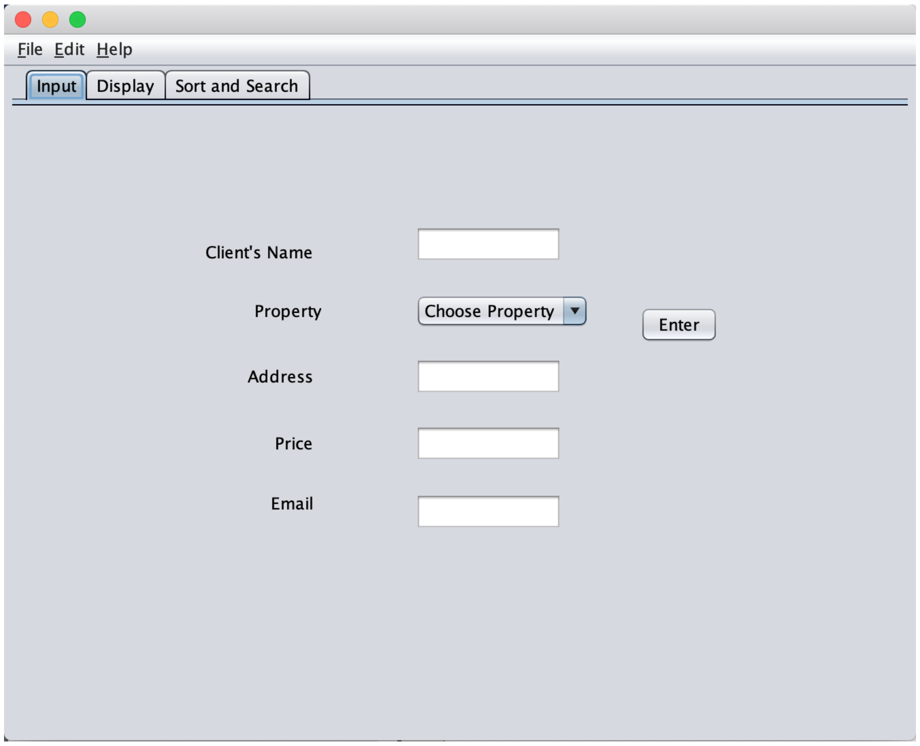
Introduction
The program I coded is called “real estate project”. This program serves the purpose of allowing clients to organize and search for data associated with each real estate client. Its function includes allowing users to enter information, a data table where the information is displayed, and being able to search up the data that’s entered. This solves the problem of having to scramble through different files and documents to find the information.

Summary of Programming Techniques
- Parameter passing
- For loop
- While loop
- Recursion
- Returning strings and int
- Getting strings and int
- Arrays
- Object-oriented programming class
- Private methods
- If/else statements
- Error handling
- GUI tabs
- GUI popup menus
- Search and sort

Program Structure
The real estate project program allows the user to enter data and search through the data that’s been stored in the program. The program was coded and run using an IDE with GUI features. The user interface allows the user to see the data that’s been entered and other information that would allow the user to run the program. The data that’s entered will be stored in the RAM, therefore, it is volatile which means that if there’s no internet, the data will not be accessible. The program consists of one class and the main GUI which allows for data to be set and retrieved. The main GUI allows for the user interface.

