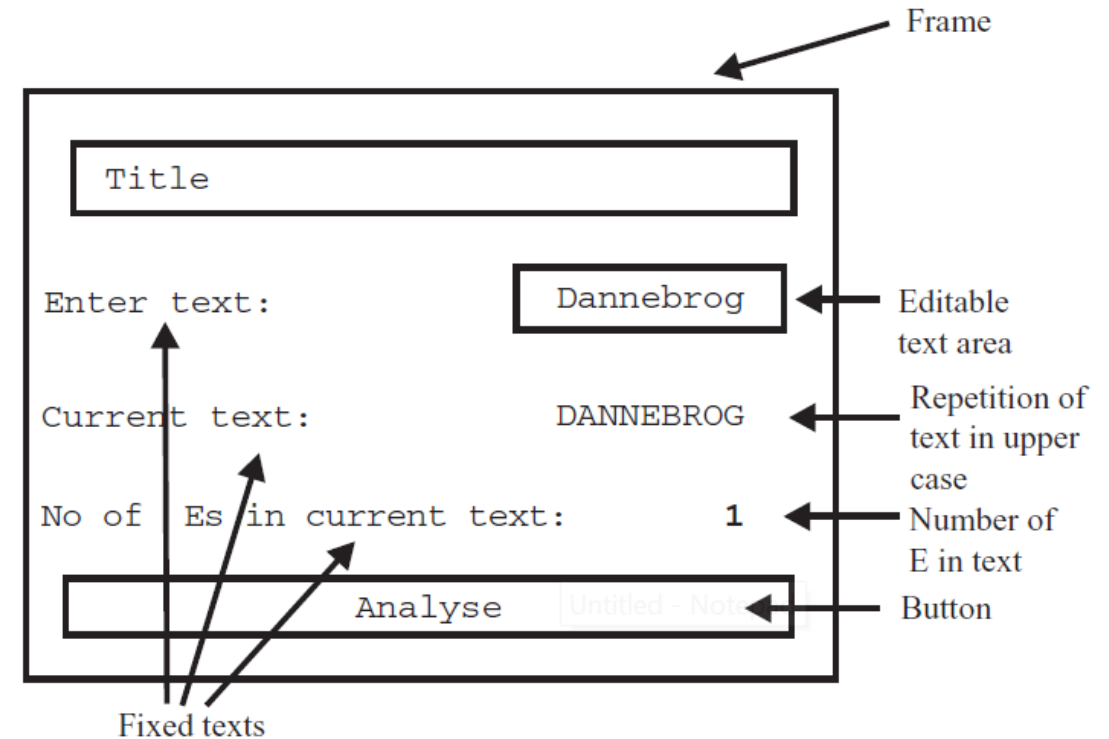


MVC

MVC Example 3 – Text Analysis

- To the right of the text 'Enter text:' is an editable area where the user can enter or change text.
- If one clicks on the button 'Analyse' then a small analysis of the text is performed and the result is displayed in the following way.
- The text is converted to capital letters and copied into the area to the right of 'Current text:'.
- The number of occurrences of the letter 'E' in upper or lower case is displayed to the right of 'No. of Es in current text:'.
- At the same time the content of the user editable area is erased.



Text Analysis - Model

```
public class TextAnalysisModel {  
    private int currentNumberOfEs;  
    private String currentText;  
  
    public TextAnalysisModel() {  
        currentNumberOfEs = 0;  
        currentText = "";  
    }  
  
    public void analyze(String str){  
        currentText = str.toUpperCase();  
        currentNumberOfEs = 0;  
        for (int i = 0; i < currentText.length();  
i++) {  
            if(currentText.charAt(i) == 'E'){  
                currentNumberOfEs++;  
            }  
        }  
    }  
  
    public int getCurrentNumberOfEs(){  
        return(currentNumberOfEs);  
    }  
  
    public String getCurrentText(){  
        return(currentText);  
    }  
}
```

