Documentation

Andrew id: yiq

Attention:

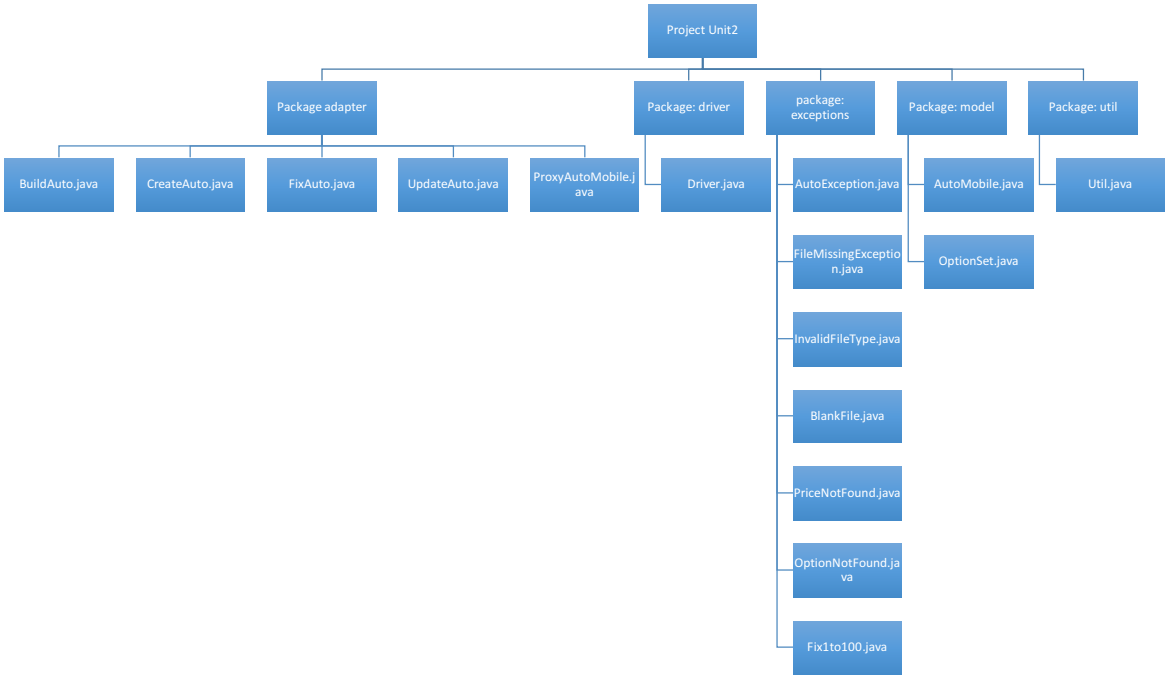
This documentation is for the project 1 unit 2 of the course 18641 in CMU. The unit is divided into two parts, part A and part B. Design details and test details on both part A and part B are included.

Part A:

1: Design details

In this part, I add exceptions package and three interfaces based on unit 1. These five designed exceptions are: FileMissingException, InvalidFileType, BlankFile, PriceNotFound and OptionNotFound. And also based on these five exceptions, I designed fix functions to make this program self-healing.

The following diagram is the package and class relationship.



2: Test details

In this part, I test five exceptions and the two update methods in interface UpdateAuto.

Here is the test process:

For exceptions test:

1) First no input filename, and this will call FileMissingException;
2) And upon screen information, I input a.java, which is not a text file. And this will call InvalidFileType exception;
3) Then upon screen information, I input Blank.txt, which is a text file in this project folder but this file is an empty file. So this will call BlankFile exception.
4) Now I input the text file: “FordZTW2.txt”, which is already put in this project folder. In this file, there is a missing price and a missing option name and I use this to test the PriceNotFound exception and OptionNotFound exception.

