```java
/*
 * Paul Park
 * November 10th 2020
 * RealEstateCLient program
 * description:
 * This program has properties and agents in company
 * User can fire/ add/ print/ sell/ change in the agent option
 * The agent will be printed in decreasing order for their profit made.
 */
import javax.swing.JOptionPane; // import JOptionPane
import javax.swing.JList; // import JList
public class RealEstateCompanyClient
{
	public static void main(String[] args)
	{
		// VARIABLE DECLARATION AND INITIALIZATION
		String mainLoop ="",
				agentName="",
				listingLoop = "",
				owner1="",
				owner2="";
		Agent CurrentAgent;
		// INSTANTIATE Broker
		Broker LePage = new Broker ("Royal LePage", "Finch Ave", "416-443-0300", 
"kfinch@trebnet.com");
		// INSTANTIATE Agents (broker will contain agent)
		Agent Smith = new Agent ("Kurt Smith", "163 Crestwood Thornhill", "645-
3456-3653", "krutSmith@gmail.com");
		Agent DeRose = new Agent ("Geraldino DeRose", "919 Riedel Street", "416-
714-5919", "GeraldinoDeRose@gmail.com");
		Agent Baker = new Agent ("Joseph N. Baker", "221 Pine Street", "647-955-
1686", "JosephNNBaker@gmail.com");
		Agent Kim = new Agent ("Jacob Kim", "35 Burberry Street", "645-3846-
8694", "JacobKim@gmail.com");
		Agent James = new Agent ("Carlos James", "15 Finch ave", "416-345-4759", 
"CJames1987@gmail.com");
		Agent Polov = new Agent ("Johnson Polov", "122 Finch West", "645-3846-
6723", "JohnsonP@gmail.com");
		Agent Lee = new Agent ("Carlos James", "67 Finch ave", "416-838-4356", 
"CJames1987@gmail.com");
		Agent Giron = new Agent ("Gian Giron", "153 Yonge Street", "6473 -565-
8248", "ggiron@gmail.com");
		// Instantiate listings
		Listing Porperty1 = new Listing ("Rhona Barrett", "Frankie Barrett", 
"505 Queens Quay W", "Toronto", "Ontario", "MK25J", 699000 , 699000, "May 14th 2020", 
0.07);
		Listing Porperty2 = new Listing ("Frank Pearson", "Emma Pearson", "436 -
33 Mill Street", "Toronto", "Ontario", "C846kI", 499000 , 500000, "June 16th 2020", 
0.06);
		Listing Porperty3 = new Listing ("Elisa Logan", "Greta Logan", "505 
Queens Quay W", "Winnipeg", "Manitoba", "MK25J", 685000 , 685000, "December 15th 
2019", 0.07);
		Listing Porperty4 = new Listing ("Moris Larsen", "Frank Zackett", "15 
Steels and Finch", "Toronto", "Ontario", "CV14J", 20000000 , 20000000, "Jan 10th 
2020", 0.07);
```

```
             Listing Porperty5 = new Listing ("Tina Garciat", "Justin Garcia", "17
Bathurst Street", "Toronto", "Ontario", "MSB94", 689888 , 690000, "April 14th 2020",
0.07);
             Listing Porperty6 = new Listing ("Brandon Bae", "Kelly Bae", "World On
Yonge", "Toronto", "Ontario", "9KD0W", 999000 , 999000, "August 27th 2020", 0.07);
             Listing Porperty7 = new Listing ("Suzy Kim", "Alex Park", "1400 Victoria
Street", "Victoria", "British Columbia", "L3K8D", 649000 , 649000, "September 7th
2020", 0.07);
             Listing Porperty8 = new Listing ("Ashley DeRose", "Jack DeRose", "5
Crestwood Bay", "Toronto", "Ontario", "L2KBI", 488000 , 488000, "August 20th 2020",
0.07);
             Listing Porperty9 = new Listing ("Robert Berry" , "Michael Berry",
"Dufferin Street and Dupont Street", "Toronto", "Ontario", "L3D83", 472900, 472900,
"August 10th 2020", 0.06);
             Listing Property10 = new Listing ("David Shin", "Charles Bae", "2083
Lake Shore Blvd W", "C8D3J", "Toronto", "Ontario", 599900, 599900, "July 27th 2020" ,
0.06);
             Listing Property11 = new Listing ("David Jung", "Amy Jung","55 East
Liberty Street TH101","K238", "Totonto", "Ontario",  799900, 799900, "March 13 2020",
0.06);
             // Add listings to Agent Smith
             Smith.addSoldList(Porperty8);
             Smith.addSoldList(Porperty5);
             // Add listings to DeRose
             DeRose.addSoldList(Porperty7);
             // Add Listing for GIron
             Giron.addSoldList(Porperty1);
             // Add listing to Baker
             Baker.addSoldList(Porperty3);
             // add listing to Polov
             Polov.addSoldList(Property11);
             Kim.addSoldList(Property10);
             // add listing to Kim
             Kim.addSoldList(Porperty9);
             Kim.addCurrentList(Porperty4); // <-- this is the most expensive
property (has been added to current listing so this agent did not sell yet)
             // add Listing to James
             James.addSoldList(Porperty6);
             // Add listing for Lee
             Lee.addSoldList(Porperty2);
             // List for Options (Every Option Should HAVE EXIT)
             String [] listingOptions = {"Print Current Listing", "Print Agents",
"Print Sold Listing", "Add Current Listing", "Sell Listing", "Change Agent","Add
Agent", "Fire This Agent" , "Exit Program"};
             // JList initializations
             JList <String> ListingMenu = new JList <String>(listingOptions);
             // add agents to LePage Company
             LePage.addAgent(DeRose);
             LePage.addAgent(Baker);
             LePage.addAgent(Smith);
             LePage.addAgent(Kim);
             LePage.addAgent(James);
             LePage.addAgent(Polov);
             LePage.addAgent(Lee);
             LePage.addAgent(Giron);
```

```
		// welcome message
		alertUser("Welcome to the Real Estate program"); // calls the alertUser
method to welcom user with a pop up message
		// print agent list
		LePage.printAgents();
		// alert user that list has been printed
		alertUser("Agent List has been printed!");
		while (!mainLoop.equals("Exit Program")) // while user does not want to
exit program
		{
			// check if user wanted to exit before going loop
			if (listingLoop.equals("Exit Program"))
			{
				break;
			}// end if
			// ask the user to select an agent
			agentName = askStringOption ("Which Agent would you like to
inspect?\nType in the name of the Agent" , "Type Agent Name");
			// when user entered agent is found
			if (LePage.findAgentName(agentName) == true)
			{
				alertUser(agentName + " was found!");
				CurrentAgent = LePage.getAgent(agentName);
				// reset listing loop
				listingLoop = "";
				// while loop to loop the listings
				while (!listingLoop.equals("Change Agent"))
				{
					listingLoop = choiceMenu(ListingMenu, agentName +
"'s Listing Options"); // this gives out what user chose
					if ( listingLoop.equals("Print Current Listing"))
					{
						alertUser("Printing Current Listing");
						System.out.println("Current Listing: ");
						// gets the userChosen Agent from Le Page and
then get the print list method from that Agent
						CurrentAgent.printCurrentList();
						System.out.println();
						System.out.println();
					}
					else if(listingLoop.equals("Print Sold Listing"))
					{
						alertUser("Printing Sold Listing for "+
agentName);

						// print sold listing option
						System.out.println("Sold Listing:");
						// from the user chosen agent print sold list
						CurrentAgent.printSoldList();
						System.out.println();
						System.out.println();
					}
					else if(listingLoop.equals("Add Current Listing"))
					{
						// calls client method to add to listing
```

```
                                        addCurrentListing(LePage.getAgent(agentName), 
agentName);
                                        System.out.println();
                                        System.out.println();
                                }
                                else if(listingLoop.equals("Sell Listing"))
                                {
                                        // when there is something in the listing
                                        if (CurrentAgent.getCurrentListings().size()
>0)
                                        {
                                                // they should enter the name1 and name 
2
                                                owner1 = askStringOption("What is the 
name of the Owner 1 you would like to sell","Input OwnerName1");
                                                owner2 = askStringOption("What is the 
name of the Owner 1 you would like to sell","Input OwnerName2");

                                                if 
(CurrentAgent.checkListOwnerNames(owner1, owner2)== true)
                                                {
                                                        // if the listing was found
        
        CurrentAgent.sellListOwnerNames(owner1, owner2);
                                                        alertUser("Listing Sold!\nNew 
List will be printing");
                                                        System.out.println("------NEW--
AGENTS---LIST");
                                                        LePage.printAgents();
                                                }
                                                else
                                                {
                                                        alertUser("Listing with " + 
owner1 +" and " + owner2 + " was not found!");
                                                }// end if (CurrentAgent listing if 
found)
                                        }
                                        else
                                        {
                                                alertUser("Nothing is in the listing 
right now!");
                                        }// end if(Check for something in list)
                                }
                                else if(listingLoop.equals( "Change Agent"))
                                {
                                        alertUser("Changing Agent");
                                        break; // breaks the loop

                                }
                                else if(listingLoop.equals( "Add Agent"))
                                {
                                        // client method to add agent to a specific 
Broker
                                        alertUser("Adding new Agent");
                                        addNewAgent(LePage);
```

```
                                                System.out.println();
                                                System.out.println();
                                        }
                                        else if(listingLoop.equals("Fire This Agent"))
                                        {
                                                LePage.fireAgent(agentName);
                                                alertUser(agentName + " has been fired!");
                                                alertUser("New Agent list will be printed");
                                                System.out.println("-------NEW---AGENTS-----
");

                                                LePage.printAgents();
                                                break; // the program breaks out of listing
loop after fireing an agent
                                        }
                                        else if(listingLoop.equals("Print Agents"))
                                        {
                                                alertUser("New Agent list will be printed");
                                                System.out.println();
                                                System.out.println("-------NEW---AGENTS-----
");

                                                LePage.printAgents();
                                        }
                                        else
                                        {
                                                // exit program
                                                alertUser("Exit Program");
                                                mainLoop = "Exit Program";
                                                break;
                                        }// end if
                                }// end listing loop
                        }
                        else
                        {
                                alertUser(agentName + " is not found!\nPlease Try
Again!"); // when user entered name is not found
                        }// end if (agent found or not)
                }// end main loop
                alertUser ("Thank you for using this program!");
        }// end main
        // Method that shows an alert message with user desired message
        public static void alertUser (String userSentence)
        {
                JOptionPane.showInternalMessageDialog(null, userSentence,"Pop-Up
Message" , JOptionPane.QUESTION_MESSAGE);
        }// end alertUser
        // method to store/ask user input (String version)
        public static String askStringOption(String message, String title)
        {
                String userMessage="";
                userMessage = JOptionPane.showInputDialog(null, message, title,
JOptionPane.QUESTION_MESSAGE);
                return userMessage;
        }// end askStringOption
        // method to store/ask user input (Double version)
        public static double askDoubleOption(String message, String title)
```

```
	{
		double userMessage;
		userMessage = Double.parseDouble(JOptionPane.showInputDialog(null,
message, title, JOptionPane.QUESTION_MESSAGE));
		return userMessage;
	}// end dou ble option
	// method that shows a menu option buy given List and JList
	public static String choiceMenu(JList<String> menu, String message)
	{
		JOptionPane.showInternalMessageDialog(null, menu, message ,
JOptionPane.INFORMATION_MESSAGE);
		return menu.getSelectedValue();
	}// end chooseFrom
	// client method that uses Listing set methods to add to the list
	public static void addCurrentListing(Agent SelectedAgent, String chosenAgent)
	{
		// create temp  Listing
		Listing TempList;
		alertUser("Adding Current Listing for "+ chosenAgent);
		TempList = new Listing(); // <-- only to add variables!! do not re-
use!!!
		// adding variables
		SelectedAgent.addCurrentList(TempList); // add a new list to currentList
		// variables for the new added list
		TempList.setOwnerName1(askStringOption ("What is the 1st Owner's
Name?","Owner 1 Name"));
		TempList.setOwnerName2(askStringOption ("What is the 2nd Owner's
Name?","Owner s Name"));
		TempList.setStreet(askStringOption ("What is the street of the
house?","Street"));
		TempList.setCity(askStringOption ("What is the city of the
house?","City"));
		TempList.setProvince(askStringOption ("What is the Province of the
house?","Province"));
		TempList.setPostal(askStringOption ("What is the postal code of the
house?","Postal"));
		TempList.setListingPrice(askDoubleOption ("What is the listing price of
the house?","ListingPrice"));
		TempList.setSoldPrice(askDoubleOption ("What is the sold price of the
house?","Sold"));
		TempList.setSoldDate(askStringOption ("What is the Sold Date of the
house?","Sold Date"));
		TempList.setCommissionRate(askDoubleOption ("What is the commission rate
for " + chosenAgent + "\n(Type in decimals eg. 6% => 0.06)","Commission Rate"));
	}//end AddCurrentListing
	public static void addNewAgent(Broker Company)		// client method to add
Agent to company using class methods
	{
		Agent newUserAgent;
		newUserAgent = new Agent (); // new agent with default constructor
		Company.addAgent(newUserAgent); // adds agent to company
		// ask user for the variables
		newUserAgent.setName(askStringOption("What is the Agent's Name?",
"Name"));
```

```
		newUserAgent.setAddress(askStringOption("What is the Agent's Address?", 
"Address"));
		newUserAgent.setPhoneNum(askStringOption("What is the Agent's Phone 
Number?", "Phone Number"));
		newUserAgent.setEmail(askStringOption("What is the Agent's Email?", 
"Email"));
		// alert user that agent has been added
		alertUser("Agent "+ newUserAgent.getName() + " has been added");
		alertUser("Printint new Agent List for " + Company.getName());
		System.out.println();
		Company.printAgents(); // this prints new list for company name
	}// end add new AGENT
}// end RealEstateClient
```

```
/*
 * Paul Park
 * November 10th 2020
 * Broker Class program
 * description: 
 * should contrain 
 *  Name, address, number, email, and an ArrayList<> of Agents.
 * 
 */
import java.text.DecimalFormat; // import deciaml Format
import java.util.ArrayList; // import arrayList
public class Broker 
{
	// Private Instance variable
	private String name;
	private String address;
	private String phoneNum;
	private String email;
	private ArrayList <Agent> TotalAgents;
	// constructor
	public Broker()
	{
		name = "Unknown";
		address = "Unknown";
		phoneNum = "Unknown";
		email = "Unknown";
		TotalAgents = new ArrayList <Agent>();
	}// end default constructor
	public Broker (String name, String address, String phoneNum,  String email)
	{
		this.name = name;
		this.address = address;
```

```
        this.phoneNum =phoneNum;
        this.email = email;
        TotalAgents = new ArrayList <Agent>();
}// end regular constructor
// set methods
public void setName (String name)
{
        this.name = name;
}// end set name
public void setAddress (String address)
{
        this.address = address;
}// end set address
public void setPhoneNum (String phoneNum)
{
        this.phoneNum = phoneNum;
}// and set phone num
public void setEmail (String email)
{
        this.email = email;
}//end set email
// get methods
public String getName()
{
        return name;
}// end get name
public String getAddress()
{
        return address;
}// end get address
public String getPhoneNum()
{
        return phoneNum;
}// end get phone num
public String getEmail()
{
        return email;
}// end get email
// methods and behavior
public void addAgent(Agent UserAgent)
{
        TotalAgents.add(UserAgent);
}// end add agent
// method to find an agent in the TotalAgent given the name of the agent
public boolean findAgentName(String userGivenName)
{
        for (Agent chosenAgent : TotalAgents)
        {
                // when chosen agent has the same name as user given name
                if (chosenAgent.getName().equalsIgnoreCase(userGivenName))
                {
                        return true;
                }// end if
        }// end for loop
        // when agent is not found
```

```
                return false;
        }// end find agent name
        // method that returns Agent
        public Agent getAgent(String agentName)
        {
                for (Agent chosenAgent : TotalAgents)
                {
                        // when chosen agent has the same name as user given name
                        if (chosenAgent.getName().equalsIgnoreCase(agentName))
                        {
                                return chosenAgent;
                        }// end if
                }// end for loop
                return TotalAgents.get(-1);
        }// end get agent method
        // method to find the total commssions rate for an agent
        public double findTCR(Agent UserAgent)// <-- agent you want to look at
        {
                Listing SelectedList;
                double totalRate =0;
                for (int i =0; i < UserAgent.getSoldListings().size(); i++)
                {
                        // UserAgent.getSoldListings() <-- this will get the soldListing
Array for that Agent
                        // find the commissionrate for that agent and add with the
totalRate
                        // UserAgent.getSoldListings().get(i) <-- this will get the
Agent's SoldListings
                        SelectedList = UserAgent.getSoldListings().get(i);
                        totalRate += SelectedList.getCommissionRate(); // <-- gets the
commission Rate for the listing
                }// end for loop
                return totalRate;
        }// end find TCR
        // method to organize agents for sold Listing
        public void organizeList()
        {
                // from TotalAgents find the total commission rate for each and compare
and swap
                for (int i =0; i < TotalAgents.size()-1; i++)
                {
                        for (int j=i+1; j <TotalAgents.size(); j++)
                        {
                                // the total CDR sold from the agent at i is smaller than
the one after
                                // swap it
                                if (TotalAgents.get(i).findAgentProfit() <
TotalAgents.get(j).findAgentProfit())
                                {
                                        // temp variable of Agent
                                        Agent Temp = TotalAgents.get(j);
                                        TotalAgents.set(j, TotalAgents.get(i)); // sets
total agents j to index i
                                        TotalAgents.set(i, Temp);
                                }// end if
```

```
				}// inner loop
			}// end for outerLoop
		}// end organizeList
		// method to give out the Agents Array
		public ArrayList<Agent> giveAgents()
		{
			return TotalAgents;
		}// giveAgents method
		// method to remove agent from list (fire agents from company)
		public void fireAgent(String agentName)
		{
			for (int i=0; i< TotalAgents.size(); i++)
			{
				if (TotalAgents.get(i).getName().equalsIgnoreCase(agentName))
				{
					// if the name is found remove
					TotalAgents.remove(i);
				}// end if
			}// end for loop
		}// end fire Agent
		// method to print the agents array
		public void printAgents ()
		{
			DecimalFormat df = new DecimalFormat ("#,##,###.00");
			// ***organize TotalAgents***
			organizeList();
			if (TotalAgents.size() == 0)
			{
				System.out.println(name + " does not have any Agents!");
			}
			else
			{
				organizeList();
				for (int i=0; i < TotalAgents.size(); i++)
				{
					System.out.println(
							"Agent Number "		+ (i+1)
						+ "\n" +
								"Name : "			+
	TotalAgents.get(i).getName()		+ "\n" +
								"Address : "		+
	TotalAgents.get(i).getAddress()	+ "\n" +
								"Phone number: "	+
	TotalAgents.get(i).getPhoneNum()	+ "\n" +
								"Email : "			+
	TotalAgents.get(i).getEmail()		+ "\n" +
								"Total Sold Commission Profit:
	$" + df.format(TotalAgents.get(i).findAgentProfit()) + "\n");
				}// end for loop
			}// end if statement
		}// end returnAgents
		// to String
		public String toString ()
		{
			// organize Agents
		String agents="";
		for (int i =0; i < TotalAgents.size(); i++)
		{
			agents += "Agent Number " + (i+1) + "\n" +
					TotalAgents.get(i);
		}// end for loop
		return ("--Broker Company--"			+ "\n" +
				"Name : "			+ name		+ "\n" +
				"Address : "	+ address	+ "\n" +
				"Phone number: "	+ phoneNum	+ "\n" +
				"Email : "			+ email		+ "\n\n"+
				"Agents List:\n"	+ agents);
	}// end toString
}// end brokerClass
```

```
/*
/*
 * Paul Park
 * November 10th 2020
 * Agent Class program
 * description:
 * should contain
 * name, phone, email,
 * ArrayList<> of Listings called CurrentListings<>
 * ArrayList<> of Listings called SoldListings<>
 */
import java.util.ArrayList; // import ArrayList
public class Agent
{
	// Private Instance variable
	private String name;
	private String address;
	private String phoneNum;
	private String email;
	private ArrayList <Listing> CurrentListings;
	private ArrayList <Listing> SoldListings;
	public Agent ()
	{
		// calls the default constructor for Broker Class
		name = "Name Not Found";
		address = "Address Not Found";
		phoneNum = "Phone Number Not Found";
		email = "Email Not Found";
		CurrentListings = new ArrayList <Listing>();
		SoldListings = new ArrayList <Listing>();
	}// end defualt constructor
```

```java
public Agent (String name, String address, String phoneNum, String email)
{
	this.name = name;
	this.address = address;
	this.phoneNum = phoneNum;
	this.email = email;
	CurrentListings = new ArrayList <Listing>();
	SoldListings = new ArrayList <Listing>();
}// end regular constructor
// set methods
public void setName (String name)
{
	this.name = name;
}// end set name
public void setAddress (String address)
{
	this.address = address;
}// end set address
public void setPhoneNum (String phoneNum)
{
	this.phoneNum = phoneNum;
}// end set phone num
public void setEmail (String email)
{
	this.email = email;
}// set email
// get methods
public String getName()
{
	return name;
}// end set name
public String getAddress()
{
	return address;
}// end get address
public String getPhoneNum()
{
	return phoneNum;
}// end get phone num
public String getEmail()
{
	return email;
}// end get email
// method to add to current listing
public void addCurrentList (Listing CurrentList)
{
	CurrentListings.add(CurrentList);
}// end addCurrentLIst
// METHOD TO CHECK IF ORGANIZE METHOD WORKS
public void addSoldList (Listing SoldListing)
{
	// sets the sold status to true when moving it to sold array
	SoldListing.setSold(true);
	SoldListings.add(SoldListing);
}// end addCurrentLIst
```

