**Documentation**

Andrew id: yiq

**Attention:**

This documentation is for the project 1 unit 4 of the course 18641 in CMU. Design details and test details are included.

**1: Client Design Details**

In this part, I add scale package named client. It includes four classes file: CarModelOptionIO.java, Client.java, DefaultSocketClient.java and SelectCarOption.java as required in project handout.

The following diagram is the package and class relationship in client part. For your convenience, I only draw out the associated packages in the part. Other packages such as exception won’t be included.

In summary, Client.java is just a wrapper and all specific implementations are done in DefaultSocketClient.java.

CarModelOptionIO.java is to read all car model file names from a directory named fileSets.txt in this project.

SelectionOption.java is to customize a car configuration when user choose a specific car model.

**2: Server Design Details**

In this part, I add scale package named server. It includes four classes file: AutoServer.java (interface), Server.java, DefaultSocketClient.java and BuildCarModelOptions.java as required in project handout.

The following diagram is the package and class relationship in server part. For your convenience, I only draw out the associated packages in the part. Other packages such as exception won’t be included.

In summary, Server.java is just a wrapper and all specific implementations are done in DefaultSocketClient.java.

AutoServer.java is an interface and it supports three methods that can make server can handle different input file types as required in handout. BuildModelOption.java implements AutoServer.java interface and call three methods that are specifically defined in ProxyAutoMobile as required in handout.

Just like what is demonstrated above, ProxyAutoMobile.java should implement AutoServer.java interface and give specific definition of these three methods and be called in BuildAutoOption.java.

Also in package util, I add a method called readAutoFromProperty() in Util.java to upload car model information with input argument is properties file type.

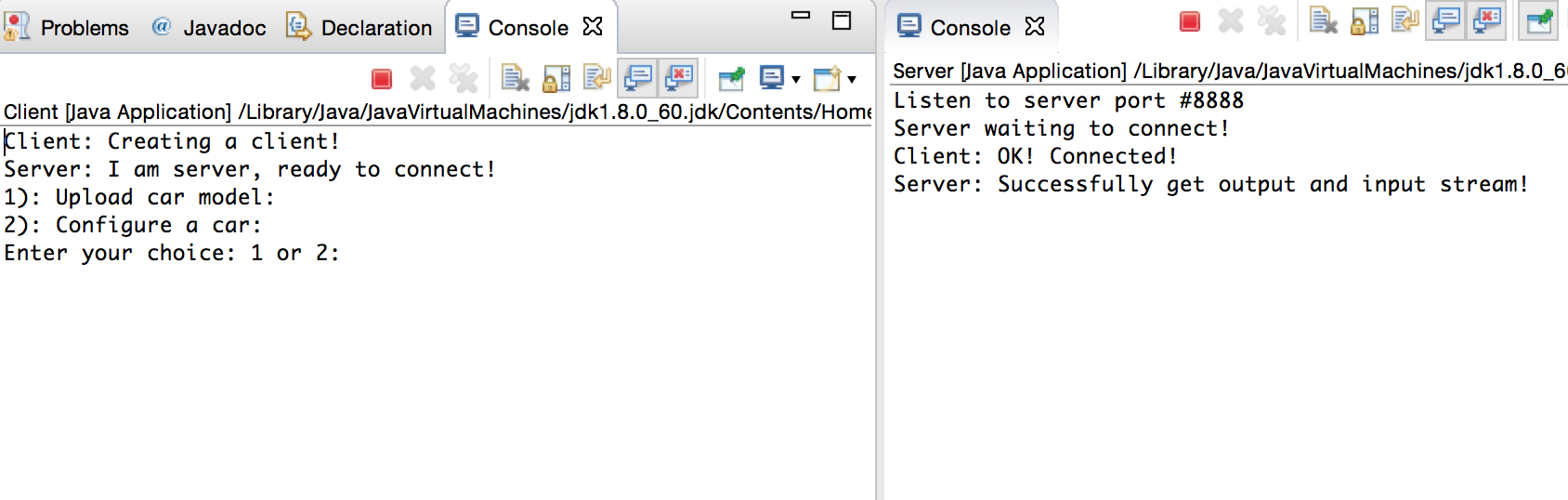
All client and server design details are strictly following project handout.

**3: Test details:**

In this part, I test client and server communication by uploading four car models information and customize two cars for user.

