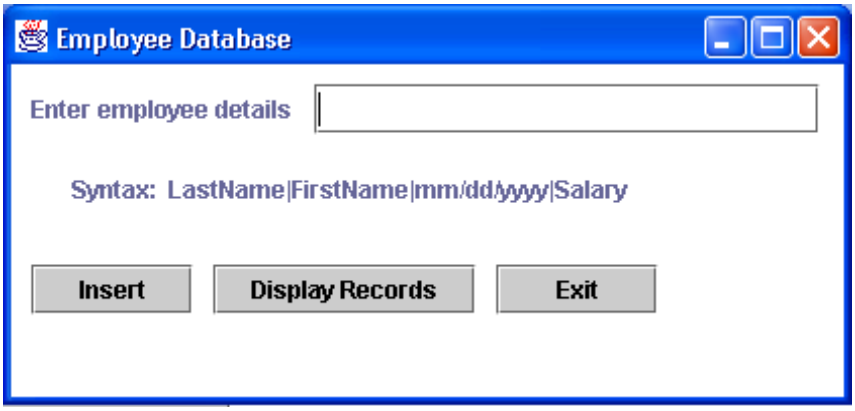
Lab Assignment 13 (week 13)

In this lab you are going to learn how to use the StringTokenizer class. Please look at the API for StringTokenizer at the below link.

http://java.sun.com/j2se/1.4.2/docs/api/java/util/StringTokenizer.html

You are going to design a GUI for implementing simple database of employees. You are going to store employee's last name, first name, data of joining and salary.

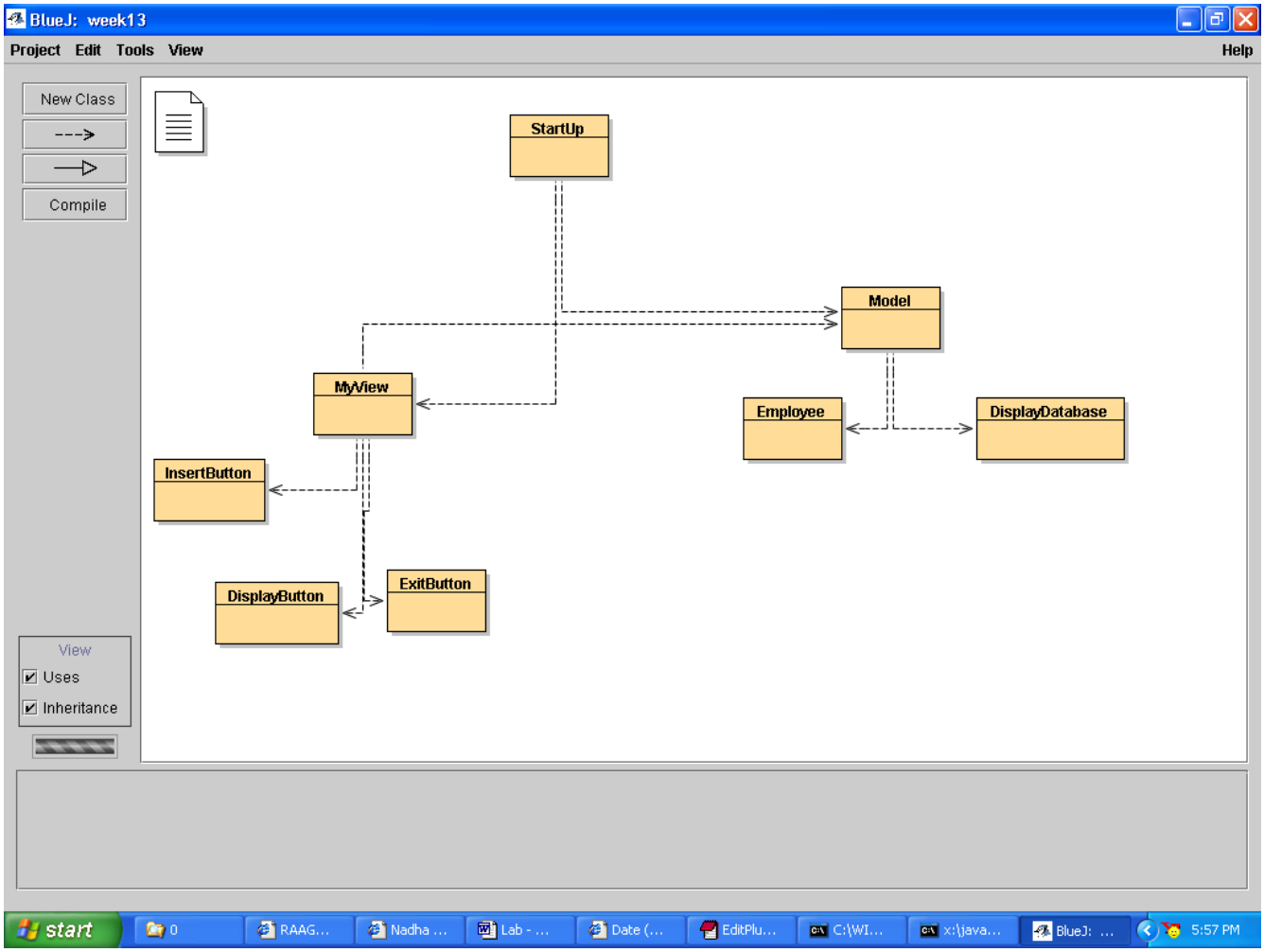
The GUI looks something like this



The only difference between this assignment and the assignment you did for Person database is that, in this assignment you have a single JTextField which takes a string which is tokenized using StringTokenizer into individual tokens containing LastName, FirstName, dataOfJoining and Salary. The delimiter used is "|". So, if you would like to store a record for an employee, you fill the JTextField as follows

LastName|FirstName|DataOfJoining(mm/dd/yyyy)|salary

At the end of the assignment your Bluej window should look something like this



Looking at the above figure, you see that there are

1. Three classes which extend JButton, one for each Jbutton, namely, Insert, DisplayDatabase and Exit.
2. Employee class is similar to the Person class, which you have seen before.
3. DisplayDatabase class is similar to SortedArrayDisplay class, which you have used before.
4. The only important thing you should look at is Model class which uses StringTokenizer.

