1. The Driver class calls the testMultiThread() to do the following:

1. buildAuto from a file
2. pass buildAuto object to 2 new EditOption Threads.
3. start 2 EditOption Thread, one update “Power Moonroof- Selected ” to 100,

the other updated to 200

Results:

Since there is no guarantee which one goes first, the final result can be either 100 or 200

Option :Power Moonroof

[none,0.0]

[selected,100.0]

Option :Power Moonroof

[none,0.0]

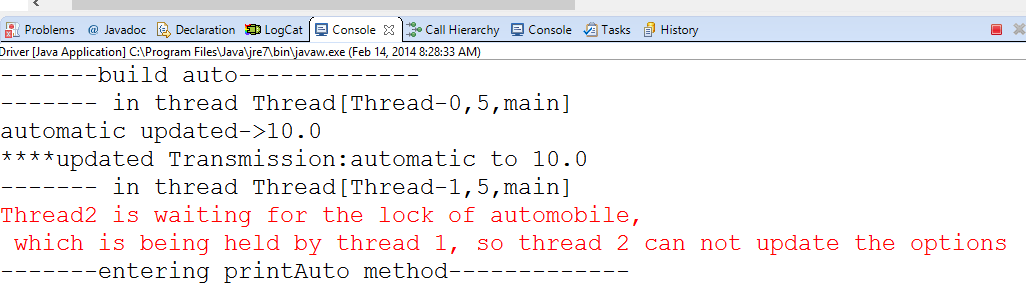
[selected,200.0]

2. The Driver class calls the testMultiThread2() to do the following:

1. buildAuto from a file
2. pass buildAuto object to 2 new EditOption Threads.
3. start 2 EditOption Thread, one update “Transmission- automatic ” to 10,

the other updated to 20. But I put Thread1 to sleep for 100 sec after it get the lock of that automobile object, so now Thread2 can not update the same Automobile object and the thread is blocked

The results show Thread2



with the sleep in the first operation - final output is that a new OptionSet gets added  
without any sleep - final output is that the automotive is unchanged (added then deleted)

\*\*\*\*updating Transmission:automatic to 10.0

automatic updated->10.0

Thread2 is waiting for the lock of automobile,

# which is being held by thread 1, so thread 2 can not update the options