User Interface/ GUI Work



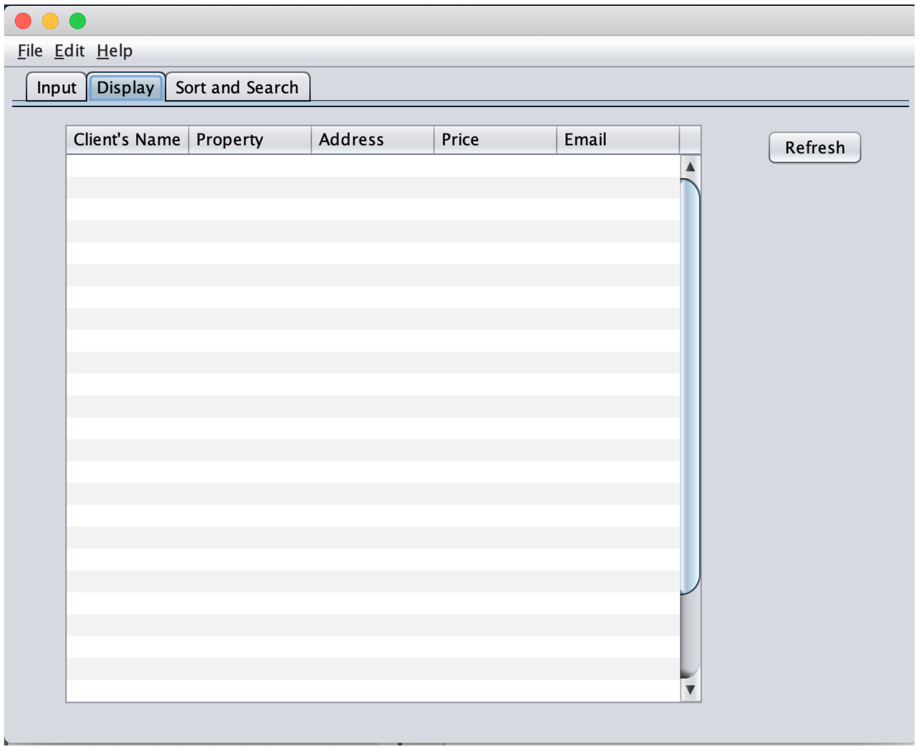
Client's Name

Property

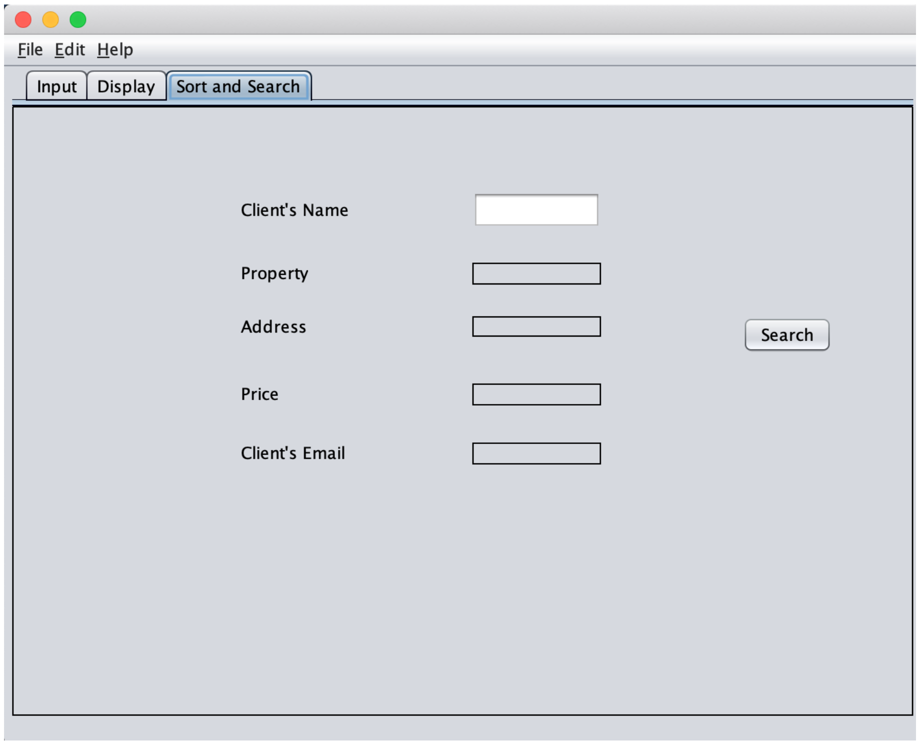
Address

Price

Email



Client's Name	Property	Address	Price	Email



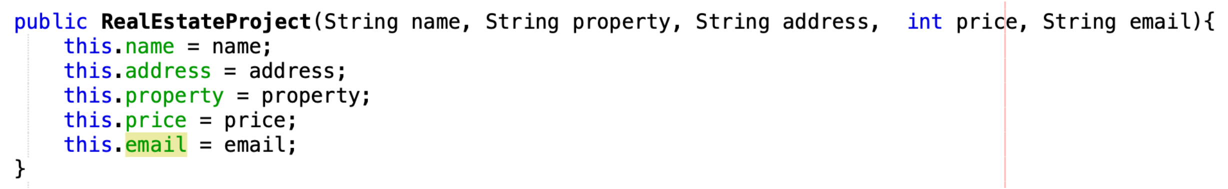
GUI components such as JLabels, JTextAreas, and JComboBoxes were used in this program to create an interactive user interface. JLabels and JTables were used to organize user inputs. The “Enter” button is used to get user input while the “refresh” button allows for users to reenter input and the “Search” button is used to allow users to search through the information entered in the program.