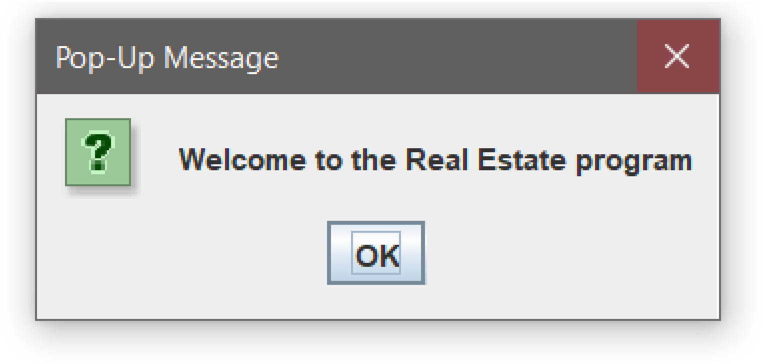
```
	// method to get the soldListing
	public ArrayList <Listing> getSoldListings()
	{
		return SoldListings;
	}// end get sold listing
	// method to get the CurrentListing
	public ArrayList <Listing> getCurrentListings()
	{
		return CurrentListings;
	}// end get current listings
	// method to check if the user entered listing is in current list
	public boolean checkListOwnerNames (String ownerName1, String ownerName2)
	{
		boolean value = false;
		//for loop to iterate through currentlistings
		for (int i=0; i < CurrentListings.size(); i++)
		{
			// if user entered owners were found in the list in current 
listings
			if ((CurrentListings.get(i).getOwnerName1()).equals(ownerName1) 
&& 

	(CurrentListings.get(i).getOwnerName2()).equals(ownerName2))
			{
				// then set value to true;
				value = true;
			}// end if
		}// end for loop
		return value;
	}// end checkListOwnerNames
	// method to add to sold listing
	public void sellListOwnerNames (String ownerName1, String ownerName2)
	{
		//for loop to iterate through currentlistings
		for (int i=0; i < CurrentListings.size(); i++)
		{
			// if statement to find the postal ownerName1 or ownerName2
			// changes everything to lower Case
			if ((CurrentListings.get(i).getOwnerName1()).equals(ownerName1) 
&& (CurrentListings.get(i).getOwnerName2()).equals(ownerName2))
			{
				// set temp variable
				Listing temp = CurrentListings.get(i);
				// remove the list from current listings
				CurrentListings.remove(i);
				// add it to the end of the soldListings
				SoldListings.add(temp);
			}// end if
		}// end for loop
	}// end sellListOwnerNames
	public void sellListNum (int listNumber)
	{
		//for loop to iterate through currentlistings
		for (int i=0; i < CurrentListings.size(); i++)
		{
```

```
                        if ( i == (listNumber-1))
                        {
                                // set sold status to true
                                CurrentListings.get(i).setSold(true);
                                // temp Listing
                                Listing Temp = CurrentListings.get(i);
                                // remove the list from current listings
                                CurrentListings.remove(i);
                                // add it to the end of the soldListings
                                SoldListings.add(Temp);
                        }// end if
                }// end for loop
        }// end sellListOwnerNames
        // method that gives out the total SOLD commission rate
        public double findAgentProfit()// <-- agent you want to look at TCR == total
commission rate
        {
                // temp variables
                Listing SelectedList;
                double totalRate =0;
                for (int i =0; i < SoldListings.size(); i++)
                {
                        // UserAgent.getSoldListings() <-- this will get the soldListing
Array for that Agent
                        // find the commissionrate for that agent and add with the
totalRate
                        // UserAgent.getSoldListings().get(i) <-- this will get the
Agent's SoldListings
                        SelectedList = SoldListings.get(i);
                        // commission rate * sold price == agent's profit
                        totalRate +=
(SelectedList.getCommissionRate()*SelectedList.getSoldPrice()); // <-- gets the
commission Rate for the listing
                }// end for loop
                return totalRate;
        }// end find TCR
        // method that gives the String componet of the List
        public String stringList(ArrayList <Listing> Lists)
        {
                // variable initialization and declaration
                int counter=1;
                String finalList = "";
                // organizes list first
                //      organizeList(Lists);
                for (Listing list : Lists)
                {
                        finalList += "-------------------"                     + "\n" +
                                     "Listing number " + counter               + "\n" +
                                     list
        + "\n" +
                                     "-------------------"                             +
"\n";
                        counter++;
                }// end for loop
                return finalList;
        }// end StringList method
        // method to print CurrentListings
        public void printCurrentList()
        {
                // variable initialization and declaration
                int counter=0;
                String finalList = "";
                if (CurrentListings.size() ==0)
                {
                        finalList = "No Listings were uploaded...";
                }
                else
                {
                        for (Listing CurrentList : CurrentListings)
                        {
                                counter++;
                                CurrentList.setListingNum(counter);
                                finalList += "------------------"              + "\n" +
                                                CurrentList                            + "\n"
+
                                                "------------------"                   +
"\n";
                        }// end for loop
                }// end if satement
                System.out.println(finalList);
        }// end StringList method
        // method to print the listing an agent has
        public void printSoldList()
        {
                // variable initialization and declaration
                int counter=0;
                String finalList = "";
                // organizes list first
                //      organizeList(SoldListings);
                if (SoldListings.size() ==0)
                {
                        finalList = "No Listings were uploaded...";
                }
                else
                {
                        for (Listing SoldList : SoldListings)
                        {
                                counter++;
                                SoldList.setListingNum(counter);
                                finalList += "------------------"      + "\n" +
                                                SoldList                               + "\n"
+
                                                "------------------"           + "\n";
                        }// end for loop
                }// end if statement
                System.out.println(finalList);
        }// end StringList method
        public String toString()
        {
                String currentL ="", soldL="";
                if (CurrentListings.size() ==0)
                {
                        currentL = "No Listings were uploaded...";
                }
                else
                {
                        // uses string list method and adds it to the currentL
                        currentL = stringList(CurrentListings);
                }// end if for current listings
                if (SoldListings.size() == 0)
                {
                        soldL = "None were sold...";
                }
                else
                {
                        // uses the stringList method and adds it to the soldL
                        soldL = stringList(SoldListings);
                }// end if for sold listings
                return ("Name of Agent: "                   + name
                        + "\n" +
                                name        + "'s Address: "        + address
                + "\n" +
                                name        + "'s Phone Number: "   + phoneNum
        + "\n" +
                                name        + "'s Email: "              + email
                        + "\n" +
                                "====================="
                        + "\n" +
                                "Current Listings...\n"                 +currentL
                        + "\n" +
                                "====================="
                        + "\n" +
                                "Sold Listings...\n"                    +soldL
                        + "\n" +
                                "=====================" );
        }// end to String
}// end Agent Class
```