NOTE: If you cannot copy the text from here, browse through this link for the code

www.cis.ksu.edu/~devaram/week13

Step 1: Open BlueJ and create a new project named Lab13.
Step 2: Create a main class named *StartUp*. Put the following code into it

```
class StartUp
{
        public static void main(String[] args)
        {
                MyView view = new MyView(new Model());
        }
}
```

Step 3: Create a class called *MyView* and paste the following code into it.

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

class  MyView extends JFrame
{
    private Container c;
    private JLabel lblPrompt = new JLabel("Enter employee details");
    private JTextField txtField = new JTextField();
    private JLabel lblSyntax = new JLabel("Syntax:
LastName|FirstName|mm/dd/yyyy|Salary");
    private InsertButton insertButton = new InsertButton("Insert", this);
    private ExitButton exitButton = new ExitButton("Exit");
    private DisplayButton displayButton = new DisplayButton("Display Records",
this);
    private Model model;

    public MyView(Model m)
    {
        super("Employee Database");
        model = m;
        c = getContentPane();
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        c.setBackground(Color.white);
        c.setLayout(null);
        setSize(420,200);

        lblPrompt.setBounds(10, 10, 180, 25);
        lblPrompt.setBackground(Color.white);

        txtField.setBounds(150,10,250,25);
        txtField.setBackground(Color.white);
```

```
        lblSyntax.setBounds(30,50,300,25);

        insertButton.setBounds(10,100,80,25);
        displayButton.setBounds(100,100,130,25);
        exitButton.setBounds(240,100,80,25);

        c.add(lblPrompt);
        c.add(txtField);
        c.add(lblSyntax);
        c.add(insertButton);
        c.add(displayButton);
        c.add(exitButton);
        setVisible(true);
    }

    public void insertRecord()
    {
        model.insertRecord(txtField.getText());
        System.out.println("The Emp Rec is :" + txtField.getText());
        txtField.setText("");
    }

    public void displayRecords()
    {
        model.displayRecords();
    }
```

Step 4: Write the three classes, which extend JButton on your own.
Step 5: The most important class is the *Model* class, which is given below.

```
import java.text.*;
import java.util.*;

class Model
{
    private final int MAX = 10;
    private DisplayDatabase displayDatabase;
    private Employee[] emp = new Employee[MAX];
    private StringTokenizer tok;
    private int count = 0;

    public void insertRecord(String empRecord)
    {
        String lName, fName, doj, sal;
        int salary;
        Date date_Of_Joining = null;
        tok = new StringTokenizer(empRecord, "|");
        lName = tok.nextToken();
        fName = tok.nextToken();
        doj = tok.nextToken();
```

```
                    sal = tok.nextToken();

                    /* Why is a try() catch block is used? What does it do?
                     */
            try
             {
                    /* TypeCasting: ( doj to date_Of_Joining )
                      * Converting the date_of_joining string to Data type
                      */
            date_Of_Joining =DateFormat.getDateInstance(DateFormat.SHORT).parse(doj);
                    }
                    catch (Exception e)
                    {
                     System.out.println("Use the format dd/mm/yy");
                    }

            /*Converting String to integer */
            salary = Integer.parseInt(sal);
            System.out.println("-->Model: " + lName + fName + date_Of_Joining.toString()
            + sal);
            emp[count] = new Employee(lName,fName,date_Of_Joining,salary);
            count++;
            }

             public void displayRecords()
             {
                    new DisplayDatabase(emp);
             }
    }
```

Step 6: Here is the *Employee* class, which looks similar to the Person class.

```
    import java.text.*;
    import java.util.Date;

    public class Employee {
        private String firstName;
        private String lastName;
        private Date doj;
        private int salary;

        public Employee(String fn, String ln, Date joiningDate, int sal) {
            firstName = fn;
            lastName = ln;
            doj = joiningDate;
            salary = sal;
        }

        public String toString() {
            SimpleDateFormat sdf = new SimpleDateFormat("MM/dd/yyyy");
              return   "Name: " + firstName  + " " + lastName +  "\tDateOfJoining: " +
sdf.format(doj) + "\tSalary: " + salary;
```

```
        }

    }
```

Step 7: The code for *DisplayDatabase* is as under

```
        import java.awt.*;
        import javax.swing.*;

        class DisplayDatabase extends JFrame
        {
                        public DisplayDatabase(Employee emp[]) {
                                super("Employee Database Display");
                                setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                                setSize(400, 200);
                                Container cp = getContentPane();
                                JTextArea text = new JTextArea();
                                cp.add(new JScrollPane(text));
                                text.append("\n");

                                for(int index = 0; index < emp.length && emp[index] != null;
index++)
                                {
                                        text.append(emp[index].toString());
                                        text.append("\n");
                                }
                                setVisible(true);
                }
        }
```

The Homework assignment due next week is based in these lines. It involves usage of StringTokenizer class as it is used in the above Model class. Please look at other methods that are provided for StringTokenizer in the API. Be ready to answer the questions that your TA asks about StringTokenizer.