

Programming Project 4

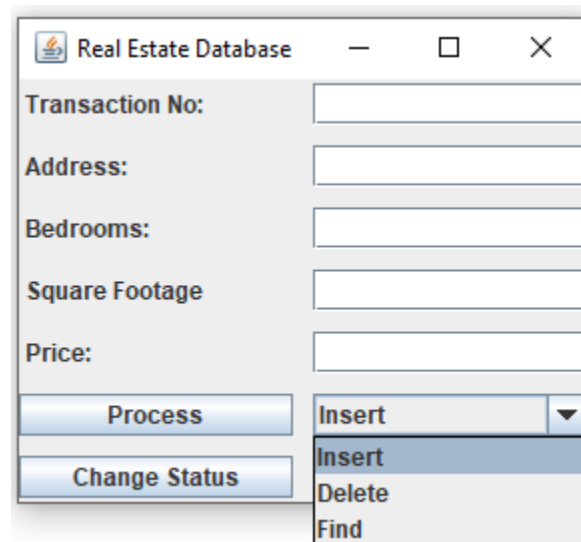
The fourth programming project involves writing a program to manage a real estate database. This program should be comprised of an enumerated type, an interface and two classes. The enumerated type should be named `Status` and should contain three enumeration literals, `FOR_SALE`, `UNDER_CONTRACT` and `SOLD`.

The interface should be a generic interface named `ChangeableState` and it should have a bounded generic type parameter whose type must be an enumerated type. It should contain one abstract method `changeState` that has a parameter whose type of the generic type parameter.

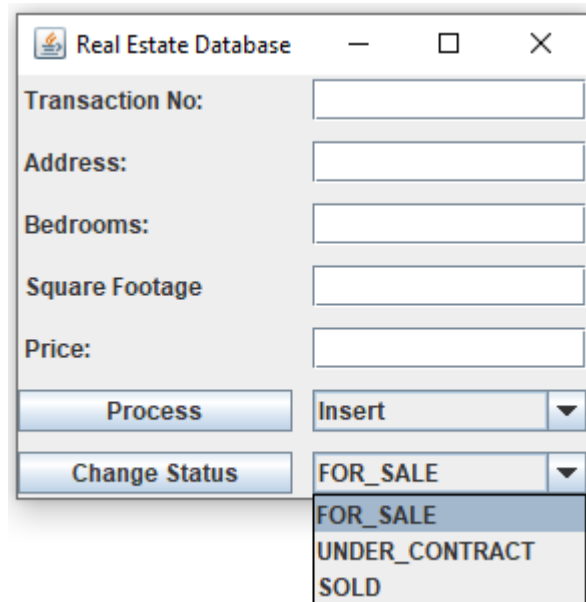
The first of the two classes should be named `Property`. It should implement the `ChangeableState` interface. It should contain five instance variables, the property address stored as a string, the number of bedrooms, the square footage and the price, all stored as integers, and the status of the property whose type should be the enumerated type `Status`. In addition, it should have the following three methods:

1. A constructor that accepts four parameters for the purpose of initializing the characteristics of the property, the address, the number of bedrooms, the square footage and the price. The status of the property should be set to `FOR_SALE`.
2. A method named `changeState` that allows the status of the property to be changed.
3. An overridden `toString` method that returns a string containing the property address, the number of bedrooms, the square footage, the price and the current status, appropriately labeled.

The second class named `Project4` should contain the main method. In addition, it should contain an instance variable that defines the database of property records, which is implemented as a `TreeMap`, with the transaction number field as the key and a `Property` object as the value. It should generate the GUI shown below:



Clicking the *Process* button should cause the selected choice of the three database actions in the combo box to its right to be executed. It should first check whether any non integer values have been entered in any of the fields that require integers. If so, an error message should be displayed in a `JOptionPane` window. The operation should be performed when the user clicks the *Process* button. If the user attempts to insert a key that is already in the database an error message should be displayed using a `JOptionPane` message dialog box. If the user attempts to delete or find a record that is not in the database, a message should also be displayed. After each successful operation is completed a `JOptionPane` window should be displayed confirming the success. In the case of a successful *Find* request, a window should pop up containing all the information in the associated `Property` object.



Transaction No:	<input type="text"/>
Address:	<input type="text"/>
Bedrooms:	<input type="text"/>
Square Footage	<input type="text"/>
Price:	<input type="text"/>
Process	Insert ▼
Change Status	FOR_SALE ▼
	FOR_SALE
	UNDER_CONTRACT
	SOLD

Clicking the *Change Status* button should cause status of the property association with the designated transaction number to be changed to status selected in the combo box to its right.

Be sure to follow good programming style, which means making all instance variables private, naming all constants and avoiding the duplication of code. Furthermore you must select enough different kinds of shapes to completely test the program.