Software Used

- NetBeans

The software used for this program is NetBeans to allow for the user interface. NetBeans is used to allow for user input and allow for those inputs to be stored and searched through.

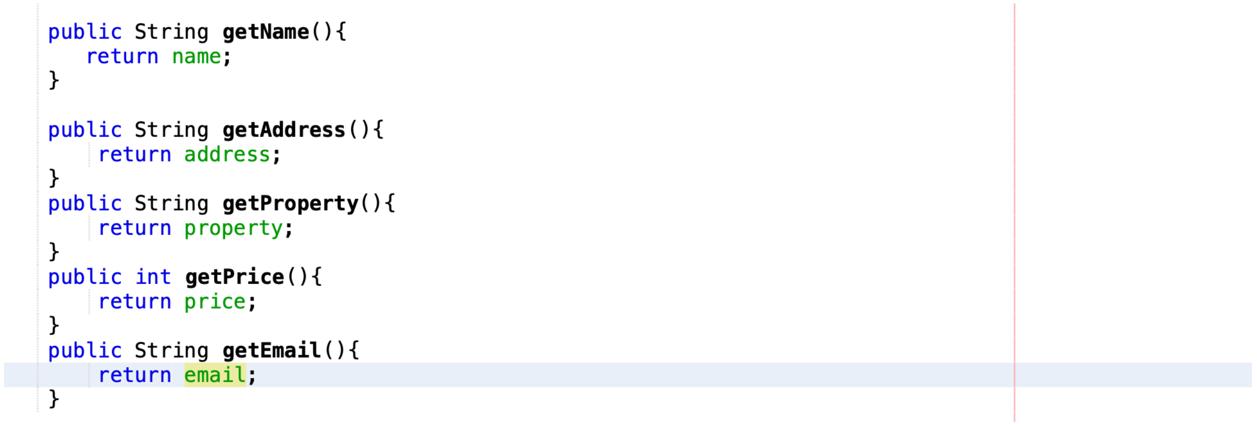
Main Unique Algorithms



```
public RealEstateProject(String name, String property, String address,  int price, String email){
    this.name = name;
    this.address = address;
    this.property = property;
    this.price = price;
    this.email = email;
}
```

Figure 1: Screenshot taken by author

This part of the code is taken from the RealEstateProject class. First I declared that the RealEstateProject would have 5 components being the String name, String property, String address, integer price, String email.

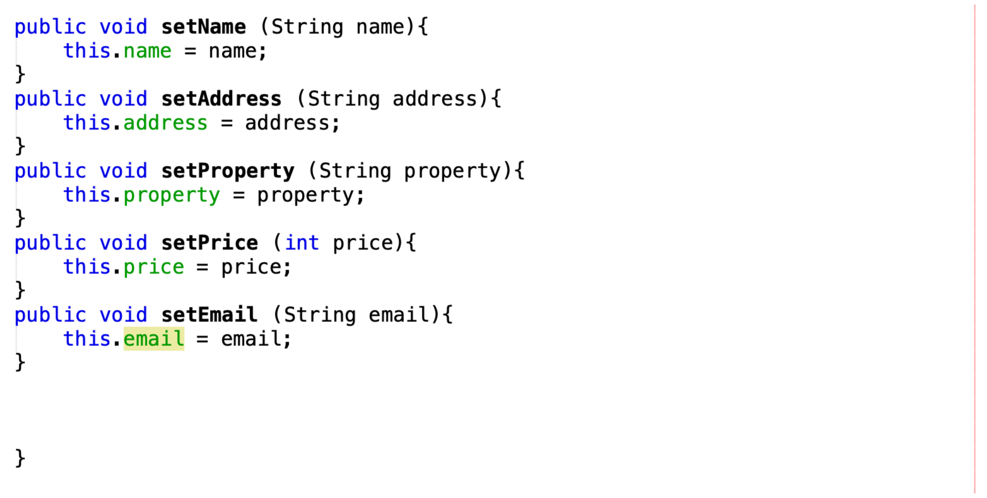


```
public String getName(){
   return name;
}

public String getAddress(){
    return address;
}
public String getProperty(){
    return property;
}
public int getPrice(){
    return price;
}
public String getEmail(){
    return email;
}
```

Figure 2: Screenshot taken by author

The next part, allows the Strings and int to be "get" meaning that by calling the method getName, getAddress, etc, the code will return the String or integer associated with it.