Text Analysis – View

```
import java.awt.*;

import javax.swing.*;

public class TextAnalysisPanel extends JPanel

{

    private JLabel lastTextLabel;

    private JLabel numberOfEsLabel;

    private JLabel numberOfTextsLabel;

    private JTextField inputField;

    private TextAnalysisModel analysisModel;

    public TextAnalysisPanel()

    {

        analysisModel = new TextAnalysisModel();

        this.setBackground(Color.yellow);

        this.setLayout(new GridLayout(3,2,10,10));

        JLabel questionLabel    = new JLabel("Enter text:");

        JLabel replyLabel       = new JLabel("Current text:");

        JLabel numberTextLabel = new JLabel("No. of Es in current text:");

        lastTextLabel           = new JLabel("");

        numberOfEsLabel         = new JLabel("--");

        inputField              = new JTextField("");

    }

}
```

```
questionLabel.setOpaque(true);
questionLabel.setBackground(Color.black);
questionLabel.setForeground(Color.white);
```

```
replyLabel.setOpaque(true);
replyLabel.setBackground(Color.black);
replyLabel.setForeground(Color.white);
```

```
numberTextLabel.setOpaque(true);
numberTextLabel.setBackground(Color.black);
numberTextLabel.setForeground(Color.white);
```

```
numberOfEsLabel.setOpaque(true);
numberOfEsLabel.setBackground(Color.red);
numberOfEsLabel.setForeground(Color.white);
```

```
lastTextLabel.setOpaque(true);
lastTextLabel.setBackground(Color.red);
lastTextLabel.setForeground(Color.white);
```

```
this.add(questionLabel);
this.add(inputField);
this.add(replyLabel);
this.add(lastTextLabel);
this.add(numberTextLabel);
this.add(numberOfEsLabel);
```

```
}
```

```
public void startAnalysisAndDisplayResult()

{

    String text = inputField.getText();
    analysisModel.analyse(text);
    lastTextLabel.setText(analysisModel.getCurrentText());
    int noOfEs = analysisModel.getCurrentNumberOfEs();
    numberOfEsLabel.setText(Integer.toString(noOfEs));
    inputField.setText("");

}
```

Text Analysis - View (cont.)

```
import its.SimpleFrame.SimpleFrame;

import java.awt.*;

import javax.swing.JButton;

public class TextAnalysisFrame extends SimpleFrame
{
    public TextAnalysisFrame()
    {
        this.setSize(300,150);

        TextAnalysisPanel taPanel = new TextAnalysisPanel();

        this.getContentPane().add(taPanel,BorderLayout.CENTER);

        JButton analyseButton = new JButton("Analyse");

        analyseButton.setBackground(Color.blue);

        analyseButton.setForeground(Color.yellow);

        this.getContentPane().add(analyseButton,BorderLayout.SOUTH);

        TextAnalysisListener taList = new TextAnalysisListener(taPanel);

        analyseButton.addActionListener(taList);
    }
}
```

Text Analysis – Controller and main

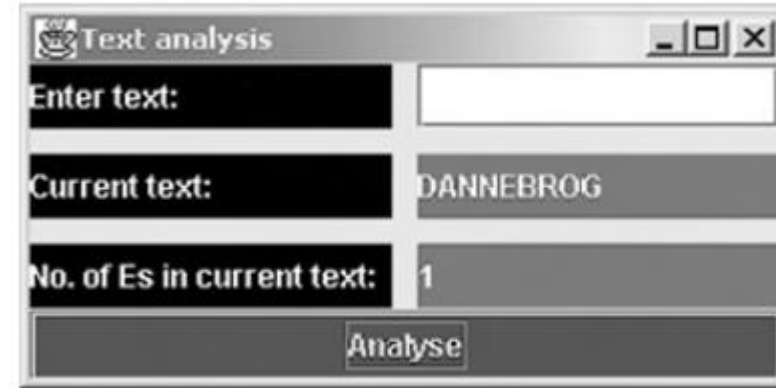
```
import java.awt.event.*;

public class TextAnalysisListener implements ActionListener
{
    private TextAnalysisPanel taPanel;

    public TextAnalysisListener(TextAnalysisPanel t)
    {
        taPanel = t;
    }

    public void actionPerformed(ActionEvent evt)
    {
        taPanel.startAnalysisAndDisplayResult();
    }
}
```

```
public class TextAnalysisDriver
{
    public static void main(String[] args)
    {
        TextAnalysisFrame taFrame = new TextAnalysisFrame();
        taFrame.showIt("Text analysis");
    }
}
```



Text Analysis - Control updates

Update : Add Reset Button and show total number of Es in all analyzed text and total number of analyzed text

The screenshot shows a window titled "Text analysis" with standard Windows window controls (minimize, maximize, close). The window contains a "Reset" button in the top right corner and an "Analyse" button in the bottom right corner. The main area is divided into five rows, each with a label on the left and a corresponding input field or display area on the right. The labels are "Enter text:", "Current text:", "No. of Es in current text:", "No. of analyzed text:", and "No. of Es in all analyzed text:". The input fields for "Current text:", "No. of Es in current text:", "No. of analyzed text:", and "No. of Es in all analyzed text:" are currently empty and display "--".

