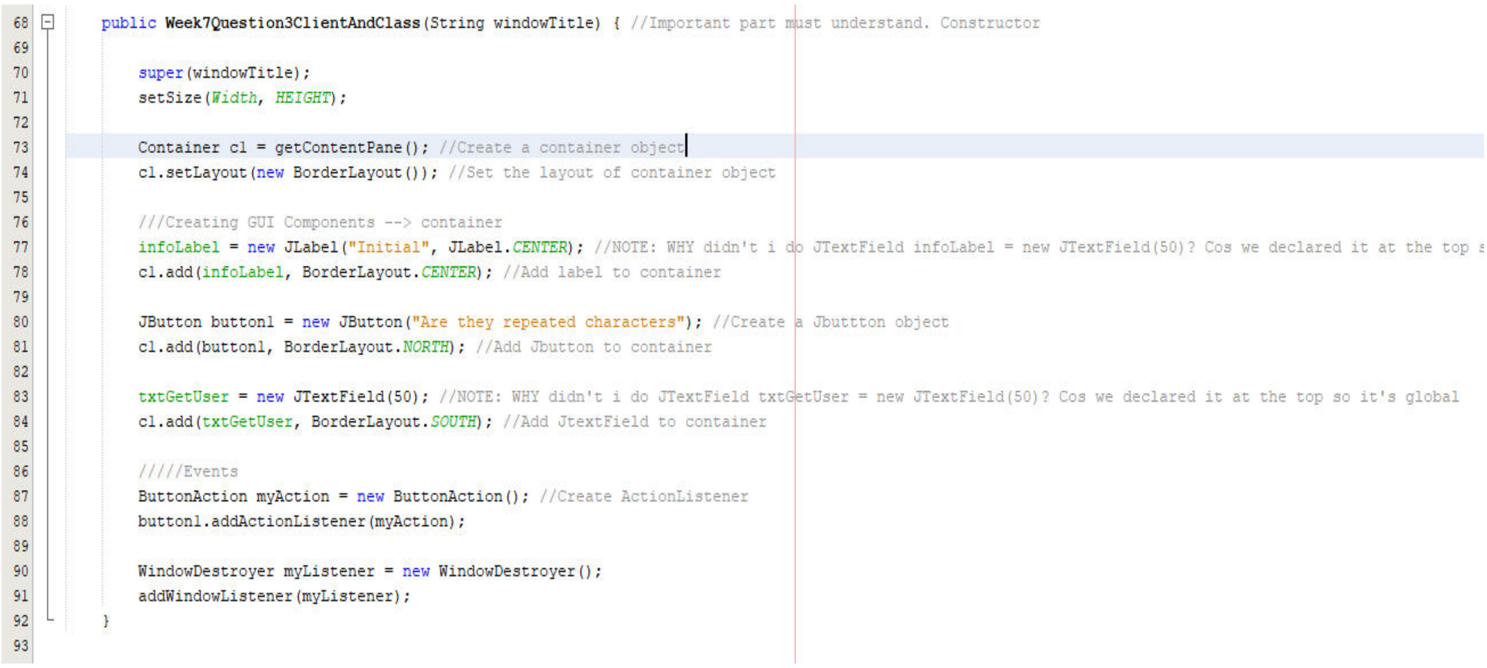
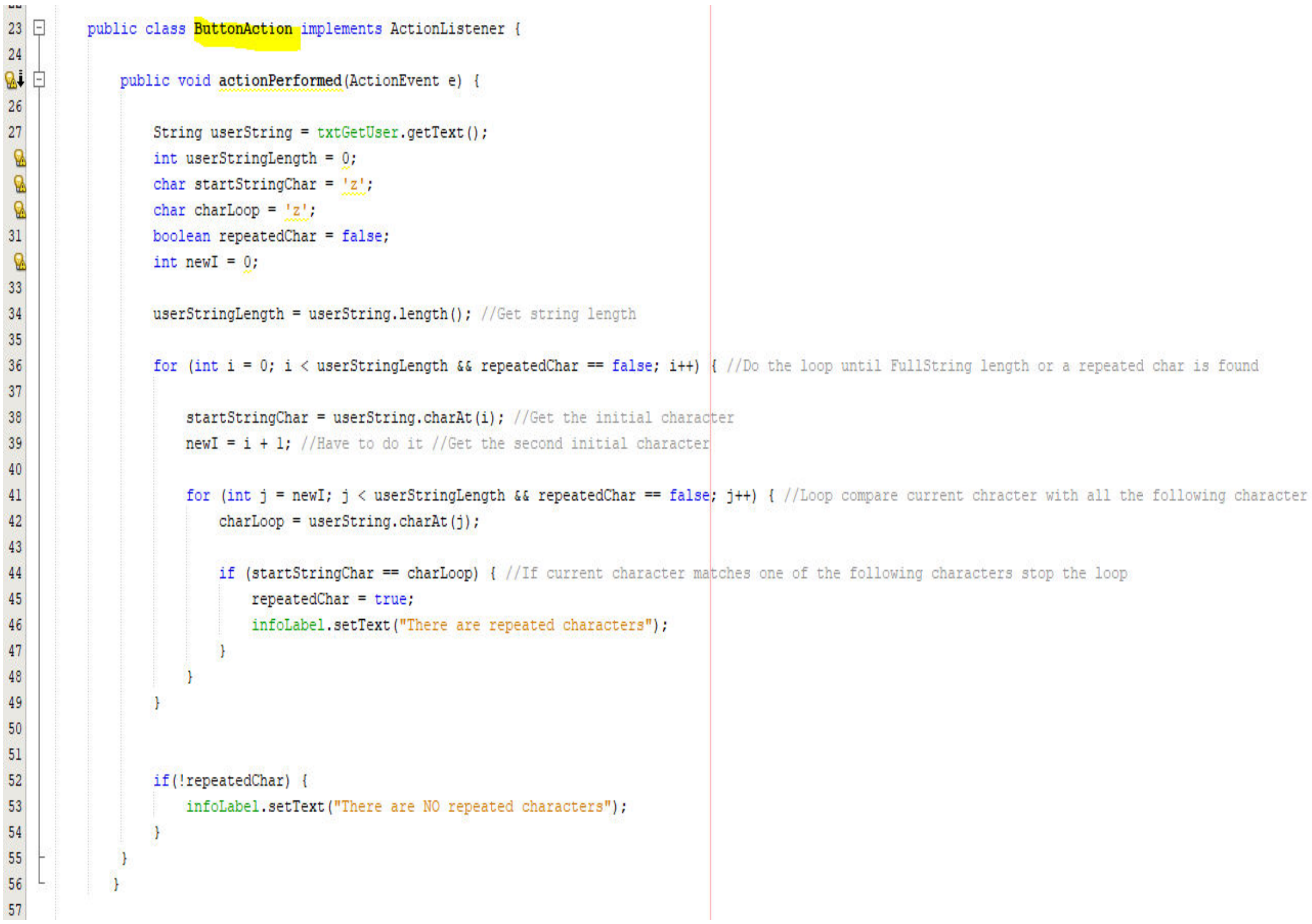
3) Create a GUI with a button, text field, and label inside container+ Determine if there are repeated characters in string

Create GUI-

```
public class Week7Question3ClientAndClass extends JFrame {


}
```



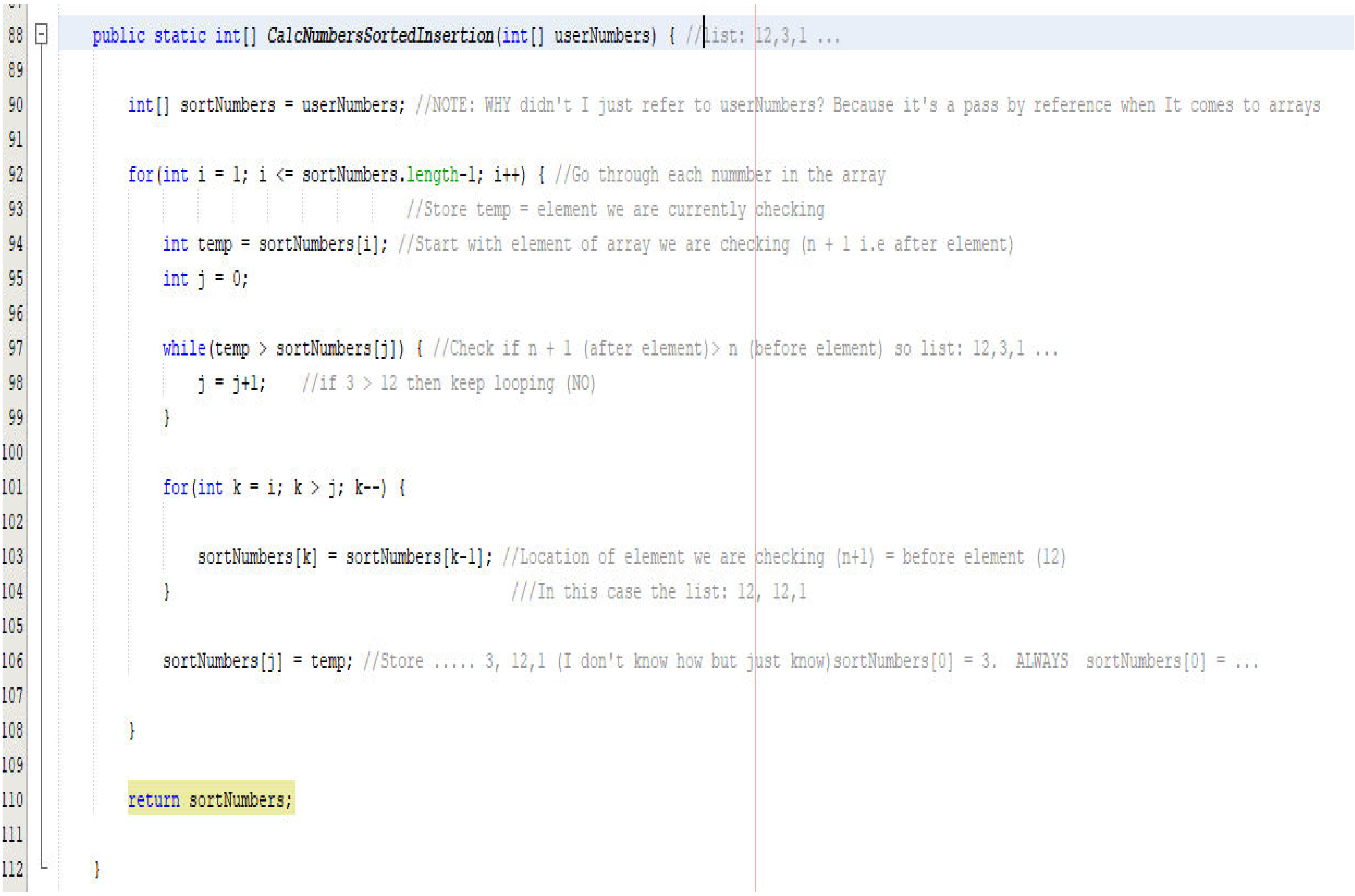
```
public Week7Question3ClientAndClass(String windowTitle) { //Important part must understand. Constructor

    super(windowTitle);
    setSize(Width, HEIGHT);

    Container cl = getContentPane(); //Create a container object
    cl.setLayout(new BorderLayout()); //Set the layout of container object

    ///Creating GUI Components --> container
    infoLabel = new JLabel("Initial", JLabel.CENTER); //NOTE: WHY didn't i do JTextField infoLabel = new JTextField(50)? Cos we declared it at the top s
    cl.add(infoLabel, BorderLayout.CENTER); //Add label to container

    JButton button1 = new JButton("Are they repeated characters"); //Create a Jbuttton object
