Assignmet1

By Yi (Elliot) Cao
Andrew: yc2

Assignment 1: 1 point taken off
Total Points: 19 out of 20
Overall Comments:
Good job! The class diagram seem incomplete or do not match the handout requirements.

Changes made to reclaim the points:
-The UML class diagram is carefully rebuilt **included in the package: yc2_project1_unit1_updated**
Test case is listed in the page 3 and 4.
And 1 possible points could be returned for the assignment1.

How to Run My Code:
1.Import the project
2. Run test01.java

Test the coin class:

the initial face is:heads

==========

the facing up side is:heads
the facing up side is:tails
the facing up side is:heads
the facing up side is:heads
the facing up side is:heads
the facing up side is:tails
the facing up side is:tails
the facing up side is:tails
the facing up side is:heads
the facing up side is:heads
the facing up side is:heads
the facing up side is:heads
the facing up side is:tails
the facing up side is:tails
the facing up side is:heads
the facing up side is:tails
the facing up side is:tails
the facing up side is:heads
the facing up side is:tails
the facing up side is:heads
the times of heads is:11;the times of tails is:9

Test the Car :

pls input the parking time and the purchased time:
121 60
the fine is:35.0
PoliceOfficer [name=wang, badge=1111]
ParkedCar [make=US, model=small, color=green, license=212, time=121]
==========
pls input the parking time and the purchased time:
119 60
the fine is:25.0
PoliceOfficer [name=wang, badge=1111]
ParkedCar [make=US, model=small, color=green, license=212, time=119]
==========
pls input the parking time and the purchased time:
61 60
the fine is:25.0
PoliceOfficer [name=wang, badge=1111]
ParkedCar [make=US, model=small, color=green, license=212, time=61]
==========
pls input the parking time and the purchased time:

Project1_Unit1

By Yi (Elliot) Cao
Andrew: yc2

Project1_unit1:

Overall Comments: 1.5 points taken off

-1 CRUD operations not demonstrated

-0.5 properties of option class should all be private (you have not mentioned any access identifier)

Changes made to reclaim the points:

- CRUD operations have been demonstrated carefully in java classes and the package and class usage also listed in the next page
- Change the option class properties to be private and option class is protected as required in the instruction

Test case is listed in the page ~~4 and 5.~~ 8 and 9

And 1.5 possible points could be returned for the project1_unit1.

Overall understanding of unit1:

As for this unit1, I strictly follow the requirements and give adequate test. The auto object can be built successfully.Automobile class completed with access method for all CRUD operations for Option, OptionSet and Automotive class. Properties Option and OptionSet class are protected. Data is populated into Automotive instance using a text file. Serialization and Deserialization is used.

How to Run My Code:

1.Import the project

2. Run driver.java

Package and classes list:

Model:

Automobile : encapsulate all necessary operations and attributes for car

OptionSet: encapsulate all optionset and options' operation and attributes

Util:

FileIO: used to build auto object and serialization and deserialization

List details of the built auto and serialize :

car name:Focus Wagon ZTW
the OptionSet name:Color
Option [name=Fort Knox Gold Clearcoat Metallic, price=0.0]
Option [name=Liquid Grey Clearcoat Metallic, price=0.0]
Option [name=Infra-Red Clearcoat, price=0.0]
Option [name=Grabber Green Clearcoat Metallic, price=0.0]
Option [name=Sangria Red Clearcoat Metallic, price=0.0]
Option [name=French Blue Clearcoat Metallic, price=0.0]
Option [name=Twilight Blue Clearcoat Metallic, price=0.0]
Option [name=CD Silver Clearcoat Metallic, price=0.0]
Option [name=Pitch Black Clearcoat, price=0.0]
Option [name=Cloud 9 White Clearcoat, price=0.0]
the OptionSet name:Transmission
Option [name=Automatic, price=0.0]
Option [name=Standard, price=-815.0]
the OptionSet name:Brakes/Traction Control
Option [name=Standard, price=0.0]
Option [name=ABS, price=400.0]
Option [name=ABS with Advance Trac, price=1625.0]
the OptionSet name:Side Impace Air Bags
Option [name=None, price=0.0]
Option [name=If Selected, price=350.0]
the OptionSet name:Power Moonroof
Option [name=None, price=0.0]
Option [name=If Selected, price=595.0]
serilization successful!