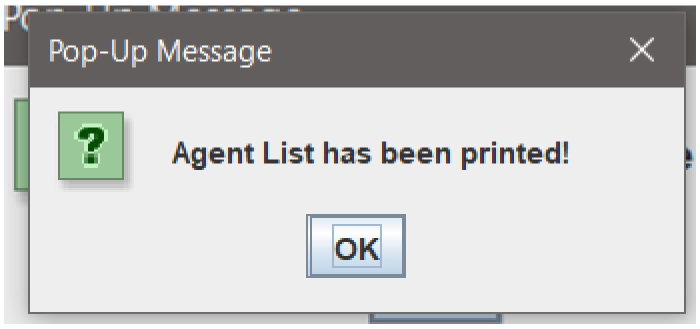
```java
/*
 * Paul Park
 * November 10th 2020
 * Listing Class program
 * description:
 * should contain
 * listingNum, ownerName1, ownerName2, street,
 * city, province, postal, listingPrice, soldPrice,
 * sell date, sold  (Boolean), commissionRate (as a percentage).
 */
import java.text.DecimalFormat; // import decimal format
public class Listing
{
	// Private Instance variable
	private int listingNum;
	private String ownerName1;
	private String ownerName2;
	private String street;
	private String city;
	private String province;
	private String postal;
	private double listingPrice;
	private double soldPrice;
	private String soldDate;
	private boolean sold;
	private double commissionRate;
	public Listing ()
	{
		listingNum = 1; // listing num will increase automatically when printed
using Agent class
		ownerName1 = "N/A";
		ownerName2 = "N/A";
		street = "N/A";
		city = "N/A";
		province = "N/A";
		postal = "N/A";
		listingPrice = 0;
		soldPrice = 0;
		soldDate = "N/A";
		sold = false;
		commissionRate = 0.06; // 0.6 is average commission rate
	}// end default constructor
	public Listing (String ownerName1, String ownerName2, String street, String
city, String province,
			String postal, double listingPrice, double soldPrice, String
soldDate, double commissionRate)
	{
		listingNum = 1;
		this.ownerName1 = ownerName1;
		this.ownerName2 = ownerName2;
		this.street = street;
		this.city = city;
		this.province = province;
		this.postal = postal;
		this.listingPrice = listingPrice;
		this.soldPrice = soldPrice;
		this.soldDate = soldDate;
		sold = false;
		this.commissionRate =commissionRate;
	}//end regular Constructor
	public Listing (int listingNum, String ownerName1, String ownerName2, String 
street, String city, String province,
			String postal, double listingPrice, double soldPrice, String 
soldDate, boolean sold, double commissionRate)
	{
		this.listingNum = listingNum;
		this.ownerName1 = ownerName1;
		this.ownerName2 = ownerName2;
		this.street = street;
		this.city = city;
		this.province = province;
		this.postal = postal;
		this.listingPrice = listingPrice;
		this.soldPrice = soldPrice;
		this.soldDate = soldDate;
		this.sold = sold;
		// commissionRate should be calculated with formula
		this.commissionRate =commissionRate;
	}//end regular Constructor
	// set methods
	public void setListingNum(int listingNum)
	{
		this.listingNum = listingNum;
	}// end set listing
	public void setOwnerName1(String ownerName1)
	{
		this.ownerName1 = ownerName1;
	}// end set owner name1
	public void setOwnerName2(String ownerName2)
	{
		this.ownerName2 = ownerName2;
	}// end set owner name2
	public void setStreet(String street)
	{
		this.street = street;
	}// end set street
	public void setCity(String city)
	{
		this.city = city;
	}// set city
	public void setProvince(String province)
	{
		this.province = province;
	}// end set province
	public void setPostal(String postal)
	{
		this.postal = postal;
	}// end set postal
	public void setListingPrice(double listingPrice)
	{
		this.listingPrice = listingPrice;
	}// end set listing price
	public void setSoldPrice(boolean sold)
	{
		this.sold = sold;
	}// end set sold price
	public void setSoldPrice(double soldPrice)
	{
		this.soldPrice = soldPrice;
	}// end set sold price
	public void setSoldDate(String soldDate)
	{
		this.soldDate = soldDate;
	}// end set sold date
	public void setSold(boolean sold)
	{
		this.sold = sold;
	}// end set sold  (boolean)
	public void setCommissionRate(double commissionRate)
	{
		this.commissionRate = commissionRate;
	}// end set commission rate
	// get method
	public int getListingNumb ()
	{
		return listingNum;
	}// end get listing number
	public String getOwnerName1 ()
	{
		return ownerName1;
	}// end get owner name 1
	public String getOwnerName2 ()
	{
		return ownerName2;
	}// end get owner name 2
	public double getSoldPrice ()
	{
		return soldPrice;
	}// end get sold price
	public double getCommissionRate ()
	{
		return commissionRate;
	}// end get commission rate
	// do not need to write get method for other variables because it wont be used
	public void setAddress(String street, String city, String province, String
postal)
	{
		this.street = street;
		this.city = city;
		this.province = province;
		this.postal = postal;
	}// end set address
	// grouped money variables
	public void setCosts(double listingPrice, double soldPrice, boolean sold,
String soldDate)
```

```java
       {
             this.listingPrice = listingPrice;
             this.soldPrice = soldPrice;
             this.sold = sold;
             this.soldDate = soldDate;
       }// end setCosts
       // get method
       public String toString()
       {
             // instantiate decimal format
             DecimalFormat df = new DecimalFormat ("#,##,###.00");
             return ("Listing Number: "       + listingNum
             + " \n" +
                          "Owner Name 1: "          + ownerName1
                    + " \n" +
                          "Owner Name 2: "          + ownerName2
                    + " \n" +
                          "Street: "                       + street
                                 + " \n" +
                          "City: "                         + city
                                       + " \n" +
                          "Province: "              + province
                          + " \n" +
                          "Postal: "                       + postal
                                 + " \n" +
                          "Listing Price: "         + df.format(listingPrice)
             + " \n" +
                          "Sold Price: "                   + df.format(soldPrice)
                          + " \n" +
                          "Sold Date: "                    + soldDate
                                 + " \n" +
                          "Sold Status: "           + sold
                                 + " \n" +
                          "Commission Rate: "       +
df.format((commissionRate*100)) + "%" + "\n" +
                          "Agent gets : " + "$" +
df.format((soldPrice*commissionRate)));
       }//end toString
}// end Listing Class
```

