println("All Cars");
        this.printHeading();
        for (Car car: list){
            System.out.println(car.toString());
        }

    }

    /**
     * Prints a list of expensive cars
     */
    public void printExpensiveCars(){
        System.out.println("Most Expensive Cars");
        this.printHeading();
        for (Car car: this.list){
            if (car.isExpensive()){
                System.out.println(car.toString());
            }
        }
    }

    /**
     * Returns the cheapest car from this list
     *
     * @return car
     */
    public Car cheapestCar(){
        Car cheapestCar = this.list.get(0);
        for(Car car : this.list){
            if (car.getCost() < cheapestCar.getCost()){
                cheapestCar = car;
            }
```

```
        }
        return cheapestCar;
    }

    /**
     * Determines how many cars with a given make are in this list
     *
     * @param String make - the make of car being queried for
     * @return int
     */
    public int countCarsWithMake(String make){
        int count = 0;
        for(Car car : this.list){
            if(car.getMake().compareToIgnoreCase(make) == 0){
                count++;
            }
        }
        return count;
    }

    /**
     * Returns an ArrayList of all antique cars
     *
     * @returns ArrayList<Car>
     */
    public ArrayList<Car> antiqueCarList(){
        ArrayList<Car> antiques = new ArrayList<Car>();

        for(Car car: this.list){
            if(car.isAntique()){
                antiques.add(car);
            }
        }
        return antiques;
    }
}
```

TestCarList.java

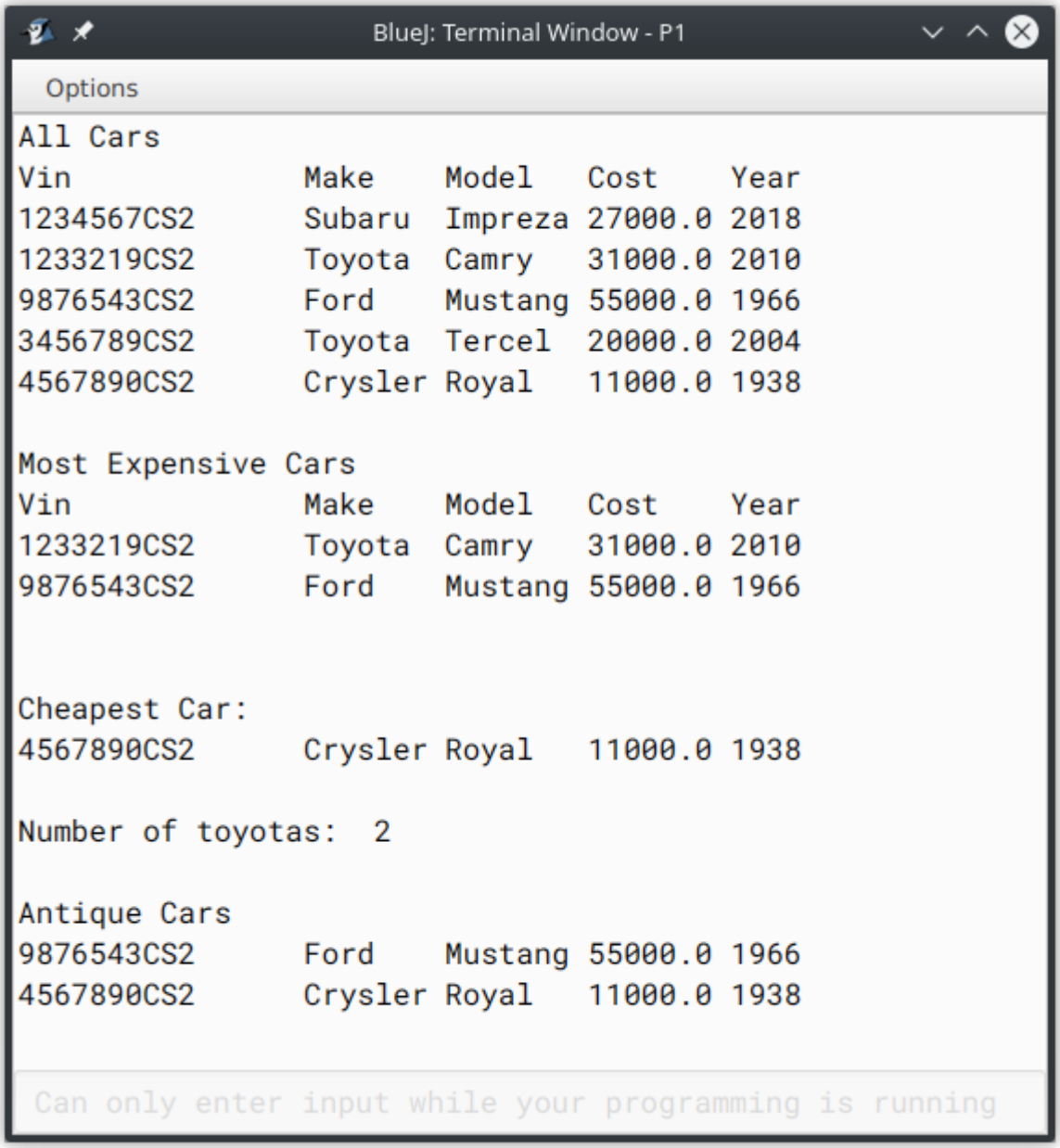
```
/**
 * Write a description of class TestCarList here.
 *
 * @author (Luis Gutierrez)
 * @version (08/29/2018)
 */
import java.io.*;
import java.util.ArrayList;
```