De-serialize ：

now beagin deserilization!
car name:Focus Wagon ZTW
the OptionSet name:Color
Option [name=Fort Knox Gold Clearcoat Metallic, price=0.0]
Option [name=Liquid Grey Clearcoat Metallic, price=0.0]
Option [name=Infra-Red Clearcoat, price=0.0]
Option [name=Grabber Green Clearcoat Metallic, price=0.0]
Option [name=Sangria Red Clearcoat Metallic, price=0.0]
Option [name=French Blue Clearcoat Metallic, price=0.0]
Option [name=Twilight Blue Clearcoat Metallic, price=0.0]
Option [name=CD Silver Clearcoat Metallic, price=0.0]
Option [name=Pitch Black Clearcoat, price=0.0]
Option [name=Cloud 9 White Clearcoat, price=0.0]
the OptionSet name:Transmission
Option [name=Automatic, price=0.0]
Option [name=Standard, price=-815.0]
the OptionSet name:Brakes/Traction Control
Option [name=Standard, price=0.0]
Option [name=ABS, price=400.0]
Option [name=ABS with Advance Trac, price=1625.0]
the OptionSet name:Side Impace Air Bags
Option [name=None, price=0.0]
Option [name=If Selected, price=350.0]
the OptionSet name:Power Moonroof
Option [name=None, price=0.0]
Option [name=If Selected, price=595.0]

Project1_Unit4

By Yi (Elliot) Cao
Andrew: yc2

Overall Total - 33.5/40

-5 constants and socket interface not implemented
-1.5 total price should be displayed and calculated as per configuration

Changes made to reclaim points:
- Socket constants interface has been implemented as SoketConstans.java both in server and client side;
- Final total price as been calculated as test in page 19

Test cases have been presented from page 16-19
6.5 possible points could be added back;

Overall understanding of unit4:
As for this unit4, I strictly follow the requirements and give adequate test. The client can Exit, Upload, Get and Select. The Server side can access multiple clients' request and operated as requested. As for the test, we can open the server first and start several clients to take different different operations. We have 6 packages for client and server.

As for the server side:

Adaptor :

BuildAuto： a class implements all functions of proxyAutomobile, CreateAuto, UpdateAuto, mainly used for hiding all these function from users.

CreateAuto: an interface, used to build auto object and print auto object

FixAuto: an interface, used to fix the exceptions

proxyAutomobile: encapsulate all “CRUD” operations for automobile

UpdateAuto: an interface, used to update the OptionSet and Option

EditThreads: an interface, used to bridge the EditOptions and BuildAuto class

Exception:

AutoException: implements FixAuto used to fix exceptions:

ExceptionNum: to enumerate all exceptions

Helpers: include different fix methods for different exceptions

log: used to record the timestamp of exception and the err message of exception

Model:

Automobile： encapsulate all necessary operations and attributes for car

OptionSet: encapsulate all optionset and options’ operation and attributes

AutoList: encapsulate automobile operations and attributes

Util:

FileIO: used to build auto object and serialization and deserialization

Scale:

EditOptions: implement multithreads operations

OptinNum: to enumerate all edit options

Server:

AutoServer: the interface includes all responses for client request operations

BuildCarModelOptions: implements the AutoServer

DefaultSocketServer: access the client side requests

Server: start the server side;

SocketConstants: include the constants for socket

As for the client side:

Adaptor :

BuildAuto： a class implements all functions of proxyAutomobile, CreateAuto, UpdateAuto, mainly used for hiding all these function from users.