Output

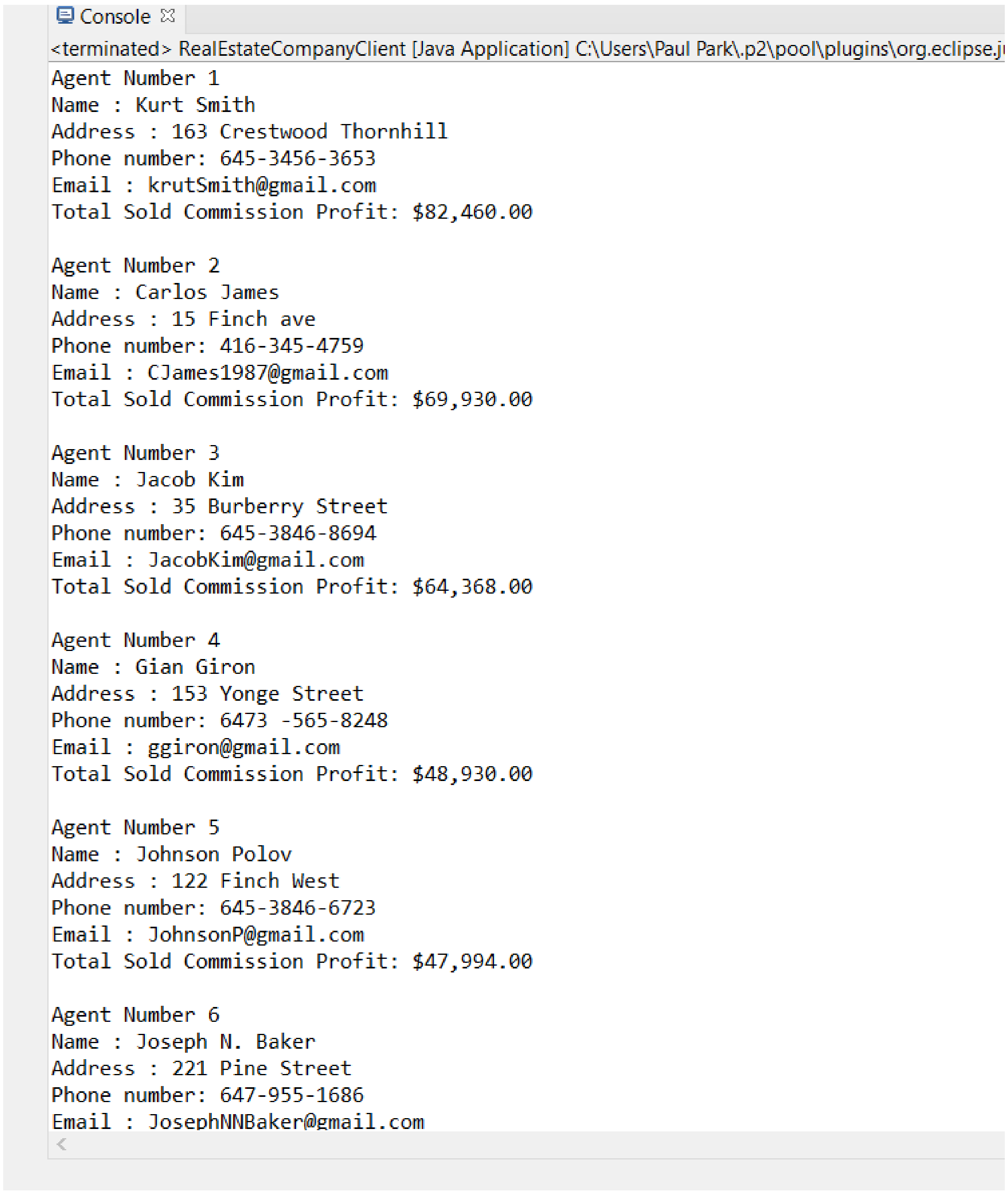

Pop-Up Message

```
Welcome to the Real Estate program
```


Pop-Up Message

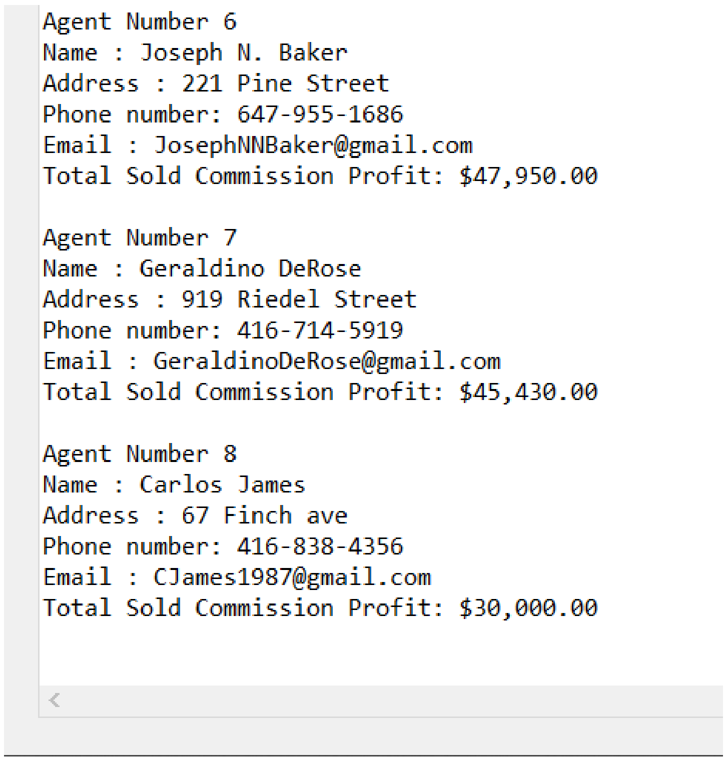
```
Agent List has been printed!
```

The agent with the most profit will be printed first.



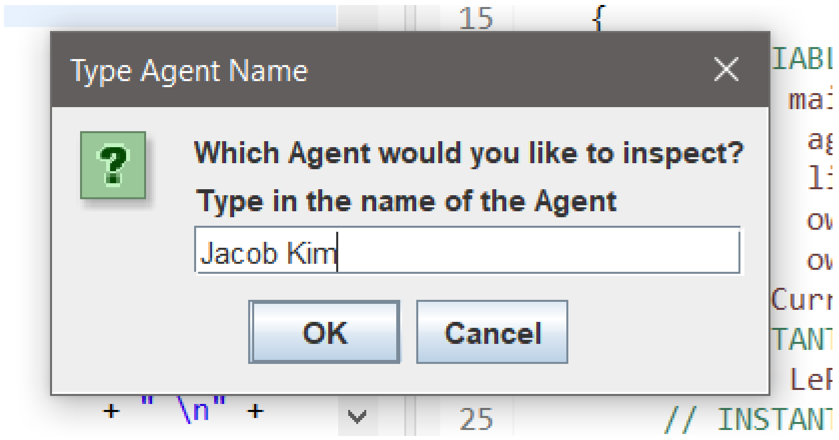
Console

```
<terminated> RealEstateCompanyClient [Java Application] C:\Users\Paul Park\.p2\pool\plugins\org.eclipse.j
Agent Number 1
Name : Kurt Smith
Address : 163 Crestwood Thornhill
Phone number: 645-3456-3653
Email : krutSmith@gmail.com
Total Sold Commission Profit: $82,460.00

Agent Number 2
Name : Carlos James
Address : 15 Finch ave
Phone number: 416-345-4759
Email : CJames1987@gmail.com
Total Sold Commission Profit: $69,930.00

Agent Number 3
Name : Jacob Kim
Address : 35 Burberry Street
Phone number: 645-3846-8694
Email : JacobKim@gmail.com
Total Sold Commission Profit: $64,368.00

Agent Number 4
Name : Gian Giron
Address : 153 Yonge Street
Phone number: 6473 -565-8248
Email : ggiron@gmail.com
Total Sold Commission Profit: $48,930.00

Agent Number 5
Name : Johnson Polov
Address : 122 Finch West
Phone number: 645-3846-6723
Email : JohnsonP@gmail.com
Total Sold Commission Profit: $47,994.00

Agent Number 6
Name : Joseph N. Baker
Address : 221 Pine Street
Phone number: 647-955-1686
Email : JosephNNBaker@gmail.com
```



```
Agent Number 6
Name : Joseph N. Baker
Address : 221 Pine Street
Phone number: 647-955-1686
Email : JosephNNBaker@gmail.com
Total Sold Commission Profit: $47,950.00

Agent Number 7
Name : Geraldino DeRose
Address : 919 Riedel Street
Phone number: 416-714-5919
Email : GeraldinoDeRose@gmail.com
Total Sold Commission Profit: $45,430.00

Agent Number 8
Name : Carlos James
Address : 67 Finch ave
Phone number: 416-838-4356
Email : CJames1987@gmail.com
Total Sold Commission Profit: $30,000.00
```

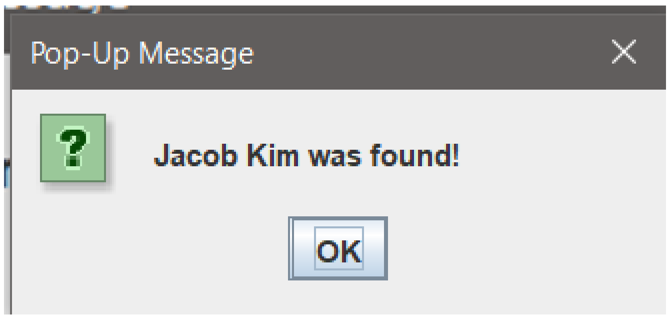
User types in the agent they want to inspect (Capital and lower case does not matter)



Type Agent Name

Which Agent would you like to inspect?
Type in the name of the Agent

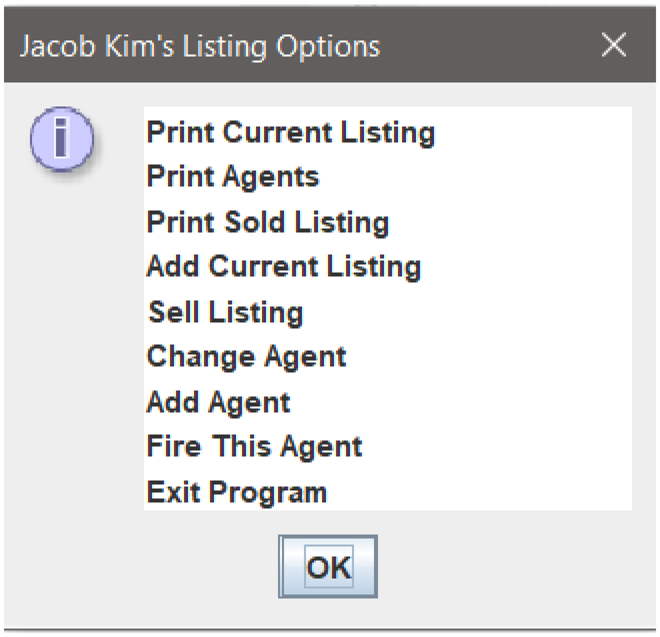
Jacob Kim



Pop-Up Message

Jacob Kim was found!