Label	Value
Enter text:	<input type="text"/>
Current text:	<input type="text"/>
No. of Es in current text:	--
No. of analyzed text:	--
No. of Es in all analyzed text:	--

Text Analysis – Model updated

```
package TextAnalysis;
```

```
public class TextAnalysisModel {

    private int totalNumberOfEs;

    private int currentNumberOfEs;

    private int totalNumberOfTexts;

    private String currentText;

    public TextAnalysisModel() {

        totalNumberOfEs = 0;
        totalNumberOfTexts = 0;
        currentText = "";

    }

    public void analyse(String str) {

        currentText = str.toUpperCase();

        currentNumberOfEs = 0;

        for (int i = 0; i < currentText.length();
i++) {

            if (currentText.charAt(i) == 'E') {

                currentNumberOfEs++;

            }

        }

        totalNumberOfEs += currentNumberOfEs;

        totalNumberOfTexts++;

    }

}
```

```
public void reset() {

    currentText = "";
    currentNumberOfEs = 0;
    totalNumberOfEs = 0;
    totalNumberOfTexts = 0;

}

public int getCurrentNumberOfEs() {

    return (currentNumberOfEs);

}

public String getCurrentText() {

    return (currentText);

}

public int getTotalNumberOfEs() {

    return (totalNumberOfEs);

}

public int getTotalNumberOfTexts() {

    return (totalNumberOfTexts);

}

}
```


Text Analysis – Controller updated

```
package TextAnalysis;

import java.awt.event.*;

public class TextAnalysisListener implements
ActionListener {

    private TextAnalysisPanel taPanel;

    public TextAnalysisListener(TextAnalysisPanel
t) {

        taPanel = t;

    }

    public void actionPerformed(ActionEvent evt) {

        if (evt.getActionCommand().equals("Reset"))
{

            taPanel.reset();

        } else if
(evt.getActionCommand().equals("Analyse")) {

            taPanel.startAnalysis();

        }

    }

}
```

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

Text Analysis – View updated

```
public class TextAnalysisPanel extends JPanel {
```

```
    private JLabel lastTextLabel;
    private JLabel numberOfEsLabel;
    private JLabel numberOfTextsLabel;
    private JLabel totalNumberOfEsLabel;
    private JTextField inputField;
    private TextAnalysisModel analysisModel;
```

```
    public TextAnalysisPanel() {
        analysisModel = new TextAnalysisModel();
        this.setBackground(Color.yellow);
        this.setLayout(new GridLayout(5, 2, 10, 10));
        JLabel questionLabel = new JLabel("Enter text:");
        JLabel replyLabel = new JLabel("Current text:");
        JLabel numberTextLabel = new JLabel("No. of Es in current
text:");
        JLabel totalLabel = new JLabel("No. of Es in all analyzed
text:");
        JLabel TextsLabel = new JLabel("No. of analyzed text:");
        lastTextLabel = new JLabel("");
        numberOfEsLabel = new JLabel("--");
        numberOfTextsLabel = new JLabel("--");
        totalNumberOfEsLabel = new JLabel("--");
        inputField = new JTextField("");
        questionLabel.setOpaque(true);
        questionLabel.setBackground(Color.black);
        questionLabel.setForeground(Color.white);
        replyLabel.setOpaque(true);
        replyLabel.setBackground(Color.black);
        replyLabel.setForeground(Color.white);
        numberTextLabel.setOpaque(true);
        numberTextLabel.setBackground(Color.black);
        numberTextLabel.setForeground(Color.white);
        TextsLabel.setOpaque(true);
        TextsLabel.setBackground(Color.black);
        TextsLabel.setForeground(Color.white);
        totalLabel.setOpaque(true);
        totalLabel.setBackground(Color.black);
        totalLabel.setForeground(Color.white);
        numberOfEsLabel.setOpaque(true);
        numberOfEsLabel.setBackground(Color.red);
        numberOfEsLabel.setForeground(Color.white)
```

```
        lastTextLabel.setOpaque(true);
        lastTextLabel.setBackground(Color.red);
        lastTextLabel.setForeground(Color.white);
        numberOfTextsLabel.setOpaque(true);
        numberOfTextsLabel.setBackground(Color.red);
        numberOfTextsLabel.setForeground(Color.white);
        totalNumberOfEsLabel.setOpaque(true);
        totalNumberOfEsLabel.setBackground(Color.red);
        totalNumberOfEsLabel.setForeground(Color.white);
        this.add(questionLabel);
        this.add(inputField);
        this.add(replyLabel);
        this.add(lastTextLabel);
        this.add(numberTextLabel);
        this.add(numberOfEsLabel);
        this.add(TextsLabel);
        this.add(numberOfTextsLabel);
        this.add(totalLabel);
        this.add(totalNumberOfEsLabel);
```

```
    }
```

```
    public void reset() {
        analysisModel.reset();
        displayResult();
    }
```

```
    private void displayResult() {
        lastTextLabel.setText(analysisModel.getCurrentText());
        int noOfEs = analysisModel.getCurrentNumberOfEs();
        numberOfEsLabel.setText(Integer.toString(noOfEs));
        inputField.setText("");
```

```
        numberOfTextsLabel.setText(Integer.toString(analysisModel.getTotalNumberOfTexts()));
        totalNumberOfEsLabel.setText(Integer.toString(analysisModel.getTotalNumberOfEs()));
    }
```

```
    public void startAnalysis() {
        String text = inputField.getText();
        analysisModel.analyse(text);
        displayResult();
    }
```

```
}
```