    cl.add(button1, BorderLayout.NORTH); //Add Jbutton to container

    txtGetUser = new JTextField(50); //NOTE: WHY didn't i do JTextField txtGetUser = new JTextField(50)? Cos we declared it at the top so it's global
    cl.add(txtGetUser, BorderLayout.SOUTH); //Add JtextField to container

    /////Events
    ButtonAction myAction = new ButtonAction(); //Create ActionListener
    button1.addActionListener(myAction);

    WindowDestroyer myListener = new WindowDestroyer();
    addWindowListener(myListener);
}
```



```java
public class ButtonAction implements ActionListener {

    public void actionPerformed(ActionEvent e) {

        String userString = txtGetUser.getText();
        int userStringLength = 0;
        char startStringChar = 'z';
        char charLoop = 'z';
        boolean repeatedChar = false;
        int newI = 0;

        userStringLength = userString.length(); //Get string length

        for (int i = 0; i < userStringLength && repeatedChar == false; i++) { //Do the loop until FullString length or a repeated char is found

            startStringChar = userString.charAt(i); //Get the initial character
            newI = i + 1; //Have to do it //Get the second initial character

            for (int j = newI; j < userStringLength && repeatedChar == false; j++) { //Loop compare current chracter with all the following character
                charLoop = userString.charAt(j);

                if (startStringChar == charLoop) { //If current character matches one of the following characters stop the loop
                    repeatedChar = true;
                    infoLabel.setText("There are repeated characters");
                }
            }
        }


        if(!repeatedChar) {
            infoLabel.setText("There are NO repeated characters");
        }
    }
   }
```

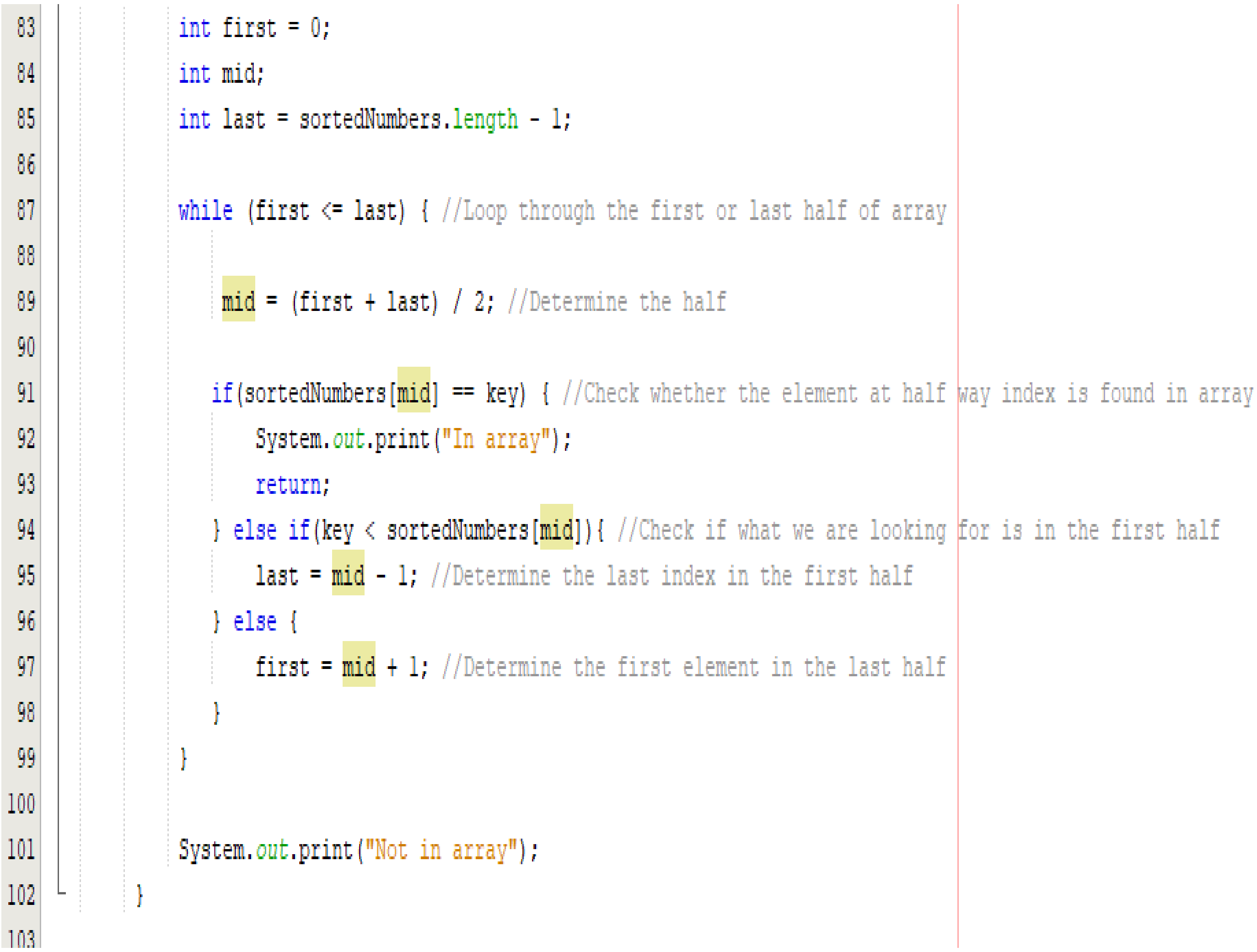
4) Get array of integers + Sort by through Insertion sort

Client program (method)-



```java