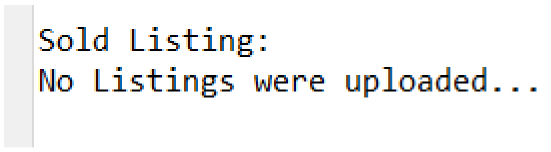
Listing Option for Jacob Kim will show…


Jacob Kim's Listing Options

User can choose any option. (Click and press ok)

“Print sold listing” option would only work if user has sold listing from current listing.

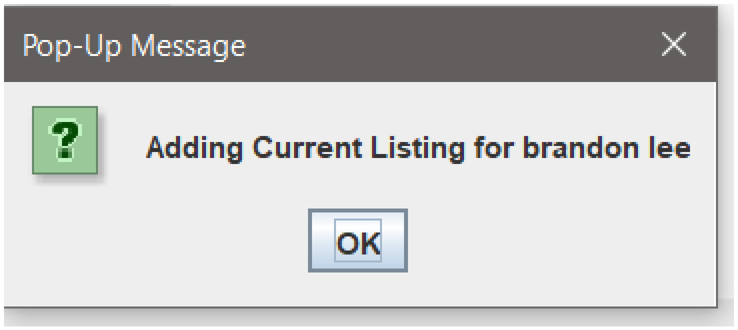
However if there are nothing in the current listing, then it will print out that the listing is empty.


```
Sold Listing:
No Listings were uploaded...
```

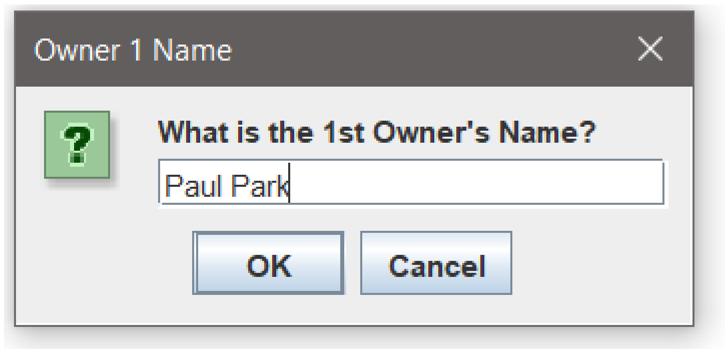
Same goes for “Print Current Listing” option.

When user wants to add a listing, the computer will ask for:

1st owner name, 2nd owner name, street of house, postal code… etc (shown below)

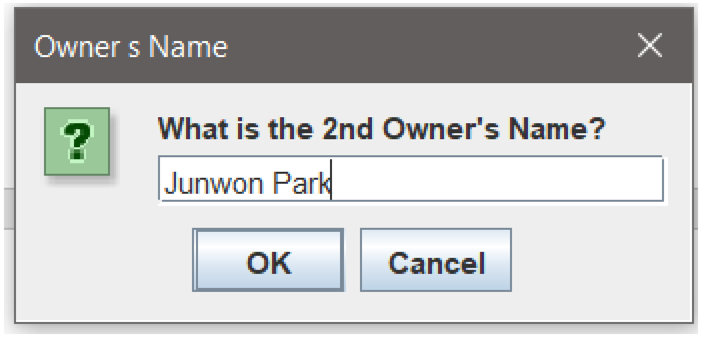

Pop-Up Message

Adding Current Listing for brandon lee



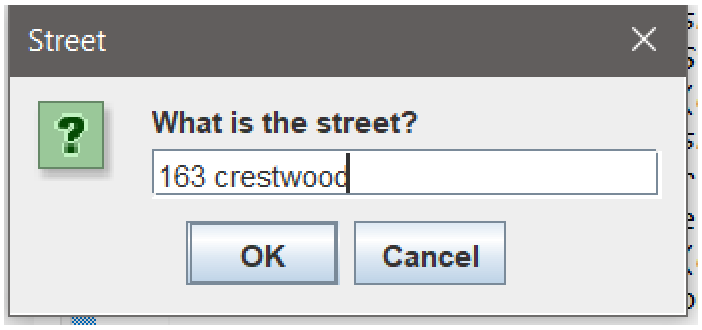
What is the 1st Owner's Name?

Paul Park



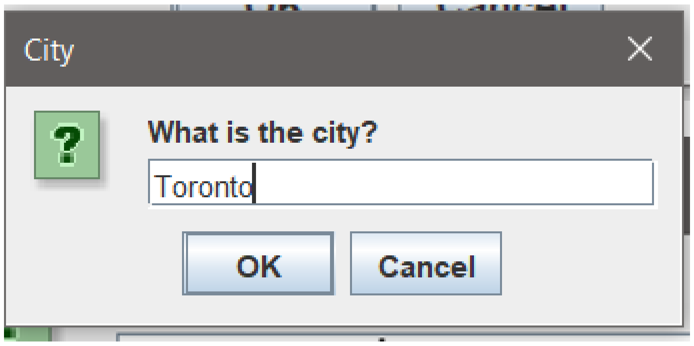
What is the 2nd Owner's Name?

Junwon Park



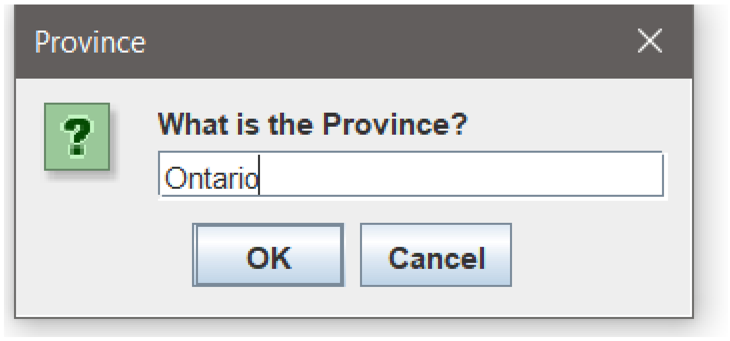
What is the street?

163 crestwood



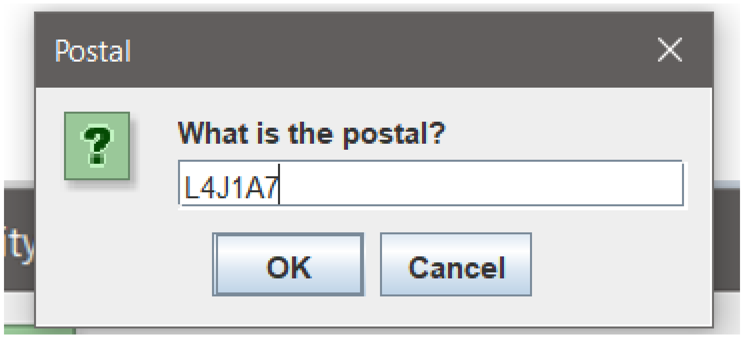
What is the city?

Toronto


Province

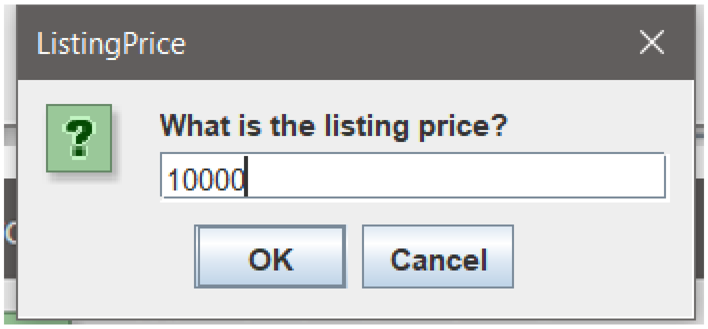
What is the Province?

Ontario


Postal

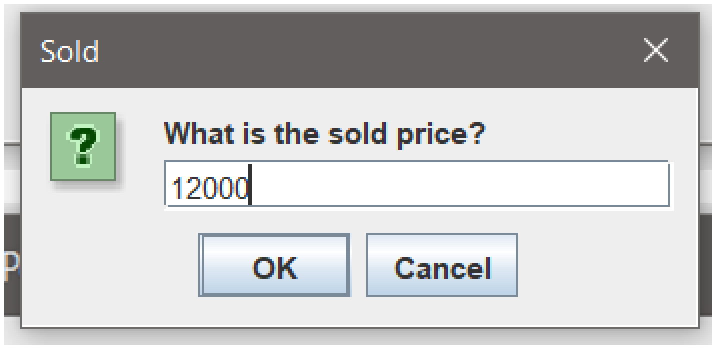
What is the postal?

L4J1A7


ListingPrice

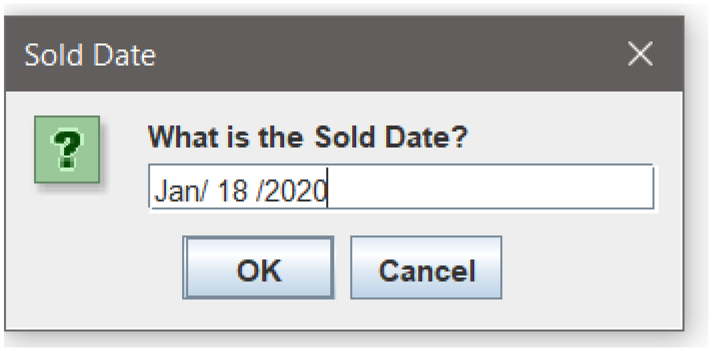
What is the listing price?

10000


Sold

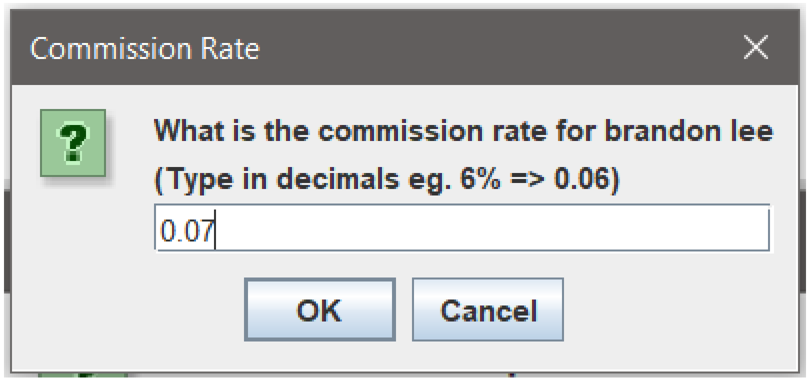
What is the sold price?

12000


Sold Date

What is the Sold Date?