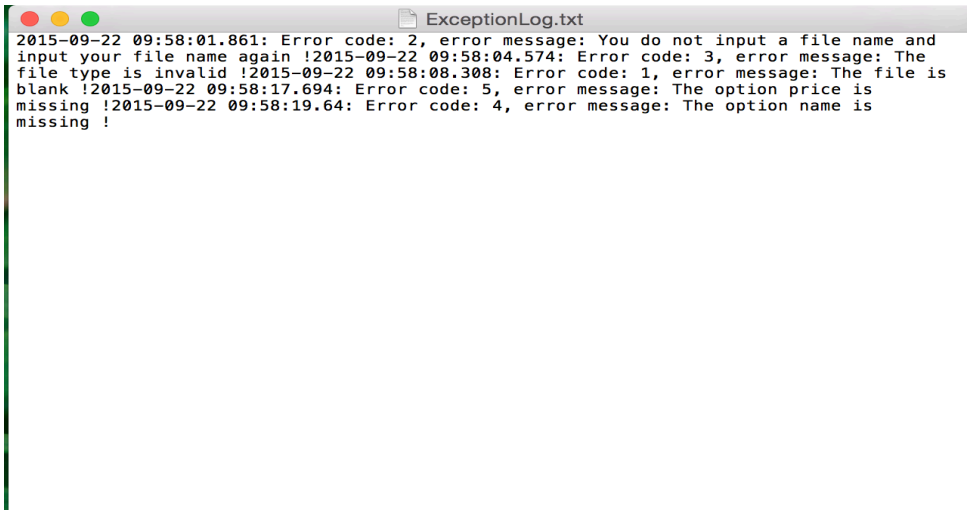
Here is the snapshot of the exceptions test:

For distinguishing from the unit 1 data, I set one of the price is 123 and an option name is Myoption, which you can clearly see from the snapshot.

```
Test begin!
------------------------------------------------------------
1: Test FileMissingException, InvalidFileType and BlankFile.
--------------------------------------
The error code is: 2
Error is: You do not input a file name and input your file name again !
Please input your new file name:
a.java
The error code is: 3
Error is: The file type is invalid !
Please input your new file name:
Blank.txt
The error code is: 1
Error is: The file is blank !
Please input your new file name:
FordZTW2.txt
The error code is: 5
Error is: The option price is missing !
You lost the price of option ABS in option set Brakers
Please input your option price:
123
The error code is: 4
Error is: The option name is missing !
You lost the option name in option set Power Moonroof
Please input your option name:
Myoption
```

```
Below is information:
The following information is model basic information:

# Car model: Focus Wagon ZTW
# Base price: $18445.00

The following information is model additional information:

#Color:
Fort Knox Gold Clearcoat Metallic: $0.00
Liquid Grey Clearcoat Metallic: $0.00
Infra-Red Clearcoat: $0.00
Grabber Green Clearcoat Metallic: $0.00
Sangria Red Clearcoat Metallic: $0.00
French Blue Clearcoat Metallic: $0.00
Twilight Blue Clearcoat Metallic: $0.00
CD Silver Clearcoat Metallic: $0.00
Pitch Black Clearcoat: $0.00
Cloud 9 White Clearcoat: $0.00

#Transmission:
automatic: $0.00
manual: $-815.00

#Brakers:
Standard: $0.00
ABS: $123.00
ABS with Advance Trac: $1625.00

#Side Impact Air Bags:
present: $350.00
not present: $0.00

#Power Moonroof:
Myoption: $595.00
not present: $0.00
```

Also exception invoke will be recorded in the text file named ExceptionLog.txt. And here is the snapshot of this file:



And this ends the exceptions test.

Now I will test the two methods defined in interface UpdateAuto.

Since my focus in this part is to test methods so I input the file: “FordZTW.txt”, which is a complete file and won’t call any exceptions.

1) First I update an option set name, which I choose here is to change **“Brakers”** to **“Control Device”**;
2) Then I update an option price, which I choose here is to change option **Present** in **Power Moonroof** from **$595** to **$790.**

Here is the snapshot of this test:

```
Test update option set name!
The following information is model basic information:

# Car model: Focus Wagon ZTW
# Base price: $18445.00

The following information is model additional information:

#Color:
Fort Knox Gold Clearcoat Metallic: $0.00
Liquid Grey Clearcoat Metallic: $0.00
Infra-Red Clearcoat: $0.00
Grabber Green Clearcoat Metallic: $0.00
Sangria Red Clearcoat Metallic: $0.00
French Blue Clearcoat Metallic: $0.00
Twilight Blue Clearcoat Metallic: $0.00
CD Silver Clearcoat Metallic: $0.00
Pitch Black Clearcoat: $0.00
Cloud 9 White Clearcoat: $0.00

#Transmission:
automatic: $0.00
manual: $-815.00

#Control Device:
Standard: $0.00
ABS: $400.00
ABS with Advance Trac: $1625.00

#Side Impact Air Bags:
present: $350.00
not present: $0.00

#Power Moonroof:
present: $595.00
not present: $0.00
```

```
Test update option price!
The following information is model basic information:

# Car model: Focus Wagon ZTW
# Base price: $18445.00

The following information is model additional information:

#Color:
Fort Knox Gold Clearcoat Metallic: $0.00
Liquid Grey Clearcoat Metallic: $0.00
Infra-Red Clearcoat: $0.00
Grabber Green Clearcoat Metallic: $0.00
Sangria Red Clearcoat Metallic: $0.00
French Blue Clearcoat Metallic: $0.00
Twilight Blue Clearcoat Metallic: $0.00
CD Silver Clearcoat Metallic: $0.00
Pitch Black Clearcoat: $0.00
Cloud 9 White Clearcoat: $0.00

#Transmission:
automatic: $0.00
manual: $-815.00

#Control Device:
Standard: $0.00
ABS: $400.00
ABS with Advance Trac: $1625.00

#Side Impact Air Bags:
present: $350.00
not present: $0.00

#Power Moonroof:
present: $790.00
not present: $0.00

-------------------------------------------
```