Text Analysis – View updated (cont.)

```
package TextAnalysis;

import Counter.SimpleFrame;
import java.awt.*;
import javax.swing.*;

public class TextAnalysisFrame extends SimpleFrame {

    public TextAnalysisFrame() {

        this.setSize(300, 150);
        TextAnalysisPanel taPanel = new TextAnalysisPanel();

        this.getContentPane().add(taPanel, BorderLayout.CENTER);

        JButton analyseButton = new JButton("Analyse");

        analyseButton.setBackground(Color.blue);

        analyseButton.setForeground(Color.yellow);

        this.getContentPane().add(analyseButton, BorderLayout.SOUTH);


        JButton resetButton = new JButton("Reset");

        resetButton.setBackground(Color.blue);

        resetButton.setForeground(Color.yellow);

        this.getContentPane().add(resetButton, BorderLayout.NORTH);


        this.pack();

        TextAnalysisListener taList = new TextAnalysisListener(taPanel);

        analyseButton.addActionListener(taList);

        resetButton.addActionListener(taList);

    }

}
```

MVC Example 4 – User Information Model

```
public class Model {  
  
    private String firstname;  
    private String lastname;  
  
    public Model(String firstname, String lastname) {  
        this.firstname = firstname;  
        this.lastname = lastname;  
    }  
  
    public String getFirstname() {  
        return firstname;  
    }  
  
    public void setFirstname(String firstname) {  
        this.firstname = firstname;  
    }  
  
    public String getLastname() {  
        return lastname;  
    }  
  
    public void setLastname(String lastname) {  
        this.lastname = lastname;  
    }  
  
}
```

MVC Example 4 – User Information View

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

public class View {
    private JFrame frame;
    private JLabel firstnameLabel;
    private JLabel lastnameLabel;
    private JTextField firstnameTextfield;
    private JTextField lastnameTextfield;
    private JButton firstnameSaveButton;
    private JButton lastnameSaveButton;
    private JButton hello;
    private JButton bye;

    public View(String title) {
        frame = new JFrame(title);
        frame.getContentPane().setLayout(new BorderLayout());
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(500, 120);
        frame.setVisible(true);

        firstnameLabel = new JLabel("Firstname :");
        lastnameLabel = new JLabel("Lastname :");
        firstnameTextfield = new JTextField();
        lastnameTextfield = new JTextField();
        firstnameSaveButton = new JButton("Save firstname");
        lastnameSaveButton = new JButton("Save lastname");
        hello = new JButton("Hello!");
        bye = new JButton("Bye!");