CreateAuto: an interface, used to build auto object and print auto object

FixAuto: an interface, used to fix the exceptions

proxyAutomobile: encapsulate all “CRUD” operations for automobile

UpdateAuto: an interface, used to update the OptionSet and Option

EditThreads: an interface, used to bridge the EditOptions and BuildAuto class

Exception:

AutoException: implements FixAuto used to fix exceptions:

ExceptionNum: to enumerate all exceptions

Helpers: include different fix methods for different exceptions

log: used to record the timestamp of exception and the err message of exception

Model:

Automobile： encapsulate all necessary operations and attributes for car

OptionSet: encapsulate all optionset and options’ operation and attributes

AutoList: encapsulate automobile operations and attributes

Util:

FileIO: used to build auto object and serialization and deserialization

Scale:

EditOptions: implement multithreads operations

OptinNum: to enumerate all edit options

Client:

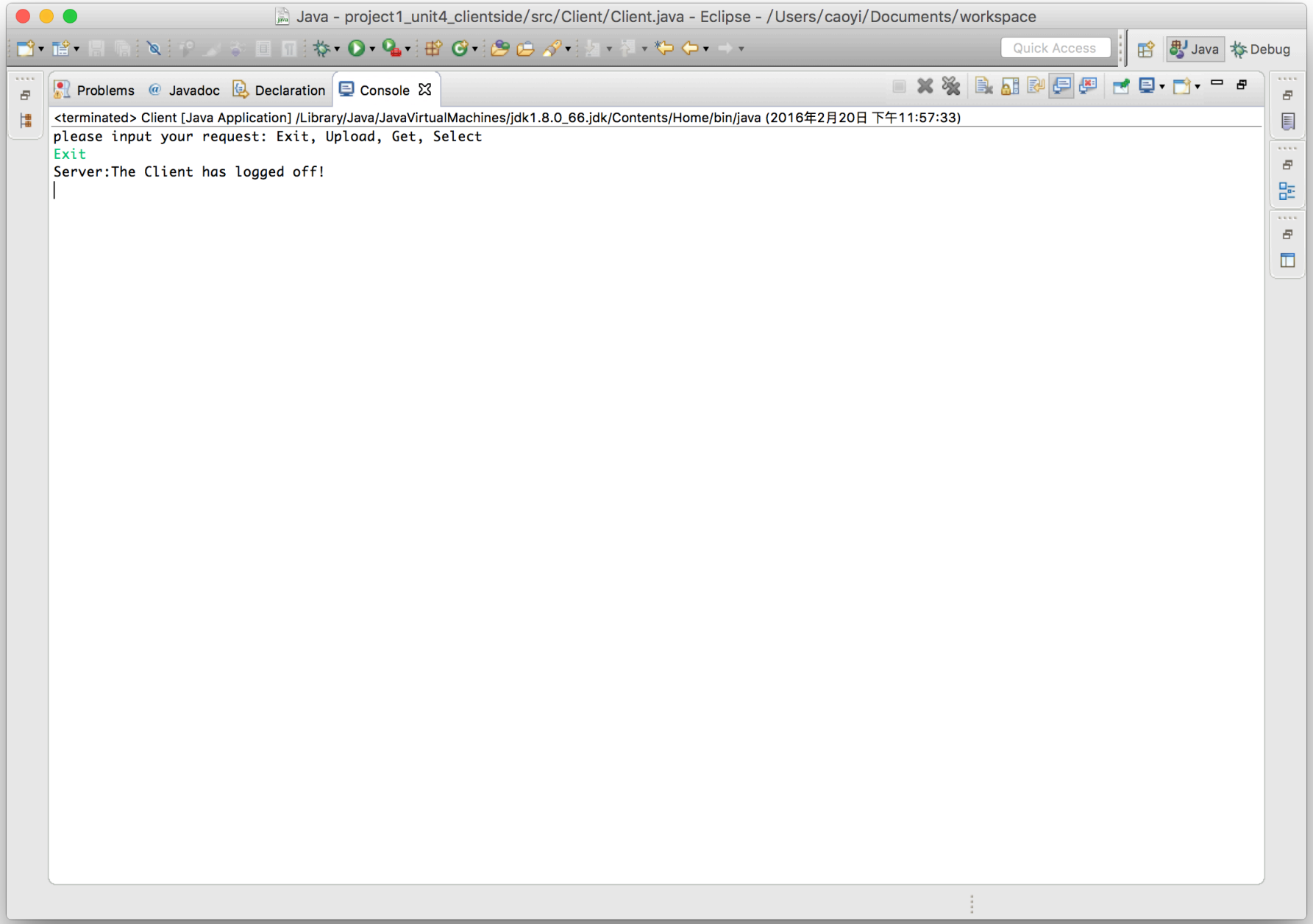
CarModelOptionsIO: bridge the communication to the server side