Please read following instruction carefully and for each step I will give my test snapshot:

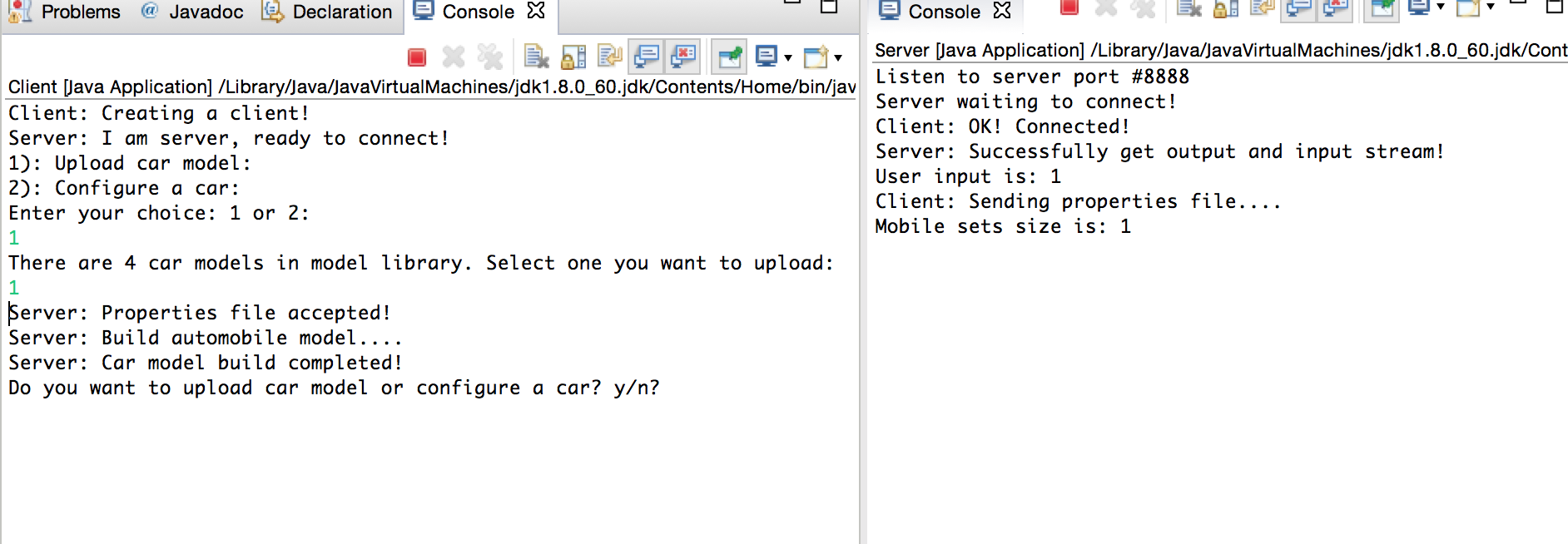
1: First build connection between server and client.



In this step, I firstly run Server.java and Client.java (the order can not be reversed.) Then from the two consoles information (left one is the client console and right one is the server console), I know that server and client connect successfully.

Then upon the client console information, next step is to ask user to input a choice, here since no car models have been uploaded so I first upload car models. So, go to step 2.