Also in this part, I test serialization and deserialization. Here is the snapshot of this test:

```
Serialization ZTW.ser
Serializarion is done!
-------------------------------------------
Deserialization ZTW.ser
Deserializarion is done!
After serialization, the data is kept intact.
The following information is model basic information:

# Car model: Focus Wagon ZTW
# Base price: $18445.00

The following information is model additional information:

#Color:
Fort Knox Gold Clearcoat Metallic: $0.00
Liquid Grey Clearcoat Metallic: $0.00
Infra-Red Clearcoat: $0.00
Grabber Green Clearcoat Metallic: $0.00
Sangria Red Clearcoat Metallic: $0.00
French Blue Clearcoat Metallic: $0.00
Twilight Blue Clearcoat Metallic: $0.00
CD Silver Clearcoat Metallic: $0.00
Pitch Black Clearcoat: $0.00
Cloud 9 White Clearcoat: $0.00

#Transmission:
automatic: $0.00
manual: $-815.00

#Control Device:
Standard: $0.00
ABS: $400.00
ABS with Advance Trac: $1625.00

#Side Impact Air Bags:
present: $350.00
not present: $0.00

#Power Moonroof:
present: $790.00
not present: $0.00

-------------------------------------------
```

And this ends the part A.

Part B:

1: Design details:

Based on part A, I made some modification to build part B. The most important change is that I use a LinkedHahMap to store different car models.

Also, for each model, option set and respectively option is stored as ArrayList.

In this part, each model can have its option choice for each option set. For example, for Brakers you can choose ABS or Standard. And according to your choice, total cost of the model will be calculated and given in the very last information. This is very useful since for different models, we can have different option choices and therefore have different total cost.

The packages and java file relationship is the same as the part A.

2: Test details:

In the test part, since my focus on this part B is to set option for each option set and thus calculate its total cost, I won't pay much efforts to demonstrate update methods, serialization and deserialization, which have been introduced in part A.

Here is the snapshot of the test code and the result:

```
BuildAuto auto2 = new BuildAuto();
auto2.buildAuto("./FordZTW.txt");
auto2.printAuto("Focus Wagon ZTW");
System.out.println("Test update option set name!");;
auto2.updateOptionSetName("Focus Wagon ZTW", "Brakers", "Control Device");
auto2.printAuto("Focus Wagon ZTW");
System.out.println("------------------------------------------------------------");
System.out.println("Test update option price!");;
auto2.updateOptionPrice("Focus Wagon ZTW", "Power Moonroof", "present", 790);
auto2.printAuto("Focus Wagon ZTW");
Automotbile myobj = auto2.getAuto("Focus Wagon ZTW");
myobj.setOptionChoice("Color", "Infra-Red Clearcoat");
myobj.setOptionChoice("Transmission", "manual");
myobj.setOptionChoice("Control Device", "ABS");
myobj.setOptionChoice("Side Impact Air Bags", "present");
myobj.setOptionChoice("Power Moonroof", "present");
auto2.prinOption("Focus Wagon ZTW");
```

And I will only give the option choice information:

```
-------------------------------------------
This is your model information:
Model name: Focus Wagon ZTW
Color: Infra-Red Clearcoat: 0.0
Transmission: manual: -815.0
Control Device: ABS: 400.0
Side Impact Air Bags: present: 350.0
Power Moonroof: present: 790.0
Next is your model total price:
Total price: $19170.0
```

As you can see from the figures, model option choices are properly displayed.

Test completed.

Part B completed.