        GroupLayout layout = new GroupLayout(frame.getContentPane());
        layout.setAutoCreateGaps(true);
        layout.setAutoCreateContainerGaps(true);
        layout.setHorizontalGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
                .addComponent(firstnameLabel)
                .addComponent(lastnameLabel)
                .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
                    .addComponent(firstnameTextfield)
                    .addComponent(lastnameTextfield)
                    .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
                        .addComponent(firstnameSaveButton)
                        .addComponent(lastnameSaveButton)
                        .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
                            .addComponent(hello)
                            .addComponent(bye))))))
            .addGap(10));
        layout.setVerticalGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
                .addComponent(firstnameLabel)
                .addComponent(firstnameTextfield)
                .addComponent(firstnameSaveButton)
                .addComponent(hello))
            .addGroup(layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
                .addComponent(lastnameLabel)
                .addComponent(lastnameTextfield)
                .addComponent(lastnameSaveButton)
                .addComponent(bye)));

        layout.linkSize(SwingConstants.HORIZONTAL, firstnameSaveButton, lastnameSaveButton);
        layout.linkSize(SwingConstants.HORIZONTAL, hello, bye);
        frame.getContentPane().setLayout(layout);
    }
}
```

MVC Example 4 – User Information View (cont.)

```
public JFrame getFrame() {    return frame;    }

    public void setFrame(JFrame frame) {    this.frame = frame;    }

    public JLabel getFirstnameLabel() {    return firstnameLabel;    }

    public void setFirstnameLabel(JLabel firstnameLabel) {    this.firstnameLabel = firstnameLabel;    }

    public JLabel getLastnameLabel() {    return lastnameLabel;    }

    public void setLastnameLabel(JLabel lastnameLabel) {    this.lastnameLabel = lastnameLabel;    }

    public JTextField getFirstnameTextfield() {    return firstnameTextfield;    }

    public void setFirstnameTextfield(JTextField firstnameTextfield) {    this.firstnameTextfield = firstnameTextfield;    }

    public JTextField getLastnameTextfield() {    return lastnameTextfield;    }

    public void setLastnameTextfield(JTextField lastnameTextfield) {    this.lastnameTextfield = lastnameTextfield;    }

    public JButton getFirstnameSaveButton() {    return firstnameSaveButton;    }

    public void setFirstnameSaveButton(JButton firstnameSaveButton) {    this.firstnameSaveButton = firstnameSaveButton;    }

    public JButton getLastnameSaveButton() {    return lastnameSaveButton;    }

    public void setLastnameSaveButton(JButton lastnameSaveButton) {    this.lastnameSaveButton = lastnameSaveButton;    }

    public JButton getHello() {    return hello;    }

    public void setHello(JButton hello) {    this.hello = hello;    }

    public JButton getBye() {    return bye;    }

    public void setBye(JButton bye) {    this.bye = bye;    }

}
```

MVC Example 4 – User Information Controller

```
import javax.swing.*;

public class Controller implements ActionListener{
    private Model model;
    private View view;

    public Controller(Model m, View v) {
        model = m;
        view = v;
        initView();
    }

    public void initView() {
        view.getFirstnameTextfield().setText(model.getFirstname());
        view.getLastnameTextfield().setText(model.getLastname());
    }

    public void initController() {
        view.getFirstnameSaveButton().addActionListener(e -> saveFirstname());
        view.getLastnameSaveButton().addActionListener(e -> saveLastname());
        view.getHello().addActionListener(e -> sayHello());
        view.getBye().addActionListener(e -> sayBye());
    }

    public void actionPerformed(ActionEvent evt) {
        if (evt.getSource().equals(view.getFirstnameSaveButton())) {
            saveFirstname();
        } else if (evt.getSource().equals(view.getLastnameSaveButton())) {
            saveLastname();
        } else if (evt.getSource().equals(view.getHello())) {
            sayHello();
        } else if (evt.getSource().equals(view.getBye())) {
            sayBye();
        }
    }

    private void saveFirstname() {
        model.setFirstname(view.getFirstnameTextfield().getText());
        JOptionPane.showMessageDialog(null, "Firstname saved : " +
            model.getFirstname(), "Info", JOptionPane.INFORMATION_MESSAGE);
    }

    private void saveLastname() {
        model.setLastname(view.getLastnameTextfield().getText());
        JOptionPane.showMessageDialog(null, "Lastname saved : " +
            model.getLastname(), "Info", JOptionPane.INFORMATION_MESSAGE);
    }

    private void sayHello() {
        JOptionPane.showMessageDialog(null, "Hello " + model.getFirstname()
            + " " + model.getLastname(), "Info", JOptionPane.INFORMATION_MESSAGE);
    }