2: Upload car models

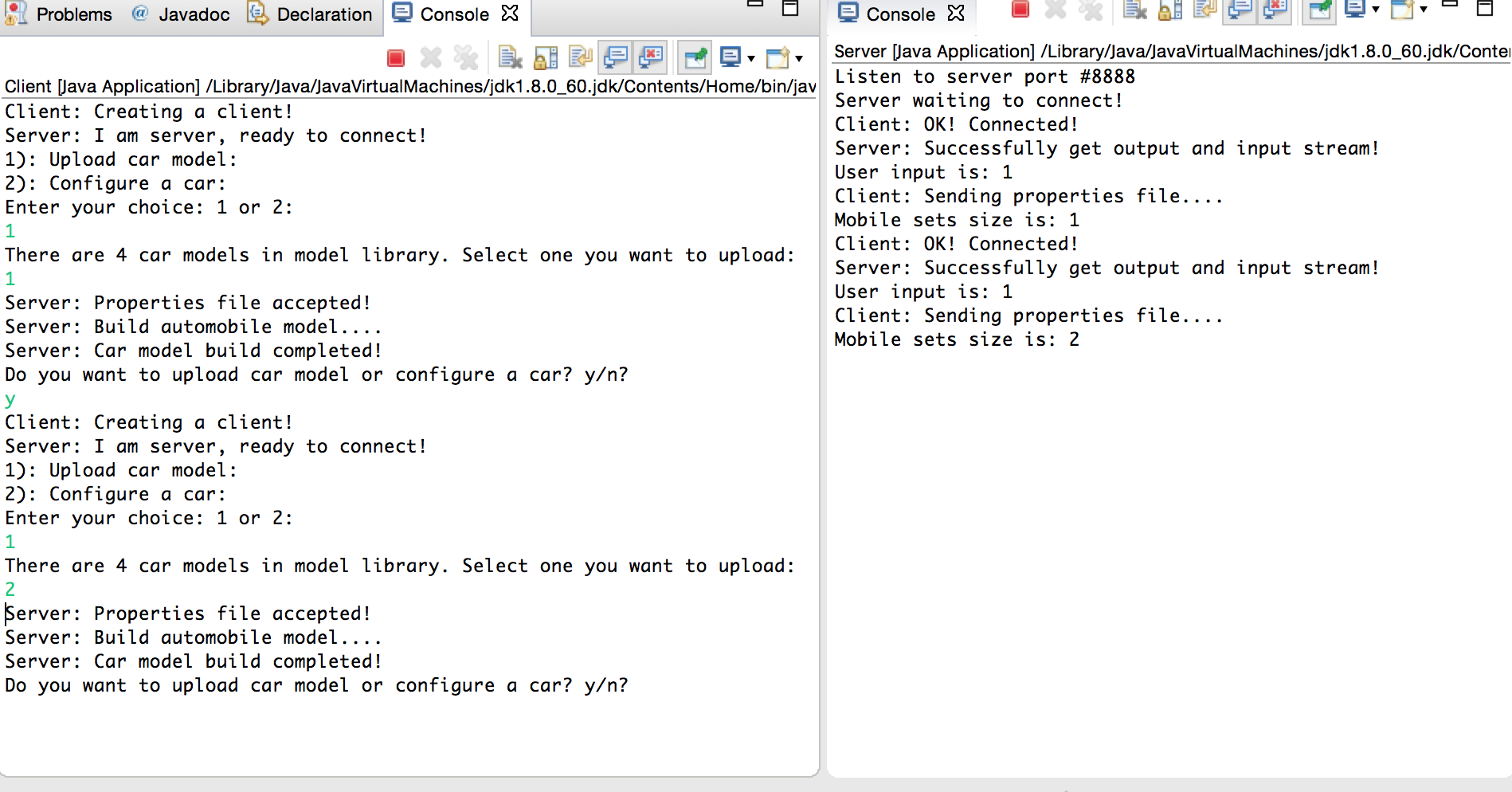


In this step, I input 1 to upload a car model, from the information, I know that there are 4 car models in the model library. So I first choose the first one to upload.

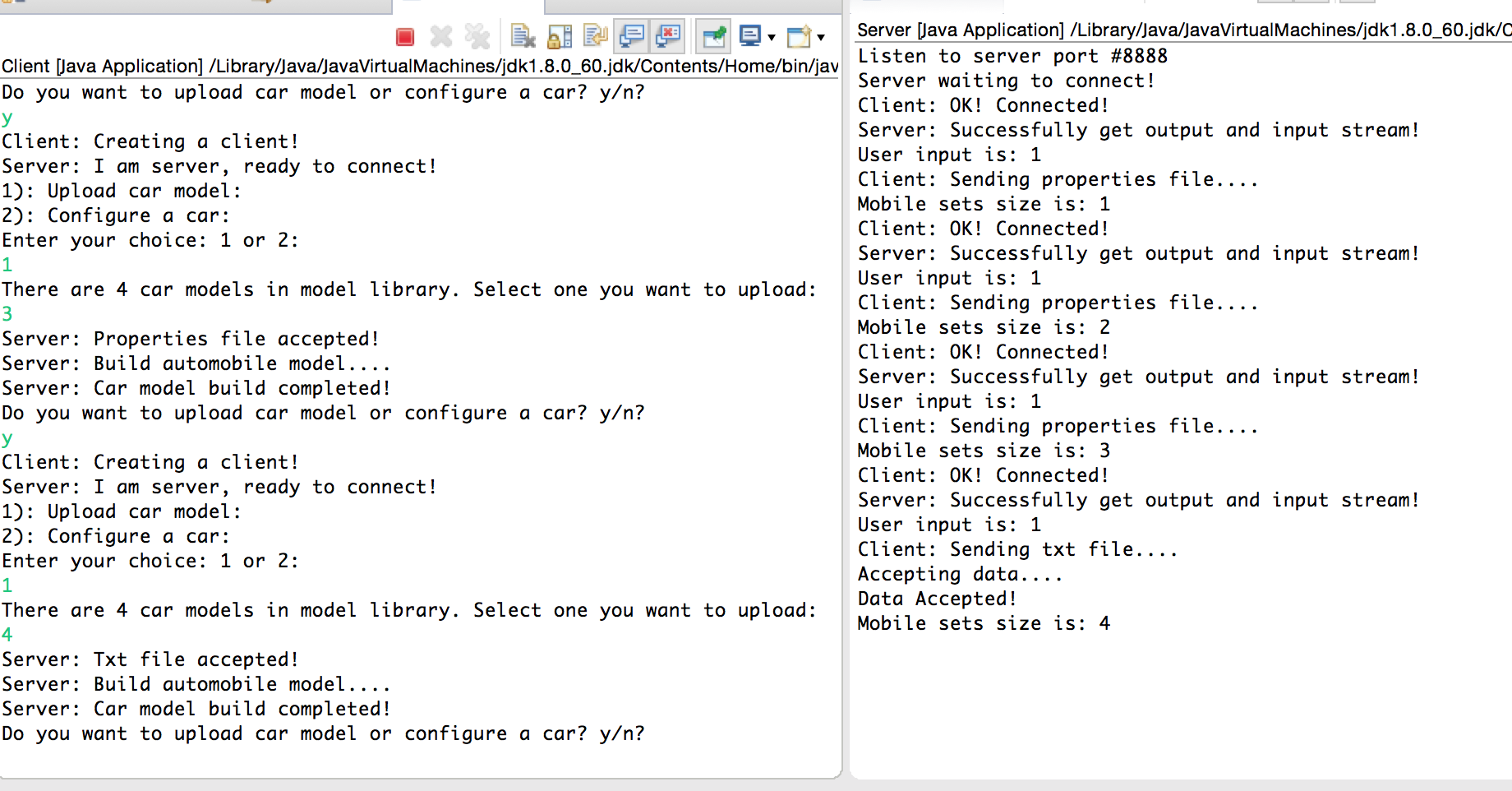
Then we can see from the client console, server send information to tell the client car model has been successfully uploaded.

