**Portfolio Milestone – Mod6 – Option 1 – Automobile Class Methods**

Logan LeBoeuf

Colorado State University Global

Course Code: CSC - 320

Dr. Vanessa Cooper

March 23, 2025

**Portfolio Milestone – Dealership Vehicle Inventory Methods**

So, we have an automobile class that would be in use by a dealership to handle inventory, and for each vehicle, it requires that they have the Make, Model, Color, Year, and Mileage. Given that this is for a dealership, I will also opt to include VIN, as an easy way to identify unique vehicles, either for removal or to update variables such as Mileage. Perhaps only mileage, but a more robust program could include stats for tires and such.

It requires a main method with a try… catch, must call an automobile class with parameterized constructors, call a remove vehicle method, call an add vehicle method (which itself calls a method to list the available vehicles), call a method that updates a vehicle, and perhaps each time a method is called it is then added to a string array. Finally, a message should ask the user if they want to print this information to a file, letting the user know if that file doesn’t exist and needs to be initialized versus just updating it.

I think vehicles should be stored as a hash map of automobiles, and each one’s key should be their VIN as a string.

So I’ll want…

1. Automobile, with getters and setters and such
2. createInventory to initialize it from file if that exists, or to make a new one if not
3. listVehicles
4. addVehicle
5. removeVehicle
6. updateVehicle
7. saveInventory
8. main

Here are the methods, with Main at the beginning.

import java.util.Scanner;

import java.io.File;

import java.io.IOException;

**public class AutomobileInventory {**

    public static void main(String[] args) {

        create static hashmap called currentInventory of string, automobile, and assign it createInventory()

        addVehicle()

        removeVehicle()

        listVehicles()

        listVehicleInfo()

        updateVehicle()

        saveInventory()

    }

**public static HashMap<String, Automobile> createInventory(){**

        create empty hashmap<string, Automobile> called carInventory

        if autos.txt {

            open file

            for each line in autos.txt{

                split each line by commas

                assign each variable to separate variables for VIN, make, model, etc.

                create new Automobile object with those inputValues

                put that Automobile into carInventory using VIN as the key

                }

            close file

            print "Inventory loaded."

        }

        else {

            print "Inventory failed to load. Creating new inventory."

            createFiles()

        }

        return carInventory

    }

**public static void createFiles{**

        while (true){

            try {

                new file autosFile

                if autosFile file created successfully{

                    print "Autos.txt created."

                } else {

                    print "Autos.txt already exists!"

                }

                break

            } catch (IOException e) {

                print"An error occurred while creating files: " + e.getmessage

                new scanner

                print "enter Y to try again, and anything else to exit.""

                if user input not equals ignorecase ("Y"){

                    system exit

                    }

                close scanner

            }

        }

    }

**public static void saveInventory(){**

        //Initialize files if they don't exist but ask the user

        New scanner

        If not Autos.txt:

            print "File doe not exist. Initialize? Y/N"

            while (true){

                try:

                    If not user input equals ignorecase("Y") and not ignorecase("N"){

                        Throw IllegalArgumentException ("Invalid input, please enter Y or N.")}

                    If user’s input equals ignorecase("Y"){

                        Print "Initializing new files."

                        Break;

                    }else{

                        print "Discarding changes."

                        close scanner

                        return

                    }

                catch (IllegalArgumentException e) {

                    Print exception message

                }

            }

            close scanner

            while (true){

                try{

                    new filewriter ("autos.txt path") and overwrite

                    for each Automobile car in carInventory values(){

                        new string line = car.getVIN + "," + car.getMake + "," + //etc for each parameter of automobile}

                        write line to file + new line

                        }

                    break;

                catch (IllegalArgumentException e) {

                    print exception message

                    print "Canceling command."

                    return;

                    }

                }

            }

        }

**public static void addVehicle(){**

        new scanner

        print "Please enter the following:" + "\n""

        print "Vin: "

        string newVIN = scanner next line

        //Continue for each string parameter, the int parameters will check for validity

        print "Year: "

        while (true){

            string newYear = scanner next line

            try {

                parse newYear as int

                break;

            } catch (NumberFormatException e) {

                print "Invalid year. Please enter an integer value only."

            }

        //Continue for next int parameter in the same way. Then...

        Automobile newCar = new Automobile (newVIN, newMake, newModel, etc.)

        add newCar to currentInventory, with its VIN as its key and the vehicle as the value

        print "Vehicle successfully added: " + newCar

        close scanner

        }

    }

**public static void removeVehicle(){**

        new scanner

        print "Please enter the VIN of the vehicle you want to remove, or N to cancel.""

        string userInput = scanner next line

        try{

            if userInput equals N{

                close scanner

                return}

            try{

                Automobile car = currentInventory.get(userInput)}

                if car is null:

                    throw new IllegalArgumentException("No vehicle found with that VIN. Canceling command.")

                remove userInput from currentInventory

            } catch IllegalException e{

                print e.message

                close scanner

                return

                }

            close scanner

        }

**public static void updateVehicle()**

        new scanner

        new Automobile car

        print "Please enter the VIN of the vehicle you want to alter, or anything else to cancel."

        string userInput = scanner next line

        try{

            car = currentInventory.get(userInput)}

            if car is null:

                throw new IllegalArgumentException("No vehicle found with that VIN. Canceling command.")

        } catch IllegalException e{

            print e.message

            }

        print "Vehicle found. Please enter what field you'd like to update."

        print "1. Make"

        print "2. Model"

        //etc... for all values except VIN.

        userInput = scanner next line, but toLowerCase

        match userInput:

                case "make":

                    print "Please enter the new make."

                    userInput = scanner next line

                    car.setMake(userInput)

                case "model":

                    print "Please enter the new model."

                    userInput = scanner next line

                    car.setModel(userInput)

                //etc...

                case "year":

                    while (true){

                        print "Please enter the new year."

                        userInput = scanner next line

                        try {

                            int number = parse userInput as integer

                            car.setYear(number)

                            break

                        }   catch (NumberFormatException e) {

                                print "Invalid input, please enter a valid integer."

                        }

                    }

                //etc for mileage as well

                default:

                    print "Invalid field. Canceling command."

        close scanner

        }

**public static void listVehicleInfo(){**

        open scanner

        new Automobile car

        print "Please enter the VIN of the vehicle you want to see, or anything else to cancel.

        string userInput = scanner next line

        try{

            car = currentInventory.get(userInput)}

            if car is null:

                throw new IllegalArgumentException("No vehicle found with that VIN. Canceling command.")

                return

        } catch IllegalException e{

            print e.message

            }

        string[] carInfo = car.listInfo()

        new string compiledInfo = ""

        for each info in carInfo{

            compiledInfo += info

        }

        print compiledInfo

    }

**public class Automobile {**

    private String vin;

    private String make;

    private String model;

    private String color;

    private int year;

    private int mileage;

**public Automobile(){**

        set vin to "00000"

        set all other strings to ""

        set integers to 0

    }

    public Automobile (string vin, string make, string model, string color, int year, int mileage){

        set private variables to those passed as parameters

    }

    //Do getters for each parameter that returns that parameter

**public String getVin(){**

        return vin

    }

**public String getMake(){**

        return make

    }

    //etc...

    //Do setters for each parameter except for vin; since vin is the hashmap key, it would be better

        // to remove that vehicle and re-enter it correctly

**public String setMake(String make){**

        this.make = make;

    }

    //etc...

**public String[] listInfo(){**

        return new string array{

            "VIN: " + vin,

            " Make: " + make,

            " Model: " + model,

            //etc.

        }

    }

}