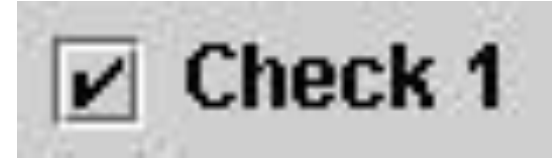
    private void sayBye() {
        System.exit(0);
    }
}
```

MVC Example 4 – User Information Main

```
public class App {  
  
    public static void main(String[] args) {  
        // Assemble all the pieces of the MVC  
        Model m = new Model("Random", "User");  
        View v = new View("MVC - User Information");  
        Controller c = new Controller(m, v);  
        c.initController();  
    }  
}
```


JCheckBox, JRadioButton

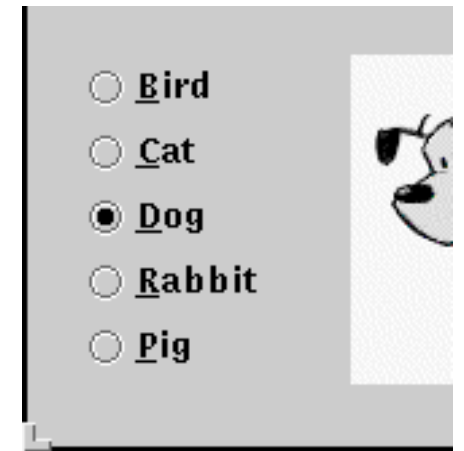
*a toggleable yes/no value (checkbox)
or a way choose between options (radio)*



- `public JCheckBox(String text)`
`public JCheckBox(String text, boolean checked)`
`public JRadioButton(String text)`
Creates a checked/unchecked check box with given text.
- `public boolean isSelected()`
Returns true if the check box is checked.
- `public void setSelected(boolean selected)`
Sets box to be checked/unchecked.

ButtonGroup

a logical collection to ensure that exactly one radio button from a group is checked at a time



- `public ButtonGroup()`
- `public void add(JRadioButton button)`
- The `ButtonGroup` is not a graphical component, just a logical group; the `RadioButtons` themselves also need to be added to an onscreen container to be seen.



Icon

a picture that can appear inside a component

- `public class ImageIcon implements Icon`
 - `public ImageIcon(String filename)`
 - `public ImageIcon(URL address)`
- **in JButton, JRadioButton, JCheckBox, JLabel, etc...**
 - **constructor that takes an Icon**
 - `public void setIcon(Icon)`
 - `public void setSelectedIcon(Icon)`
 - `public void setRolloverIcon(Icon)`

JComboBox

a drop-down list of selectable items



- `public JComboBox()`
- `JComboBox(String[] listEntries)`
- `JComboBox(Vector[] listEntries)`
Constructs a combo box.
- `int getItemCount()`
- `setMaximumRowCount(int rowCount)`
- `public void
addActionListener(ActionListener al)`
Causes an action event to be sent to listener `al` when the user selects or types a new item in the combo box.

JComboBox methods

- `public void addItem(Object item)`
 - `public Object getItemAt(int index)`
 - `public void removeAllItems()`
 - `public void removeItem(Object item)`
 - `public void removeItemAt(int index)`

 - `public int getSelectedIndex()`
 - `public Object getSelectedItem()`
 - `public void setSelectedItem(Object item)`
 - `public void setSelectedIndex(int index)`
 - `public void setEnabled(boolean enabled)`
 - `public void setEditable(boolean editable)`
- If editable, the user can type new arbitrary values into the box.**

JList

a list of selectable pre-defined text items



- `public JList()`
Constructs an empty JList.
- `public JList(ListModel model)`
`public JList(Object[] data)`
`public JList(Vector data)`
Constructs a JList that displays the given data.
- `public void addListSelectionListener(
 ListSelectionListener lsl)`
Adds the given listener to be informed when the selected index changes for this list.

List Model

- `DefaultListModel()` `DefaultListModel()` is the constructor which creates a list model without any entries.
- `int size()` `size()` returns the number of entries currently in the list model.
- `boolean contains(Object entry)` `contains(Object entry)` returns true if object entry is in the list model, and false otherwise.
- `Object get(int position)` `get(int position)` returns the list entry at position position. Throws an exception if position is negative or greater than or equal to the current number of list entries. As the return type is `Object`, an explicit cast to the correct type might be needed.
- `void insertElementAt(Object entry, int position)` `insertElementAt(Object entry, int position)` adds entry entry at position position to the list model. Throws an exception if position is an invalid index.
- `void addElement(Object entry)` `addElement(Object entry)` adds an entry at the end of the list.
- `void removeElementAt(int position)` `removeElementAt(int position)` removes the entry at position position. Throws an exception if position is an invalid index. The remaining list entries are renumbered to maintain a consecutive indexing.
- `boolean removeElement(Object entry)` `removeElement(Object entry)` removes the entry entry. If successful true is returned, otherwise false is returned.
- `void removeAllElements()` `removeAllElements()` removes all entries from the list model.

JTextArea

an input control for typing text values (area = multi-line)