From the server console, we know that the mobile sets size is increasing indicates that the car model has been uploaded successfully. Then I continue input y and digit to upload all these 4 car models into the mobile sets.

In this part, another part should be explained.

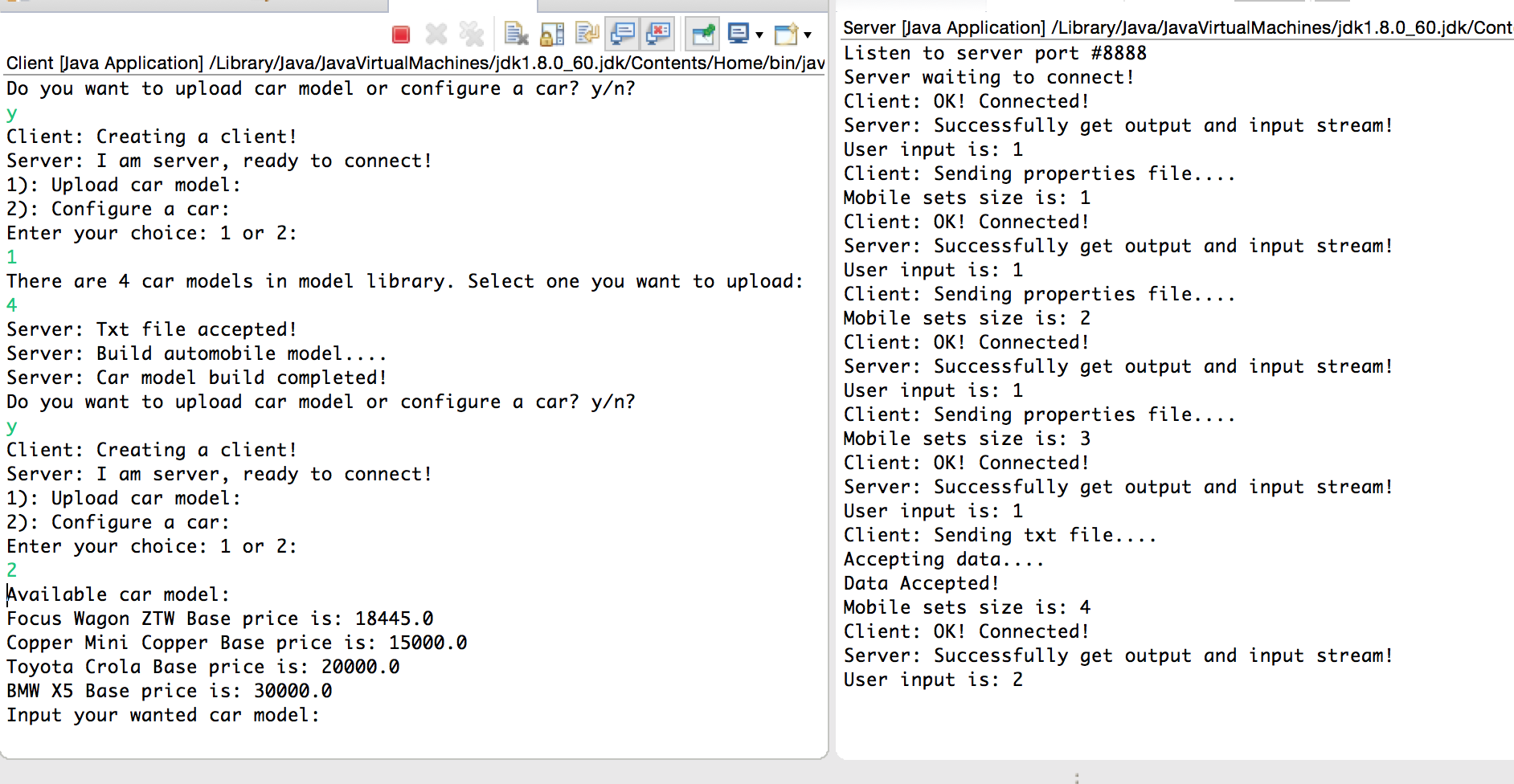


Now I upload two car models, then continue to upload next car models. Finally, we can get:



As you can see form the 4 input, this is a txt file not a properties file, so some different information is printed.