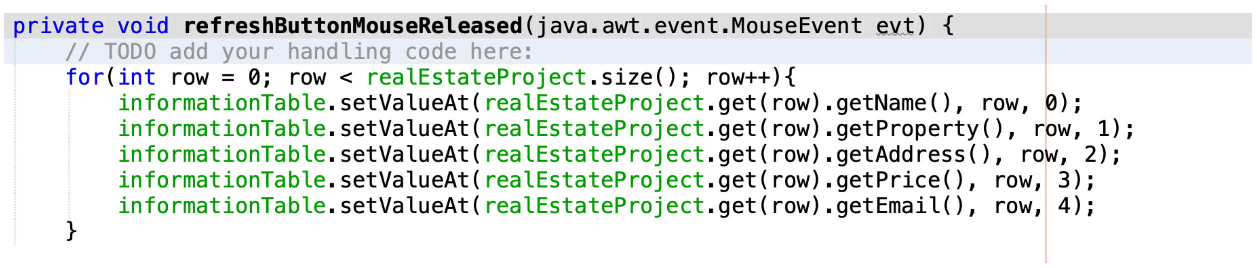
```
public void setName (String name){
    this.name = name;
}
public void setAddress (String address){
    this.address = address;
}
public void setProperty (String property){
    this.property = property;
}
public void setPrice (int price){
    this.price = price;
}
public void setEmail (String email){
    this.email = email;
}



}
```

Figure 3: Screenshot taken by author

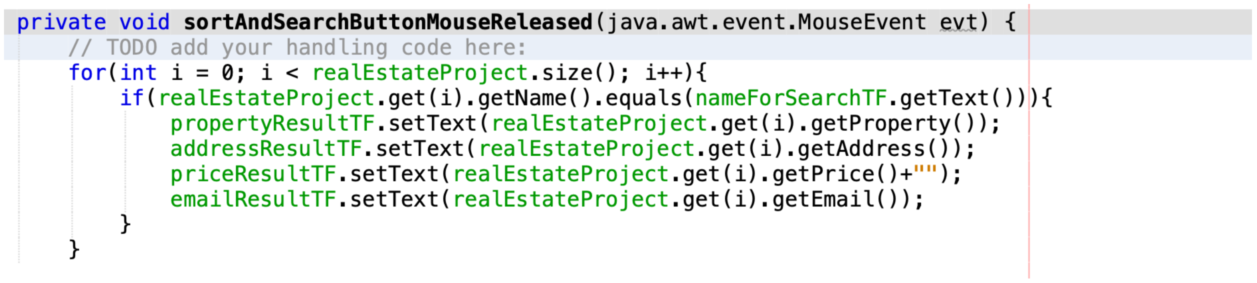
In this part of the code, I'm enabling the Strings of name, address, property and email as well as the int of price to be set in by the user input.



```java
private void refreshButtonMouseReleased(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    for(int row = 0; row < realEstateProject.size(); row++){
        informationTable.setValueAt(realEstateProject.get(row).getName(), row, 0);
        informationTable.setValueAt(realEstateProject.get(row).getProperty(), row, 1);
        informationTable.setValueAt(realEstateProject.get(row).getAddress(), row, 2);
        informationTable.setValueAt(realEstateProject.get(row).getPrice(), row, 3);
        informationTable.setValueAt(realEstateProject.get(row).getEmail(), row, 4);
    }
```

Figure 4: Screenshot taken by author

This section of the code refers to the refresh button associated with the Display tab. At the event of the release of the button, the code will run through the for loop. The different values and Strings will be set as can be seen from ".setValueAt". Then the code gets the variable (name, property, address, price, and email) and the row it corresponds with.



```java
private void sortAndSearchButtonMouseReleased(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    for(int i = 0; i < realEstateProject.size(); i++){
        if(realEstateProject.get(i).getName().equals(nameForSearchTF.getText())){
            propertyResultTF.setText(realEstateProject.get(i).getProperty());
            addressResultTF.setText(realEstateProject.get(i).getAddress());
            priceResultTF.setText(realEstateProject.get(i).getPrice()+"");
            emailResultTF.setText(realEstateProject.get(i).getEmail());
        }
    }
```

Figure 5: Screenshot taken by author

This section of the code refers to the search button in the sort and search tab. At the release of the button, the code will run through the for loop then the if statement. The code will run through the if statement as many times as the size of the realEstateProject list. Then if the if statement is true, it'll enter the following part of the code where the it'll set the textField in the panel according to the variables associated with it.