Centre for Information and Communication Technology

|  |  |
| --- | --- |
| Assignment 2: | Real Estate Application using Arrays |

# Due Date

This assignment uses the skills and knowledge you have learnt in Java 1 up to and including Arrays. It is a simulation of a work task.

The completed application is due 21 May 2017 at 11pm by submitting in eLearn. Face-to-face students should also demonstrate it to teachers in class. The resubmit, if required, is due 28 May 2017 at 11pm.

Please note that procedures in the IT Department at CIT require submission on time unless a doctors certificate is provided, otherwise you will need to resubmit having lost your first chance.

# Assessment Task Description

This assignment is a work simulation task. You work for an IT company that has been engaged by a local real estate agency to develop a GUI application to list properties that they have for sale and those that they have sold. Your boss has met with the CEO of the Real Estate Company to determine requirements and he has done analysis on the project and come up with a design for the solution.

Your boss has got one programmer working on the user interface and the task she has given to you is to code two classes matching the URL diagrams she provides you. The classes that she wants you to create are a Property class (which will store information such as the id, address, asking price, whether the property has been sold & the sold price) and a RealEstateCompany class. The GUI created by the other employee will use these two classes that you create.

A test script is provided by your boss in a class called “TestRealEstateCompany”. To run the test script you right click on the class in BlueJ and choose to run the void main method. You should check the output that the script produces against the output that she has provided as “expected results”. Your boss is happy for the formatting of your output to be a little different as long as all the information is there and is correct.

You are required to code the two classes (Property & RealEstateCompany) in a BlueJ project. Also include the TestRealEstateCompany class with no changes made to the code except to change the name of the company from Fred Jones Realty to your name. After successfully testing you should zip up the whole folder in which your BlueJ project is stored and submit it on eLearn.

Some requirements to successfully pass the assignment are:

1. Classes match class diagrams & all attributes are declared private
2. Code is well structured, uses camel case naming standards & is commented including javadoc comments (the /\*\* comments generated by default at the top of each class where you can enter your name and the date)
3. Put your name as the name of the real estate company in the test script and ensure that it works and produces the expected results

The class diagrams that you are provided with, for the development of the Property & RealEstateCompany classes, are:

|  |
| --- |
| Property |
| int id  String address  boolean sold  int askingPrice  int soldPrice |
| Property(int inId, String inAddress, int inPrice)  Property(int inId, String inAddress)  int getId()  String getAddress()  boolean isSold()  int getAskingPrice()  int getSoldPrice()  void printDetails()  void setAskingPrice(int inAskingPrice)  void sell(int inSoldPrice) |

To enable your IT company to sell the software to other Real Estate companies your boss decided to store the name of the company as a String attribute in the RealEstateCompany class rather than hard-coding it.

|  |
| --- |
| RealEstateCompany |
| String name  int numProperties  final int MAX\_PROPERTIES = 10  Property[] properties |
| RealEstateCompany(String inName)  void addProperty(Property inProperty)  void printAllPropertiesForSale()  void printAllPropertiesSold()  void sellProperty(int inId, int inSoldPrice)  void printExpensivePropertiesForSale(double threshhold) |

**The test class that your boss provides to you is:**

public class TestRealEstateCompany

{

public static void main (String[] args)

{

RealEstateCompany re = new RealEstateCompany( "Fred Jones Realty"); //Change to your name

Property p1 = new Property(1, "1 Cheap St", 1000);

Property p2 = new Property(22, "2 Expensive Place", 2222222);

Property p3 = new Property(33, "3 Average St"); //use constructor that sets price to zero

p3.setAskingPrice(330000); //set the asking price now

Property p4 = new Property(44, "4 Average St", 340000);

Property p5 = new Property(555, "5 Average St", 350000);

re.addProperty (p1);

re.addProperty (p2);

re.addProperty (p3);

re.addProperty (p4);

re.addProperty (p5);

// properties 44 & 22 sell, number 22 at a reduced price

re.sellProperty(44, 340000);

re.sellProperty(22, 2222000);

re.printAllPropertiesForsale(); //print a list of all properties for sale

re.printAllPropertiesSold(); //print a list of all properties sold

re.printExpensivePropertiesForSale(300000); //print properties for sale worth more than $300000

re.sellProperty(5, 2000); //produces id error

re.sellProperty(44, 1000); //produces already sold error

}

}

To run the test right click on the TestCompany class and choose public void main and then just click OK when it offers you the opportunity to supply parameters. The output should be something like the following (you should check to make sure you get similar results and if you don’t then work out why & fix before submitting):