- ```
public JTextArea(int lines, int columns)
```

Creates a new field, the given number of letters wide.
- ```
public String getText()
```

Returns the text currently in the field.
- ```
public void setText(String text)
```

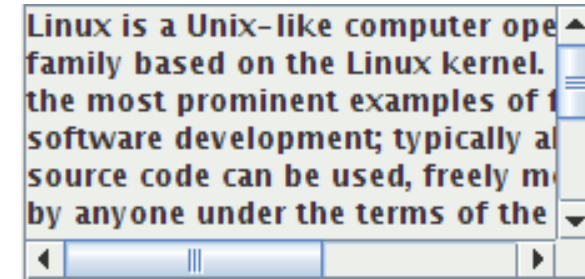
Sets field's text to be the given string.

Verify that the RJ45 cable is connected to the WAN plug on the back of the Pipeline unit.



# JScrollPane

*a container that adds scrollbars  
around any other component*

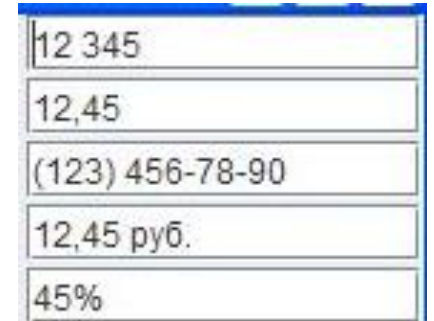


- `public JScrollPane(Component comp)`  
Wraps the given component with scrollbars.
  - After constructing the scroll pane, you must add the scroll pane, not the original component, to the onscreen container:

```
myContainer.add(new JScrollPane(textarea),
 BorderLayout.CENTER);
```

# JFormattedTextField

*a text box that allows special formatting  
and can enforce constraints about allowable text*



|                 |
|-----------------|
| 12 345          |
| 12,45           |
| (123) 456-78-90 |
| 12,45 pyđ.      |
| 45%             |

- `public JFormattedTextField(Format format)`  
Creates a new field that constrains itself to the given text format.  
(e.g. `DateFormat`, `NumberFormat`, `CurrencyFormat`, `MaskFormat`)
- `public Object getValue()`  
`public void setValue(Object value)`  
The value currently set in the field, which may lag behind the text.
- `public void setFocusLostBehavior(int b)`  
Sets what field should do if user stops editing and value is illegal.

# MVC Example 5 – Person Information - Controller

```
package controller;

import Model.*;
import gui.*;
import java.awt.event.*;
import java.io.*;

public class PersonController implements ActionListener {

 FormPanel fpanel;

 public PersonController(FormPanel fpanel) {
 this.fpanel = fpanel;
 fpanel.addListener(this);
 fpanel.addExportListener(this);
 }

 @Override
 public void actionPerformed(ActionEvent e) {
 if (e.getActionCommand().equals("Add")) {
 Person p = fpanel.setPersonInfo();
 fpanel.getDatabase().addPerson(p);

 fpanel.appendText(p.getName() + "," +
p.getOccupation() + "\n");
 } else if (e.getActionCommand().equals("Export")) {
 writeToDatabase(fpanel.getDatabase());
 }
 }
}
```