public static int[] CalcNumbersSortedInsertion(int[] userNumbers) { //list: 12,3,1 ...

    int[] sortNumbers = userNumbers; //NOTE: WHY didn't I just refer to userNumbers? Because it's a pass by reference when It comes to arrays

    for(int i = 1; i <= sortNumbers.length-1; i++) { //Go through each nummber in the array
                                        //Store temp = element we are currently checking
        int temp = sortNumbers[i]; //Start with element of array we are checking (n + 1 i.e after element)
        int j = 0;

        while(temp > sortNumbers[j]) { //Check if n + 1 (after element)> n (before element) so list: 12,3,1 ...
            j = j+1;    //if 3 > 12 then keep looping (NO)
        }

        for(int k = i; k > j; k--) {

            sortNumbers[k] = sortNumbers[k-1]; //Location of element we are checking (n+1) = before element (12)
        }                                       ///In this case the list: 12, 12,1

        sortNumbers[j] = temp; //Store ..... 3, 12,1 (I don't know how but just know)sortNumbers[0] = 3.  ALWAYS  sortNumbers[0] = ...

    }

    return sortNumbers;

}
```

5) Get array of integers + Sort by through Insertion sort + Determine whether a certain integer exist in array using Binary search



```
        int first = 0;
        int mid;
        int last = sortedNumbers.length - 1;

        while (first <= last) { //Loop through the first or last half of array

            mid = (first + last) / 2; //Determine the half

            if(sortedNumbers[mid] == key) { //Check whether the element at half way index is found in array
                System.out.print("In array");
                return;
            } else if(key < sortedNumbers[mid]){ //Check if what we are looking for is in the first half
                last = mid - 1; //Determine the last index in the first half
            } else {
                first = mid + 1; //Determine the first element in the last half
            }
        }

        System.out.print("Not in array");
    }
```