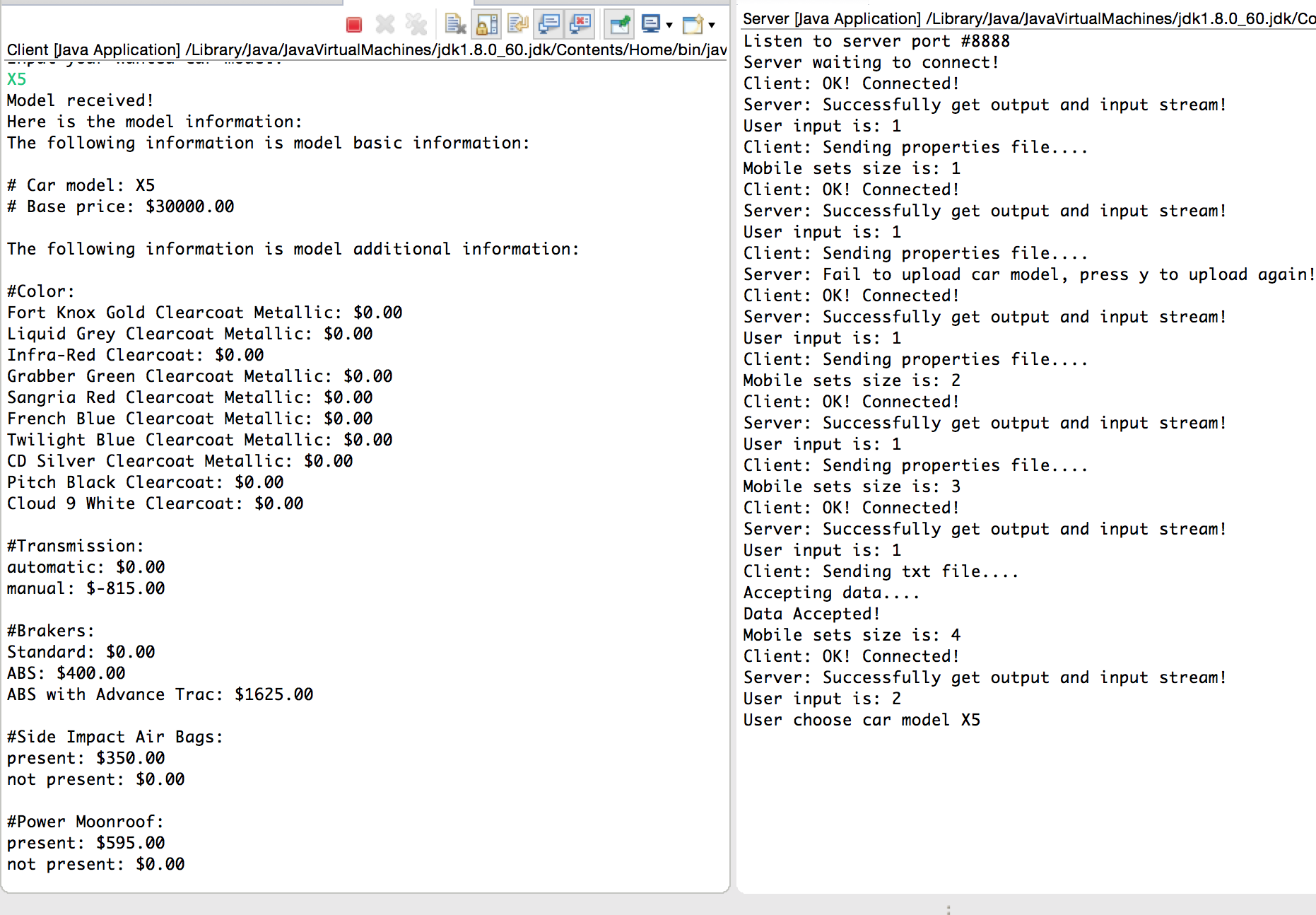
After upload all car models, now I try to customize user’s car. See below:



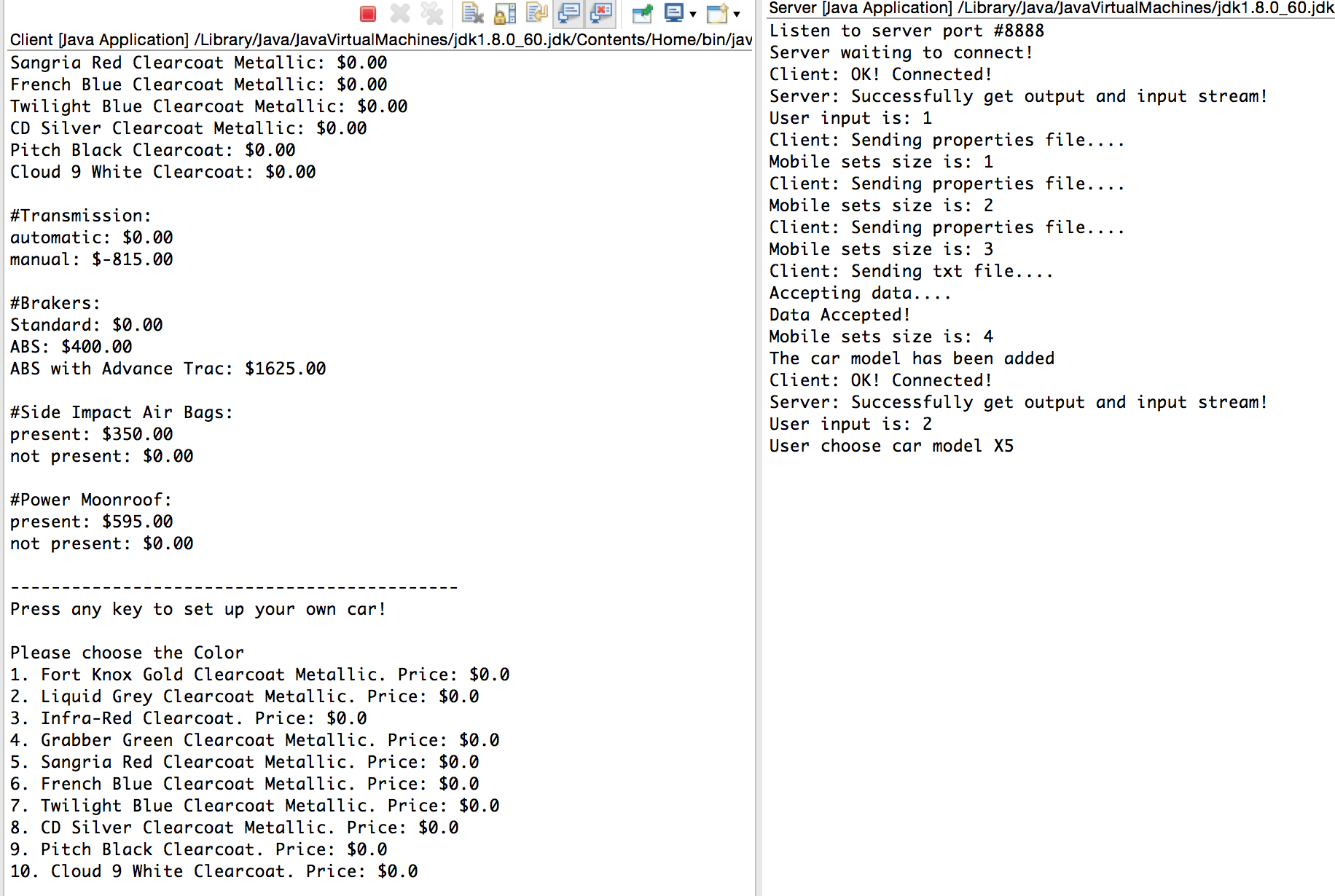
From the client console, we can know that there are 4 available car models, indicates that the previous steps success. Then go to step 3:

3: Customize two cars and print their information

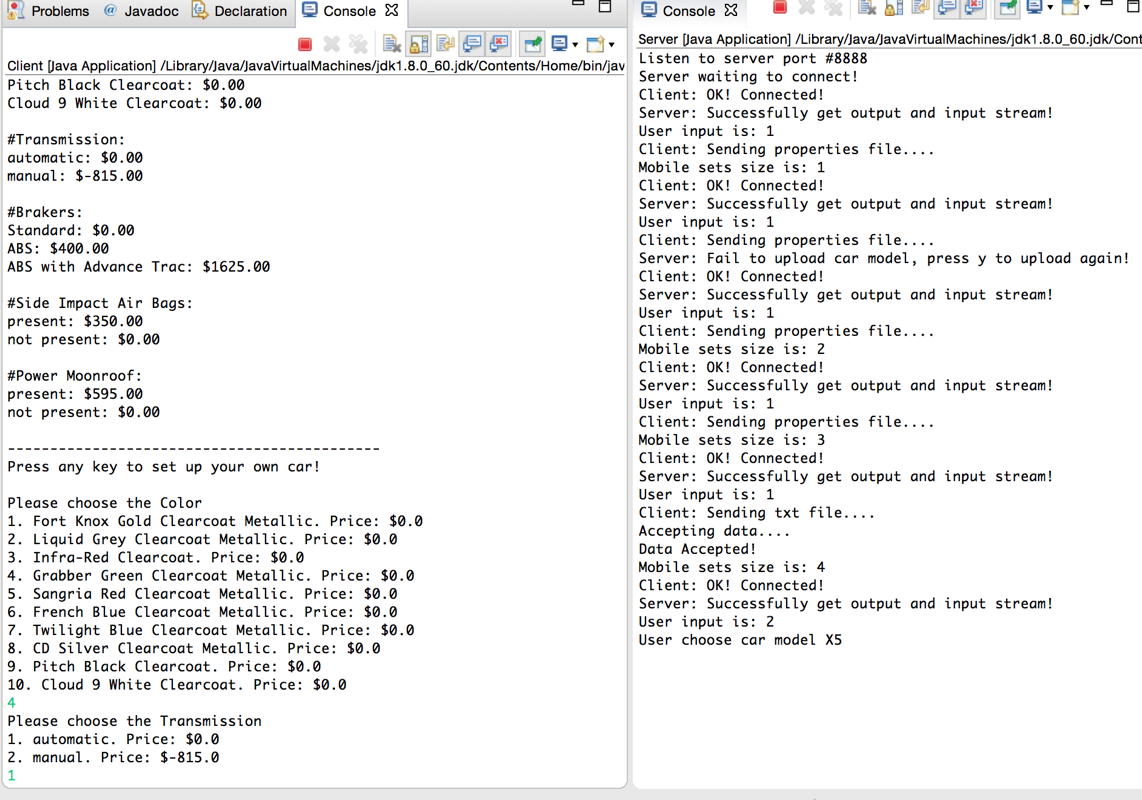
Firstly, I choose (capital x here) X5 model, then the client console will print out X5 model information:



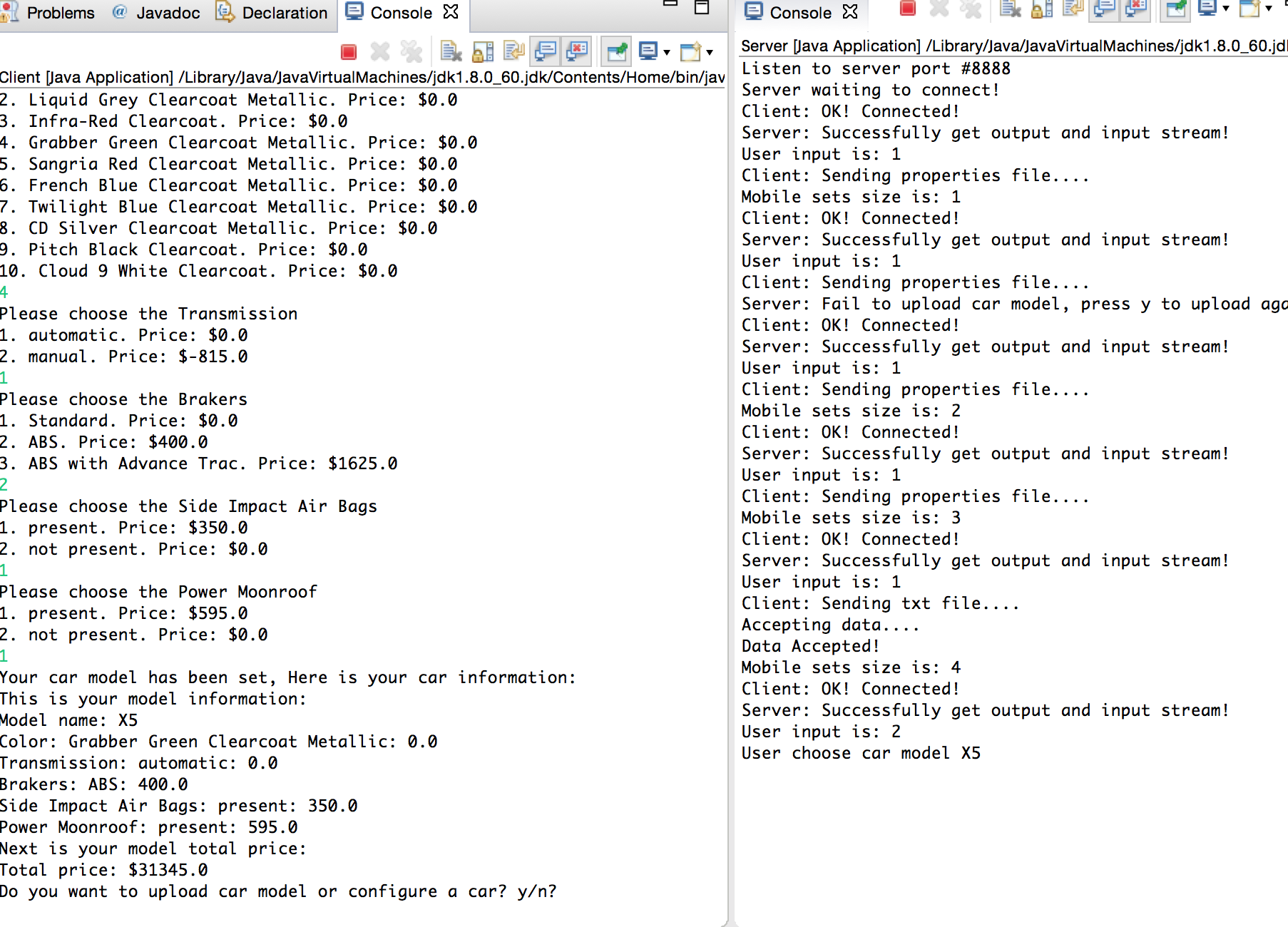
Then the client console prompts that press any key to continue to customize your own car:



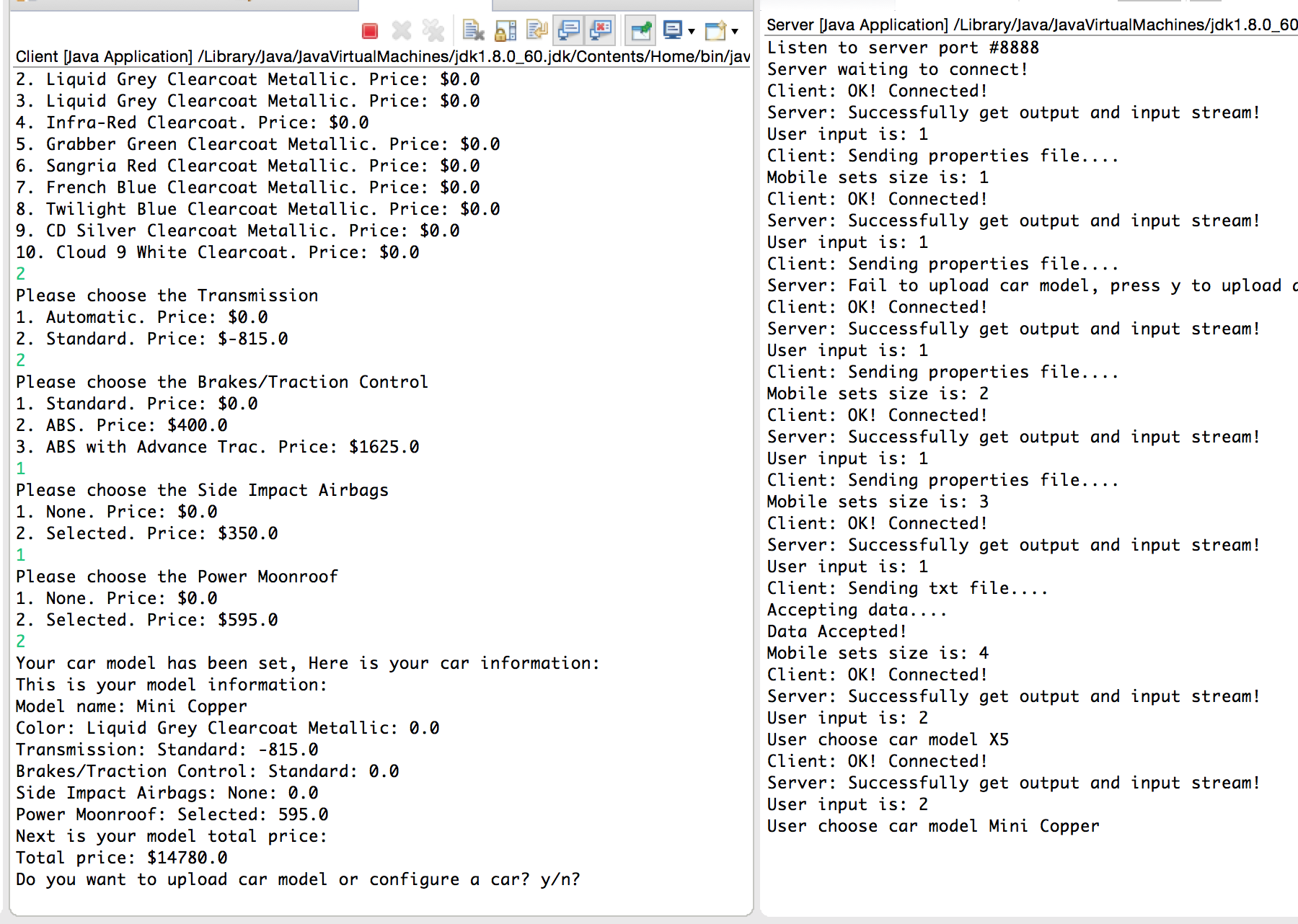
Then the client console asks for Color selection and I randomly choose 4 (digit required):



Then the client console asks for Transmission selection, I randomly choose 1. And also the next is other features selection, and I will skip this snapshot. Finally, the client console will print out you customize car information and your total price



And next I input y to customize another car, the steps are the same as above. I will skip that and give the final snapshot of this customize car.



And finally input n to exit this program.

Test completed.

**4: Special cases**

During your test, if prompt port already in use error, remember close both client and server session. Then test again.