```
public class TestCarList
{
    public static void main(String[] args) throws IOException{
        File file = new File("inData.txt");
        CarList cars = new CarList(file);

        cars.printList();
        System.out.println();
        cars.printExpensiveCars();
        System.out.println();

        Car cheapestCar = cars.cheapestCar();
        System.out.println("\nCheapest Car:\n" + cheapestCar.toString());
        System.out.println();

        String make = "toyota";
        int makeCount = cars.countCarsWithMake(make);
        System.out.println(String.format("Number of %1ss: %2s", make, makeCount ));
        System.out.println();

        ArrayList<Car> antiques = cars.antiqueCarList();
        System.out.println("Antique Cars");
        for(Car car : antiques){
            System.out.println(car.toString());
        }
    }
}
```

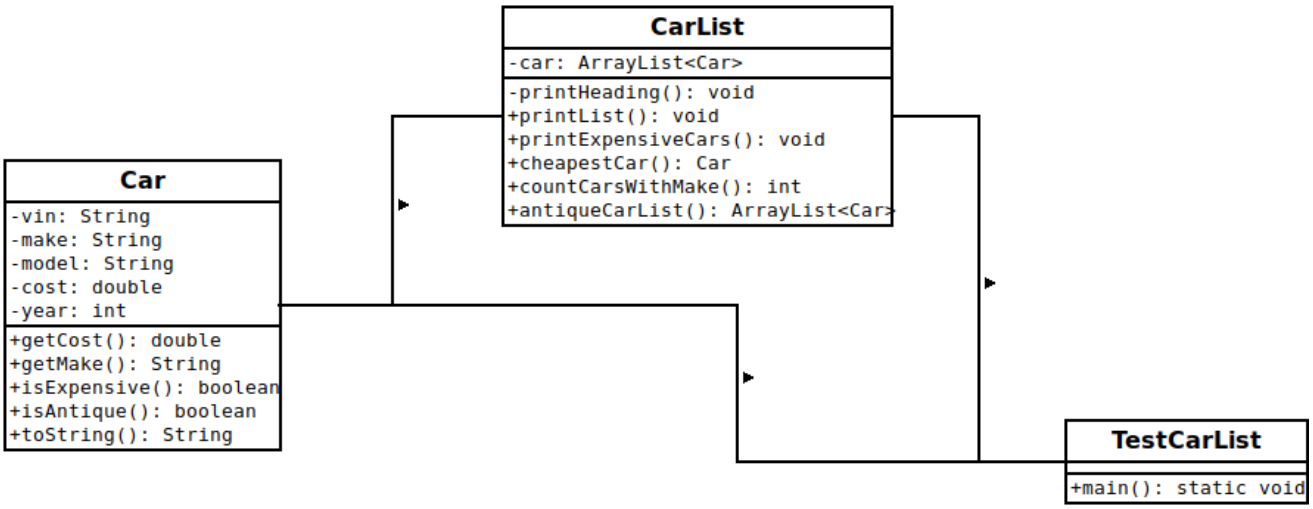
Input file

```
1234567CS2 Subaru Impreza 27000 2018
1233219CS2 Toyota Camry 31000 2010
9876543CS2 Ford Mustang 55000 1966
3456789CS2 Toyota Tercel 20000 2004
4567890CS2 Crysler Royal 11000 1938
```

Terminal screenshot


BlueJ: Terminal Window - P1
Options
All Cars
Vin Make Model Cost Year
1234567CS2 Subaru Impreza 27000.0 2018
1233219CS2 Toyota Camry 31000.0 2010
9876543CS2 Ford Mustang 55000.0 1966
3456789CS2 Toyota Tercel 20000.0 2004
4567890CS2 Crysler Royal 11000.0 1938
Most Expensive Cars
Vin Make Model Cost Year
1233219CS2 Toyota Camry 31000.0 2010
9876543CS2 Ford Mustang 55000.0 1966
Cheapest Car:
4567890CS2 Crysler Royal 11000.0 1938
Number of toyotas: 2
Antique Cars
9876543CS2 Ford Mustang 55000.0 1966
4567890CS2 Crysler Royal 11000.0 1938
Can only enter input while your programming is running

UML Diagram


CarList
-car: ArrayList<Car>
-printHeading(): void
+printList(): void
+printExpensiveCars(): void
+cheapestCar(): Car
+countCarsWithMake(): int
+antiqueCarList(): ArrayList<Car>
Car
-vin: String
-make: String
-model: String
-cost: double
-year: int
+getCost(): double
+getMake(): String
+isExpensive(): boolean
+isAntique(): boolean
+toString(): String
TestCarList
+main(): static void