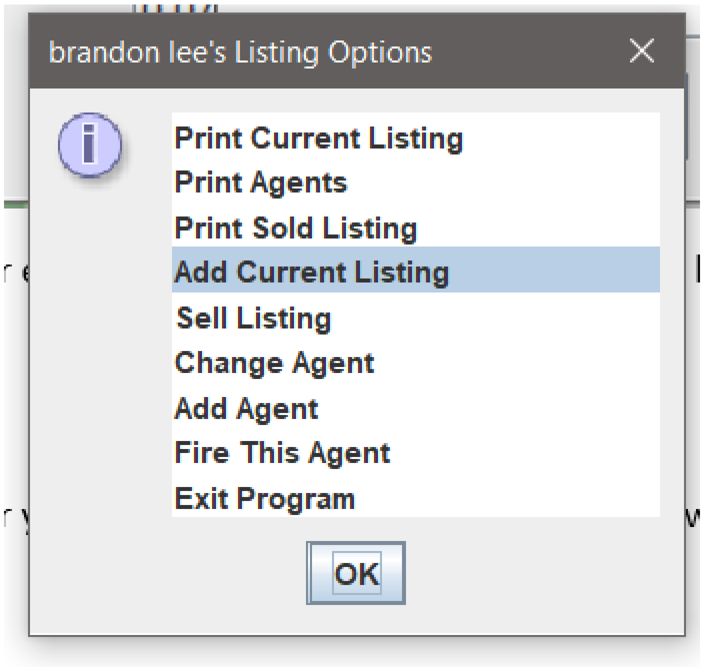
Jan/ 18 /2020


Commission Rate

What is the commission rate for brandon lee
(Type in decimals eg. 6% => 0.06)

0.07

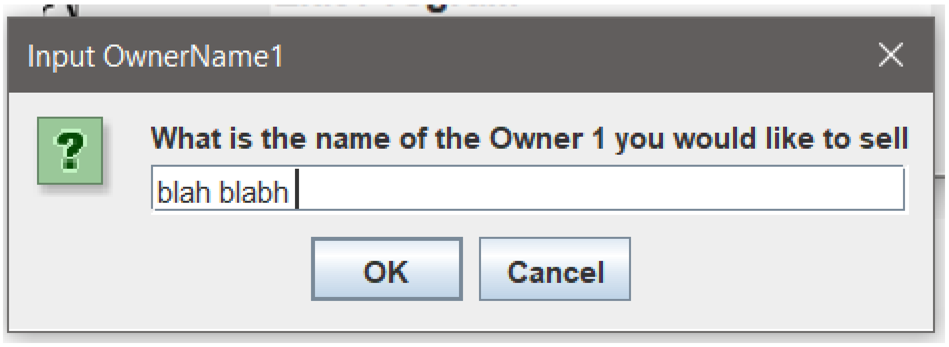
After entering the options it will go back to the listing option for the agent.


brandon lee's Listing Options

Now, the user is able to sell the listing

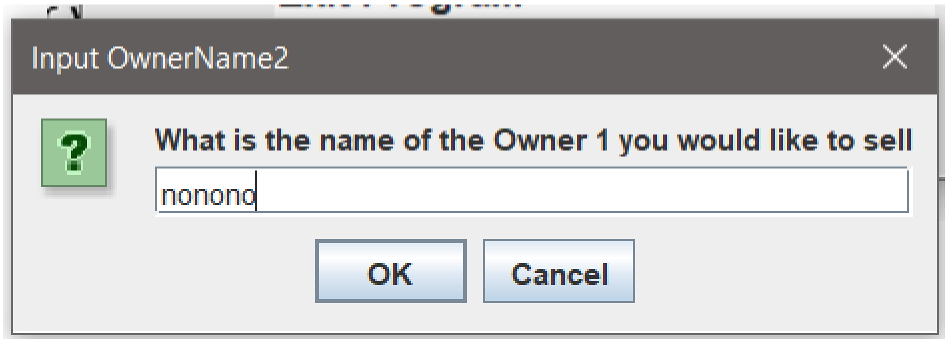
The computer asks for the owner 1 name and owner 2 name (so it does not sell duplicate owner 1 name)

HOWEVER, if the users enter the wrong name…


Input OwnerName1

What is the name of the Owner 1 you would like to sell

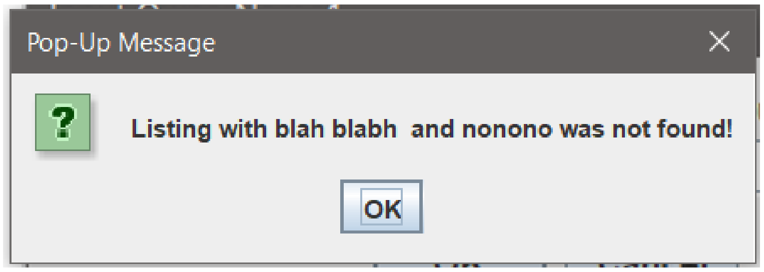
blah blabh


Input OwnerName2

What is the name of the Owner 1 you would like to sell

nonono

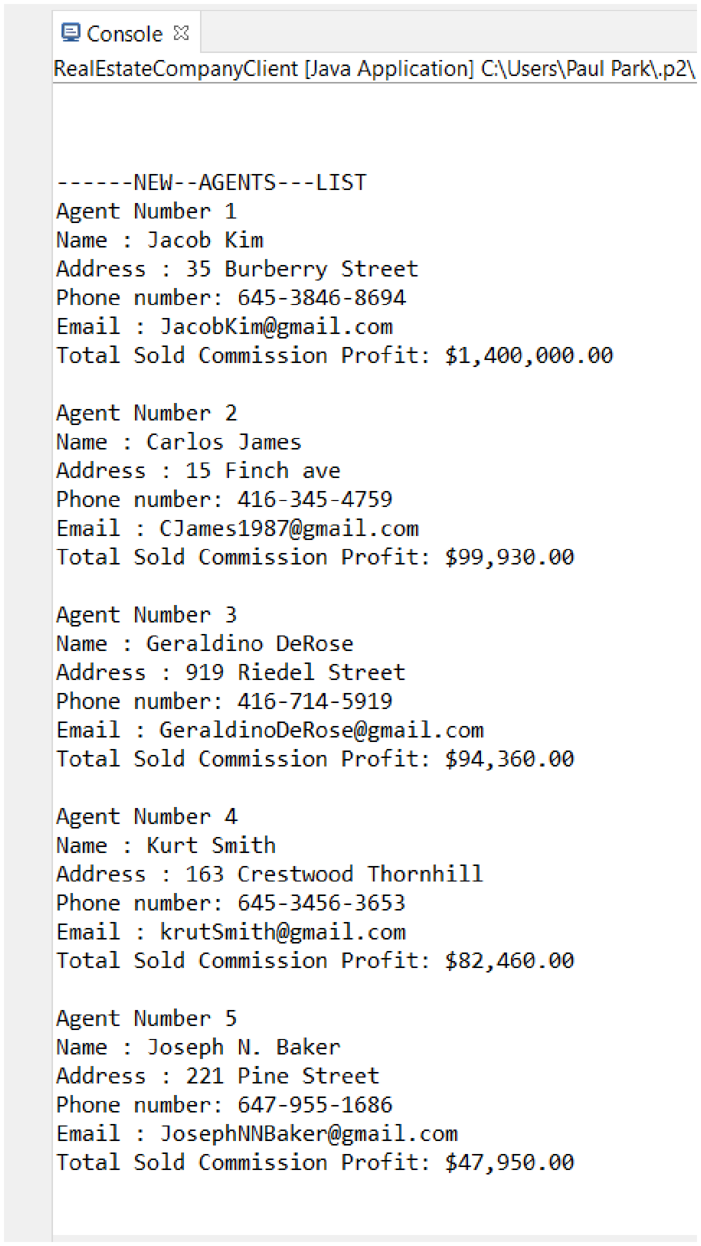
The computer recognizes it and …


Pop-Up Message

Listing with blah blabh and nonono was not found!

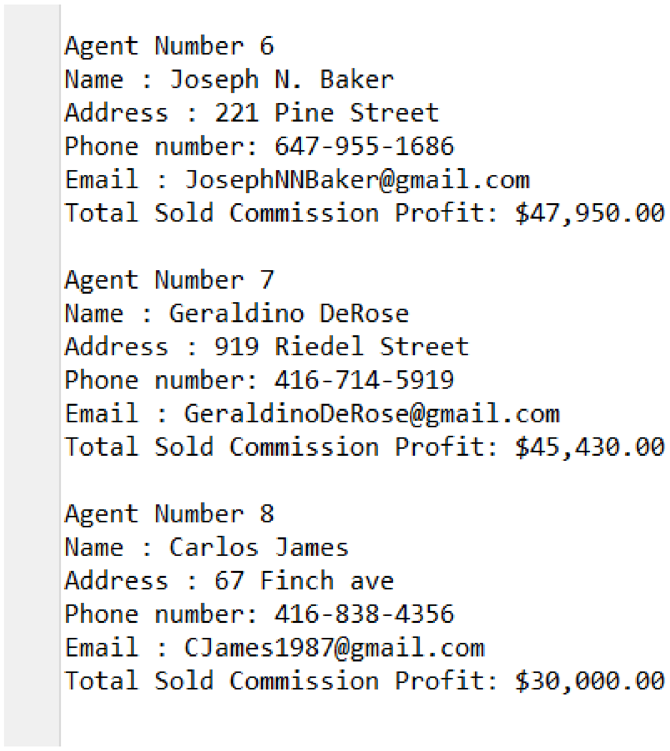
So after when the user enters the correct name the listing is sold.

Then agent “Jacob Kim” is moved to the top of the agent list because he has the most Profit made.


Console

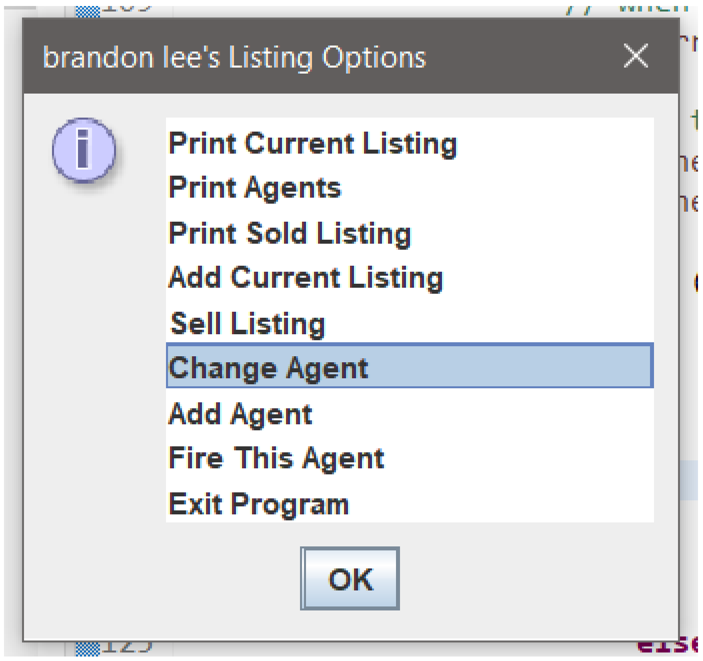
RealEstateCompanyClient [Java Application] C:\Users\Paul Park\.p2\