DefaultSocketClient: access the server side requests

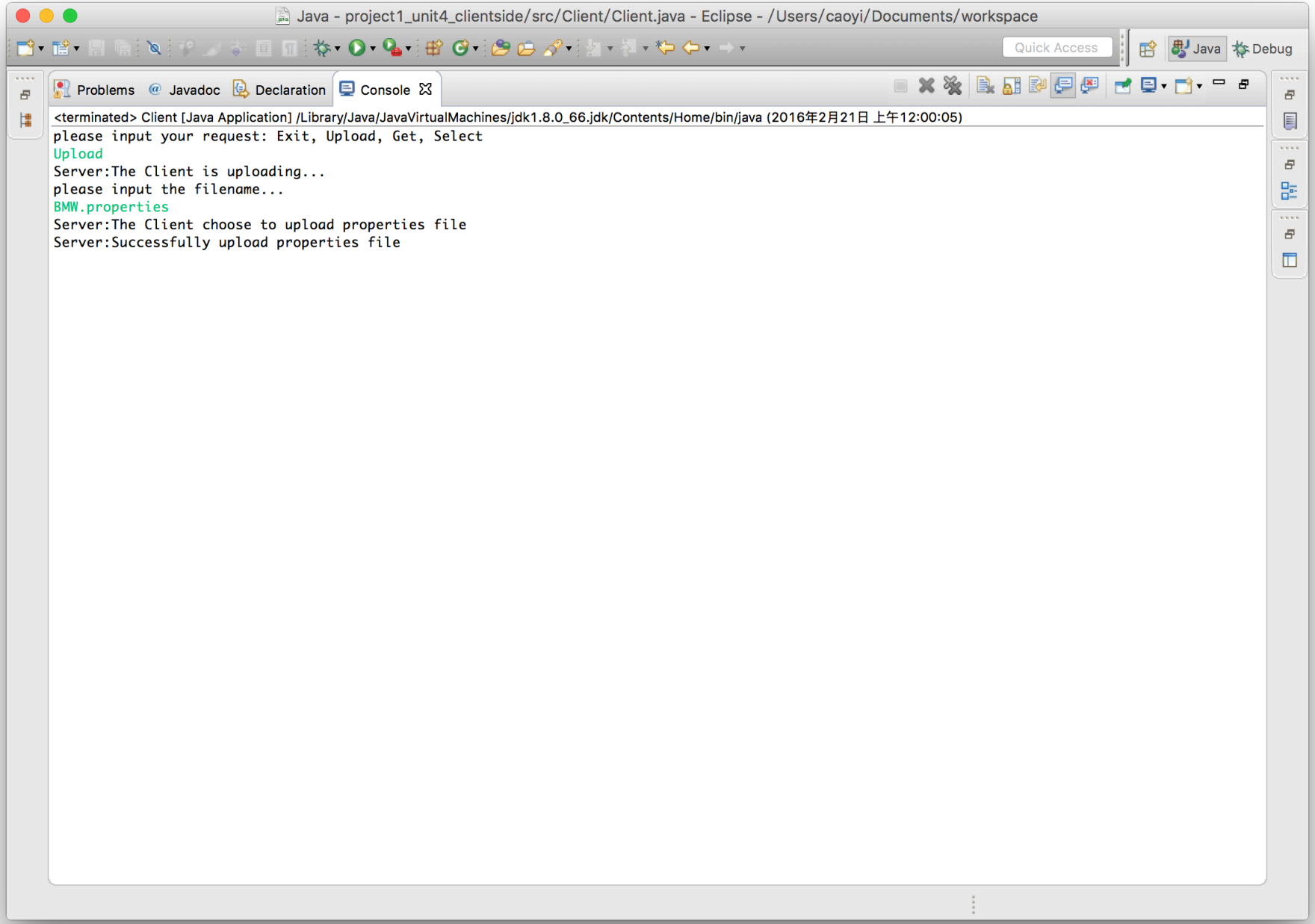
Client: start the client side;