**The output that your boss expects the running of the test class is something like the following:**

\*\*\* Listing of all properties for sale

1 Cheap St at $1000

3 Average St at $330000

5 Average St at $350000

\*\*\* End of properties for sale listing

\*\*\* Listing of all properties sold

4 Average St at $340000

2 Expensive St at $2222000

\*\*\* Total price of all sales is $2562000

\*\*\* Listing of properties for sale >= $300000

3 Average St at $330000

5 Average St at $350000

\*\*\* End of properties for sale >= $300000 listing

Error – No property with id of 5 to sell

Error – Property with id 44 is already sold

# Hints

1. A property can be created with the constructor that sets the initial asking price or a constructor that sets the asking price to zero before being correctly set with the setAskingPrice() method. The boolean called sold is initially set to false & the soldPrice is set to zero.
2. The property class has a get method for a boolean attribute called “sold”. It can be called isSold() or getSold() – your choice.
3. The sell(int inSoldPrice) should set the sold attribute to true & set the soldPrice to the input parameter
4. The RealEstateCompany has an array that holds all Properties that it has dealt with in the current year – both properties for sale and those it has sold. It has an attribute to keep track of how many properties are in the array.
5. The printAllPropertiesForSale() method should print a heading & then loop through the properties array & print out the details of each property in it that has a sold attribute value of false & lastly print a report footer
6. The printAllPropertiesSold() method should print a heading & then loop through the properties array & print out the details of each property in it that has a sold attribute value of true. It should also total up the soldPrice of each of those properties to give a total sales figure
7. The sellProperty(int inId, int inSoldPrice) method needs to find the property that has an id matching inId in the properties array, and then call the sell method of the property. It should print out an error if there is no property with that id & a different error printed if the property is found but its sold attribute indicates that it is already sold. See the following NS chart
8. printExpensivePropertiesForSale(double threshold) should print a heading & then the details of any property that is for sale with a price greater than or equal to the threshold amount before printing a report footer.

**NS chart for the sellProperty(int inId, int insoldPrice) method:**

propertyFound = false

for (int i=0; i<numProperties; i++)

if properties[i].getId() == inId

T F

if properties[i].isSold()

T F

Error – property properties[i].sell(inPrice)

already sold propertyFound= true

if propertyFound

T F

Error property not found

# Submission Format

You are required to code the two classes (Property & RealEstateCompany) in a BlueJ project. Also include the TestRealEstateCompany class with no changes made to the code except to change the name of the company from Fred Jones Realty to your name. After successfully testing you should zip up the whole folder in which your BlueJ project is stored and submit it on eLearn.

# Requirements

Some requirements to successfully pass the assignment are:

1. Classes match class diagrams & all attributes are declared private
2. Code is well structured, uses camel case naming standards & is commented including javadoc comments (the /\*\* comments generated by default at the top of each class where you can enter your name and the date)
3. Put your name as the name of the real estate company in the test script and ensure that it works and produces the expected results

# Assessment

Your assignment will be assessed against the following performance, knowledge & evidence criteria & all must be passed. The corresponding requirement is shown in brackets after each one.

**ICTPRG406**

1.1 Apply basic language syntax rules and best practices

1.2 Select and use language data types, operators and expressions, in order to create clear and concise code

1.3 Use the appropriate language syntax for sequence, selection and iteration constructs

1.4 Use a modular programming approach within member or function logic

1.5 Apply arrays, including arrays of objects to introductory programming tasks

1.6 Use standard-array processing algorithms

2.1 Implement a class that contains primitive member or instance variables

2.2 Implement a class that contains multiple options for object construction

2.3 Implement a class that uses user-defined aggregation (object instance or member variables)

4.1 Follow organisational guidelines for developing maintainable code, and adhere to the provided coding standards, when documenting activities

4.2 Apply internal documentation to all the code created, and use the documentation tools available in the target language, when documenting activities

6.1 Develop a solution, when provided with a basic object-oriented design document

P1 - use an application program to design, and build, standard reusable software modules in response to a design specification

P2 - generate the code, and the documentation

P3 - undertake testing and debugging, to meet specifications.