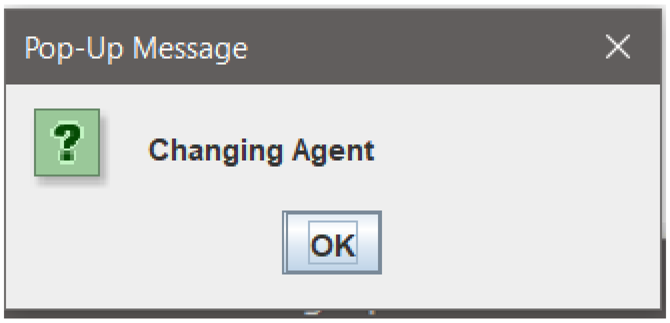
```
------NEW--AGENTS---LIST
Agent Number 1
Name : Jacob Kim
Address : 35 Burberry Street
Phone number: 645-3846-8694
Email : JacobKim@gmail.com
Total Sold Commission Profit: $1,400,000.00

Agent Number 2
Name : Carlos James
Address : 15 Finch ave
Phone number: 416-345-4759
Email : CJames1987@gmail.com
Total Sold Commission Profit: $99,930.00

Agent Number 3
Name : Geraldino DeRose
Address : 919 Riedel Street
Phone number: 416-714-5919
Email : GeraldinoDeRose@gmail.com
Total Sold Commission Profit: $94,360.00

Agent Number 4
Name : Kurt Smith
Address : 163 Crestwood Thornhill
Phone number: 645-3456-3653
Email : krutSmith@gmail.com
Total Sold Commission Profit: $82,460.00

Agent Number 5
Name : Joseph N. Baker
Address : 221 Pine Street
Phone number: 647-955-1686
Email : JosephNNBaker@gmail.com
Total Sold Commission Profit: $47,950.00
```



```
Agent Number 6
Name : Joseph N. Baker
Address : 221 Pine Street
Phone number: 647-955-1686
Email : JosephNNBaker@gmail.com
Total Sold Commission Profit: $47,950.00

Agent Number 7
Name : Geraldino DeRose
Address : 919 Riedel Street
Phone number: 416-714-5919
Email : GeraldinoDeRose@gmail.com
Total Sold Commission Profit: $45,430.00

Agent Number 8
Name : Carlos James
Address : 67 Finch ave
Phone number: 416-838-4356
Email : CJames1987@gmail.com
Total Sold Commission Profit: $30,000.00
```

User then can change the Agent

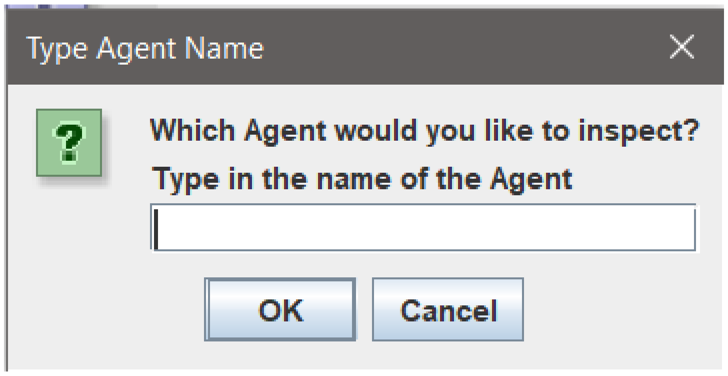



Pop-Up Message

Changing Agent

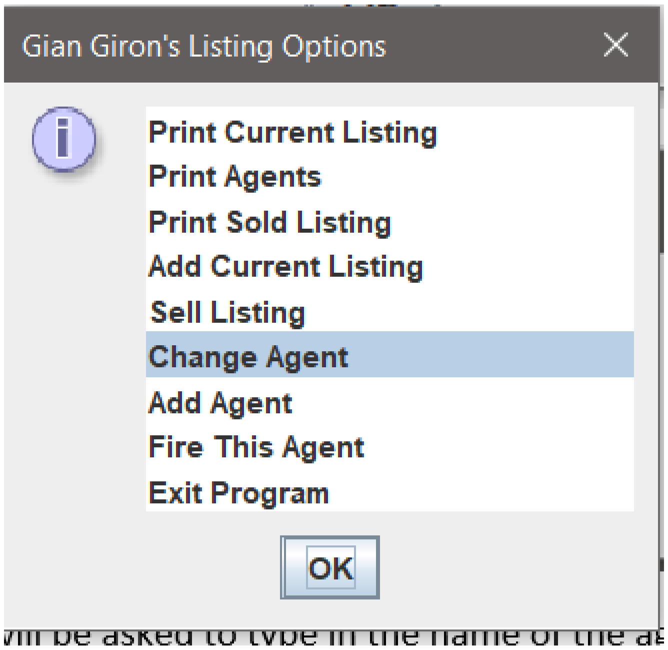
And will be asked to type in the name of the agent to search.

User can use the agent list printed before to find the list of agents.

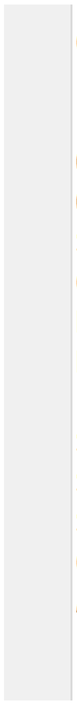

Type Agent Name

Which Agent would you like to inspect?
Type in the name of the Agent

After user types in a name, the listing option appears again.

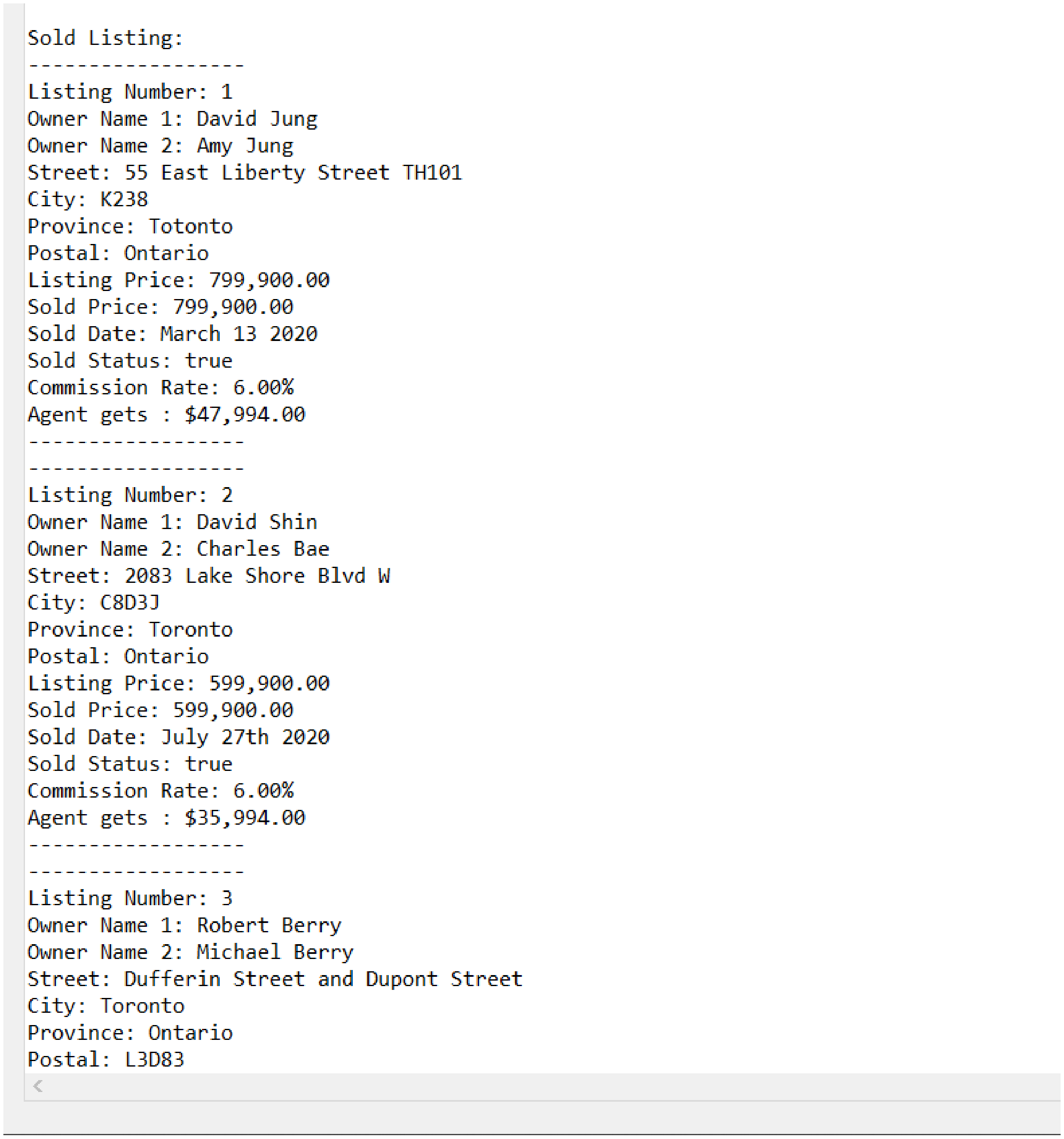

Gian Giron's Listing Options

“Print Current Listing” would print the Agent’s listing that are not sold yet, but is going to be sold.

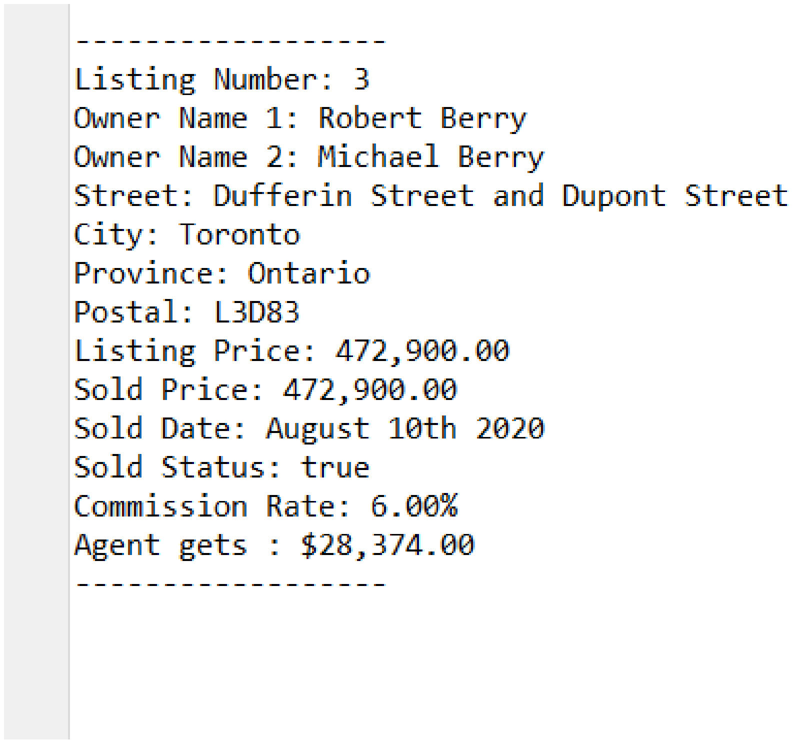


```
Current Listing:
------------------
Listing Number: 1
Owner Name 1: Moris Larsen
Owner Name 2: Frank Zackett
Street: 15 Steels and Finch
City: Toronto
Province: Ontario
Postal: CV14J
Listing Price: 20,000,000.00
Sold Price: 20,000,000.00
Sold Date: Jan 10th 2020
Sold Status: false
Commission Rate: 7.00%
Agent gets : $1,400,000.00
------------------
```

"Print Sold Listing" will print the listing the Agent sold!