SocketConstants: include the constants for socket

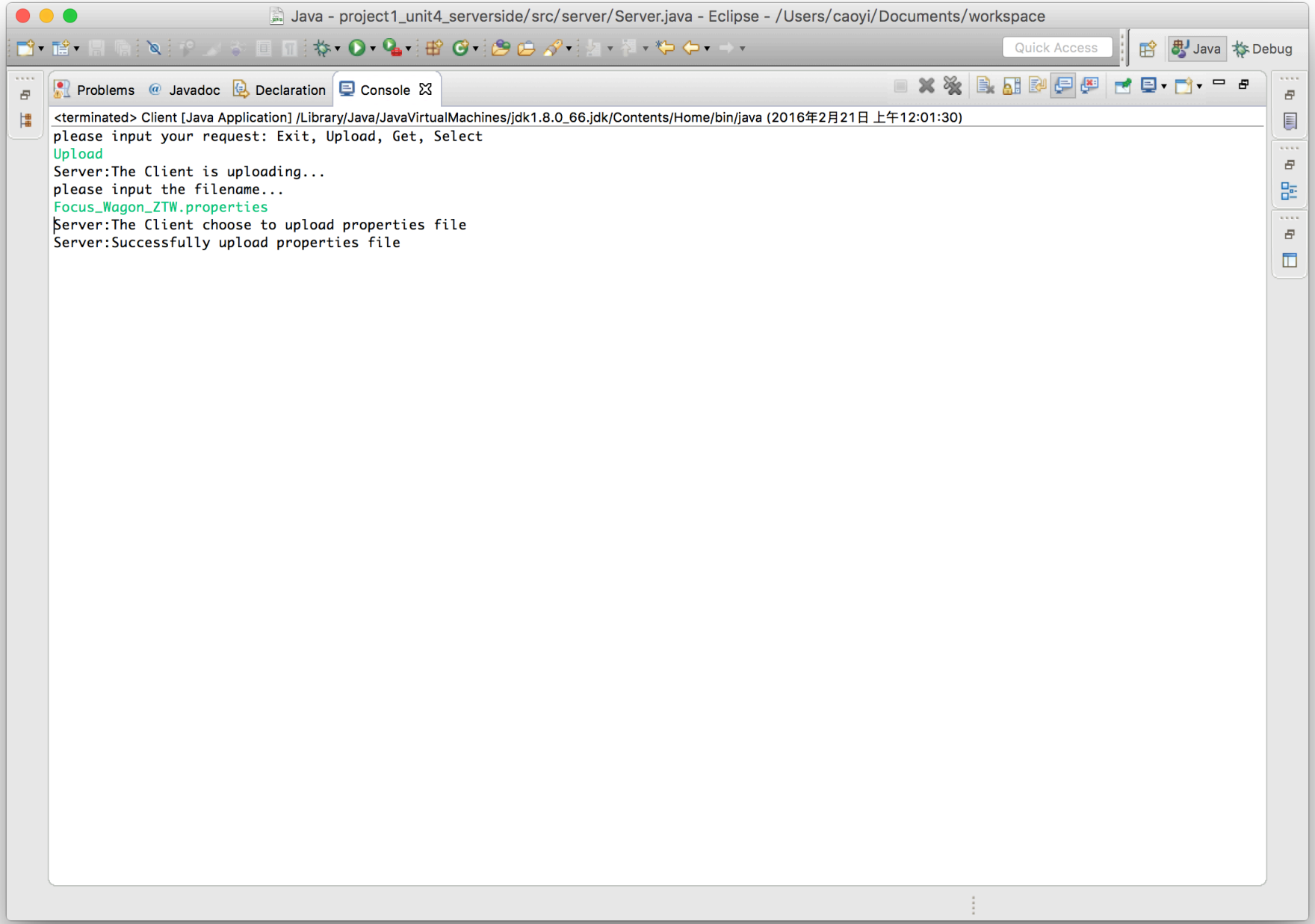
Test Exit


Java - project1_unit4_clientside/src/Client/Client.java - Eclipse - /Users/caoyi/Documents/workspace
Quick Access
Java
Debug
Problems
Javadoc
Declaration
Console
<terminated> Client [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_66.jdk/Contents/Home/bin/java (2016年2月20日 下午11:57:33)
please input your request: Exit, Upload, Get, Select
Exit
Server:The Client has logged off!