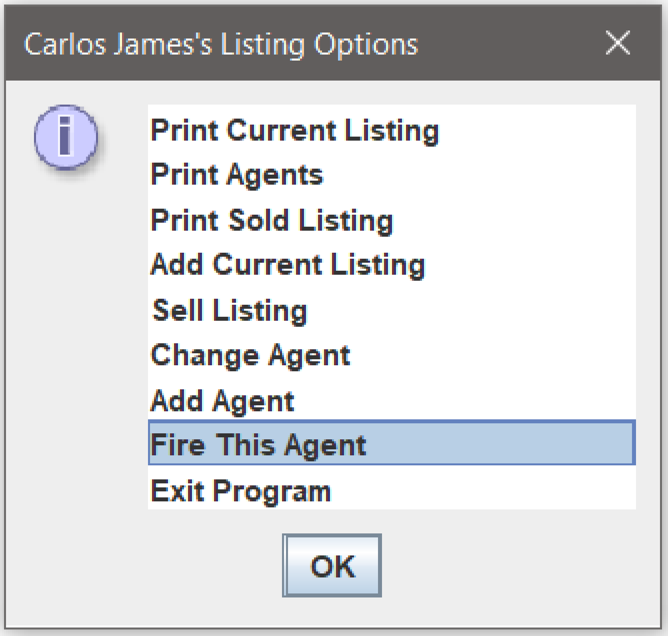


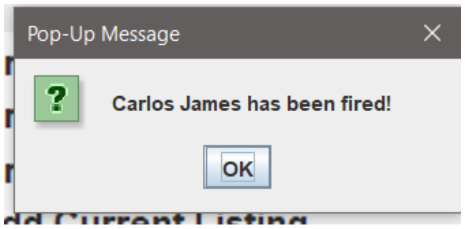
```
Sold Listing:
------------------
Listing Number: 1
Owner Name 1: David Jung
Owner Name 2: Amy Jung
Street: 55 East Liberty Street TH101
City: K238
Province: Totonto
Postal: Ontario
Listing Price: 799,900.00
Sold Price: 799,900.00
Sold Date: March 13 2020
Sold Status: true
Commission Rate: 6.00%
Agent gets : $47,994.00
------------------
------------------
Listing Number: 2
Owner Name 1: David Shin
Owner Name 2: Charles Bae
Street: 2083 Lake Shore Blvd W
City: C8D3J
Province: Toronto
Postal: Ontario
Listing Price: 599,900.00
Sold Price: 599,900.00
Sold Date: July 27th 2020
Sold Status: true
Commission Rate: 6.00%
Agent gets : $35,994.00
------------------
------------------
Listing Number: 3
Owner Name 1: Robert Berry
Owner Name 2: Michael Berry
Street: Dufferin Street and Dupont Street
City: Toronto
Province: Ontario
Postal: L3D83
```



```
------------------
Listing Number: 3
Owner Name 1: Robert Berry
Owner Name 2: Michael Berry
Street: Dufferin Street and Dupont Street
City: Toronto
Province: Ontario
Postal: L3D83
Listing Price: 472,900.00
Sold Price: 472,900.00
Sold Date: August 10th 2020
Sold Status: true
Commission Rate: 6.00%
Agent gets : $28,374.00
------------------
```

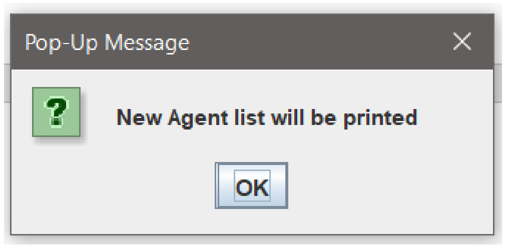
User can also fire and Agent.




Pop-Up Message

Carlos James has been fired!

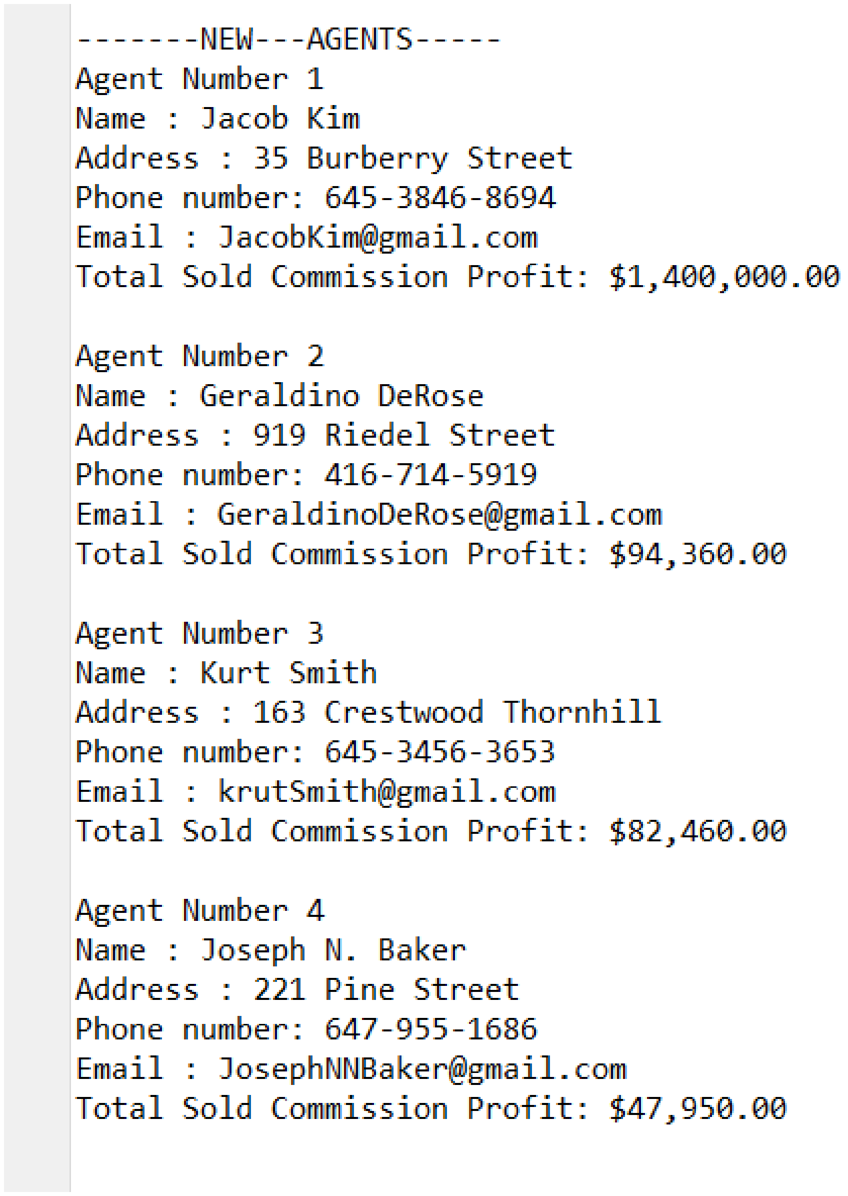
OK


Pop-Up Message

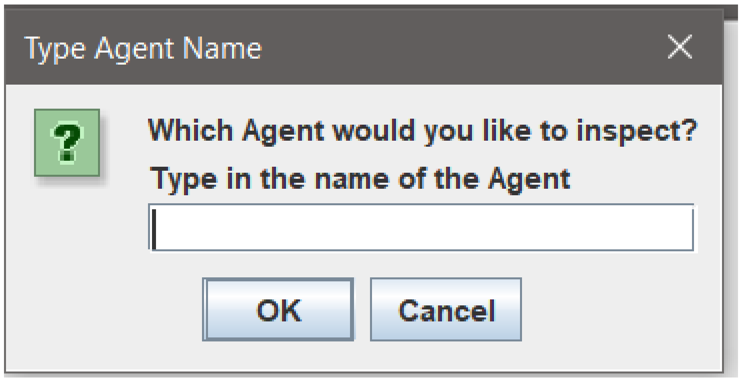
New Agent list will be printed

OK

The new List without the fired agent is printed



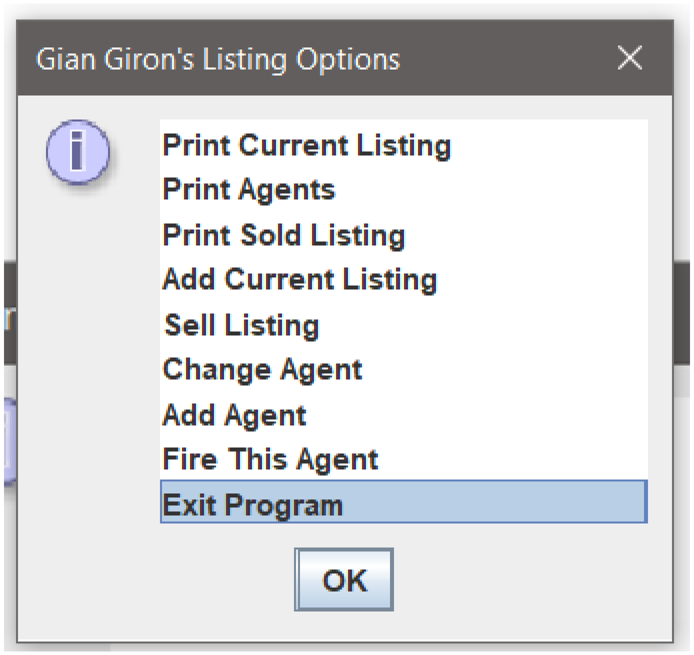
```
-------NEW---AGENTS-----
Agent Number 1
Name : Jacob Kim
Address : 35 Burberry Street
Phone number: 645-3846-8694
Email : JacobKim@gmail.com
Total Sold Commission Profit: $1,400,000.00

Agent Number 2
Name : Geraldino DeRose
Address : 919 Riedel Street
Phone number: 416-714-5919
Email : GeraldinoDeRose@gmail.com
Total Sold Commission Profit: $94,360.00

Agent Number 3
Name : Kurt Smith
Address : 163 Crestwood Thornhill
Phone number: 645-3456-3653
Email : krutSmith@gmail.com
Total Sold Commission Profit: $82,460.00

Agent Number 4
Name : Joseph N. Baker
Address : 221 Pine Street
Phone number: 647-955-1686
Email : JosephNNBaker@gmail.com
Total Sold Commission Profit: $47,950.00
```

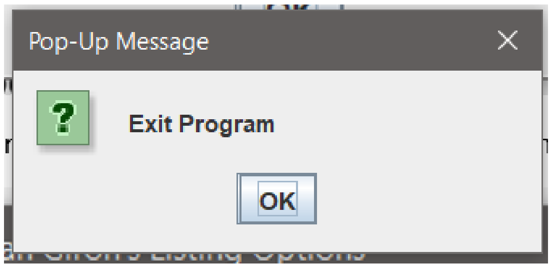


Type Agent Name

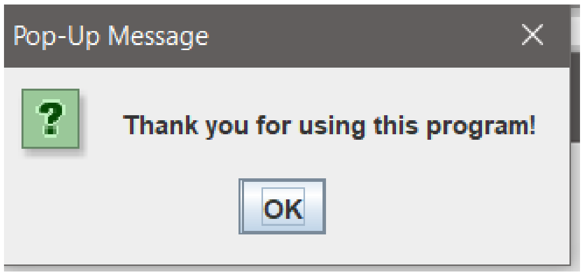
Which Agent would you like to inspect?
Type in the name of the Agent

If user wants to leave the program the program will exit.


Gian Giron's Listing Options


Pop-Up Message

Exit Program


Pop-Up Message

Thank you for using this program!