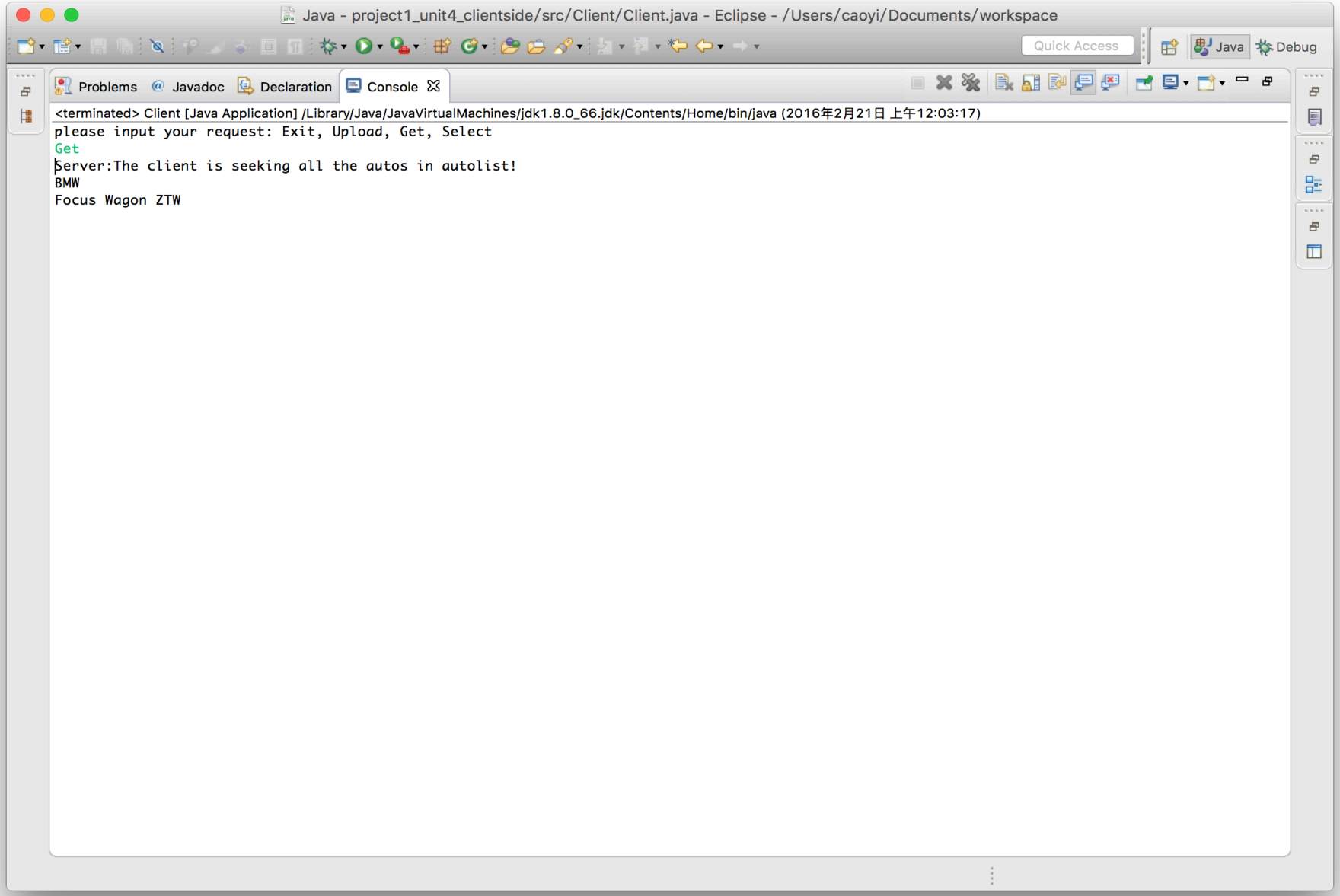
Text Upload


Java - project1_unit4_clientside/src/Client/Client.java - Eclipse - /Users/caoyi/Documents/workspace
Quick Access
Java
Debug
Problems
Javadoc
Declaration
Console
<terminated> Client [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_66.jdk/Contents/Home/bin/java (2016年2月21日 上午12:00:05)
please input your request: Exit, Upload, Get, Select
Upload
Server:The Client is uploading...
please input the filename...
BMW.properties
Server:The Client choose to upload properties file
Server:Successfully upload properties file

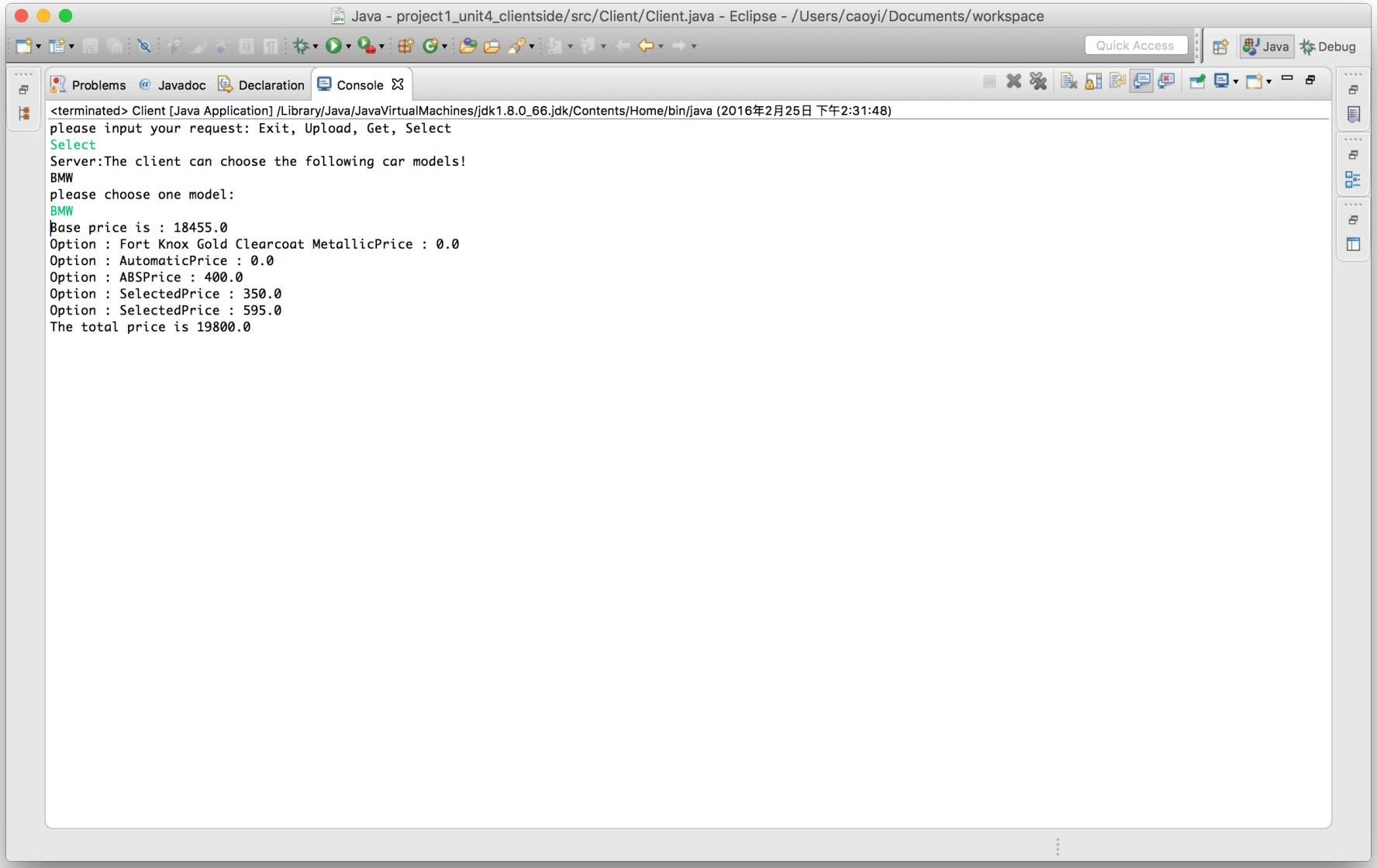
Text Upload


Java - project1_unit4_serverside/src/server/Server.java - Eclipse - /Users/caoyi/Documents/workspace
Quick Access
Java
Debug
Problems
Javadoc
Declaration
Console
<terminated> Client [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_66.jdk/Contents/Home/bin/java (2016年2月21日 上午12:01:30)
please input your request: Exit, Upload, Get, Select
Upload
Server:The Client is uploading...
please input the filename...
Focus_Wagon_ZTW.properties
Server:The Client choose to upload properties file
Server:Successfully upload properties file

Test Get


Java - project1_unit4_clientside/src/Client/Client.java - Eclipse - /Users/caoyi/Documents/workspace
Quick Access
Java
Debug
Problems
Javadoc
Declaration
Console
<terminated> Client [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_66.jdk/Contents/Home/bin/java (2016年2月21日 上午12:03:17)
please input your request: Exit, Upload, Get, Select
Get
Server:The client is seeking all the autos in autolist!
BMW
Focus Wagon ZTW

Test Select


Java - project1_unit4_clientside/src/Client/Client.java - Eclipse - /Users/caoyi/Documents/workspace
Quick Access
Java
Debug
Problems
Javadoc
Declaration
Console
<terminated> Client [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_66.jdk/Contents/Home/bin/java (2016年2月25日 下午2:31:48)
please input your request: Exit, Upload, Get, Select
Select
Server:The client can choose the following car models!
BMW
please choose one model:
BMW
Base price is : 18455.0
Option : Fort Knox Gold Clearcoat MetallicPrice : 0.0
Option : AutomaticPrice : 0.0
Option : ABSPrice : 400.0
Option : SelectedPrice : 350.0
Option : SelectedPrice : 595.0
The total price is 19800.0