```
public void writeToDatabase(Database db) {
 File file = new File("output.txt");
 FileWriter fr = null;
 BufferedWriter br = null;

 try {
 fr = new FileWriter(file);
 br = new BufferedWriter(fr);
 for (Person person : db.getPeople()) {
 br.write(person.getName() + ",");
 br.write(person.getOccupation() + ",");
 br.write(person.getGender().toString() + ",");
 br.write(person.getAgeCategory().toString() +
",");

 br.write(person.getEmpCat().toString());
 br.newLine();
 }
 } catch (IOException e) {
 e.printStackTrace();
 } finally {
 try {
 br.close();
 fr.close();
 } catch (IOException e) {
 e.printStackTrace();
 }
 }

 System.out.println("ArrayList written to file
successfully.");
}
}
```

# MVC Example 5 – Person Information - Model

```
public class Person {

 private static int count = 0;
 private int id;
 private String name;
 private String occupation;
 private AgeCategory ageCategory;
 private EmploymentCategory empCat;
 private Gender gender;
 public Person() {
 this.id = count;
 count++;
 }

 public int getId() { return id; }

 public void setId(int id) {this.id = id;}

 public String getName() {return name;}

 public void setName(String name) { this.name = name;}

 public String getOccupation() {return occupation;}

 public void setOccupation(String occupation) { this.occupation = occupation;}

 public AgeCategory getAgeCategory() {return ageCategory;}

 public void setAgeCategory(AgeCategory ageCategory) {this.ageCategory = ageCategory;}

 public EmploymentCategory getEmpCat() {return empCat;}

 public void setEmpCat(EmploymentCategory empCat) {this.empCat = empCat;}

 public Gender getGender() {return gender;}
```

# MVC Example 5 – Person Information – Model (cont.)

```
public enum AgeCategory {
 child,
 adult,
 senior
}

public enum EmploymentCategory {
 employed,
 self_Employed,
 unemployed,
 other
}

public enum Gender {
 male,
 female
}

import java.util.ArrayList;
import java.util.List;

public class Database {

 private ArrayList<Person> people;

 public Database() {
 people = new
ArrayList<Person>();
 }

 public void addPerson(Person person)
 {
 people.add(person);
 }

 public List<Person> getPeople() {
 return people;
 }
}
```

# MVC Example 5 – Person Information - View

```
package gui;

import java.awt.*;
import javax.swing.*;
import controller.PersonController;

public class MainFrame extends JFrame {

 private FormPanel formPanel;
 PersonController pc;

 public MainFrame() {
 super("Add Person");

 formPanel = new FormPanel();
 pc = new PersonController(formPanel);

 setLayout(new BorderLayout());
 add(formPanel, BorderLayout.CENTER);
 setMinimumSize(new Dimension(500, 400));
 setSize(600, 500);
 pack();
 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 setVisible(true);
 }
}
```

```
public class App {

 public static void main(String[] args)
 {
 new MainFrame();
 }
}
```

# MVC Example 5 – Person Information – View (cont.)

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

public class FormPanel extends JPanel {

 private Database db;
 private JLabel nameLabel;
 private JLabel occupationLabel;
 private JLabel ageLabel;
 private JLabel genderLabel;
 private JLabel empLabel;
 private JTextField nameField;
 private JTextField occupationField;
 private JList ageList;
 private JRadioButton maleRadio;
 private JRadioButton femaleRadio;
 private ButtonGroup genderGroup;
 private JComboBox empCombo;
 private JTextArea textArea;
 JPanel westPanel;
 JPanel textPanel;
 private JButton addBtn;
 private JButton exportBtn;

 public FormPanel() {
 db = new Database();
 westPanel = new JPanel();
 textPanel = new JPanel();
 textArea = new JTextArea(35, 20);
 textPanel.add(textArea);

 nameLabel = new JLabel("Name: ");
 westPanel.add(nameLabel);
 nameField = new JTextField(10);
 westPanel.add(nameField);
 occupationLabel = new JLabel("Occupation: ");
 westPanel.add(occupationLabel);
 occupationField = new JTextField(10);
 westPanel.add(occupationField);
```

```
 ageLabel = new JLabel("Age Category: ");
 westPanel.add(ageLabel);
 ageList = new JList();
 // Set up list box
 DefaultListModel ageModel = new DefaultListModel();
 ageModel.addElement(new AgeCategory(0, "child"));
 ageModel.addElement(new AgeCategory(1, "adult"));
 ageModel.addElement(new AgeCategory(2, "senior"));
 ageList.setModel(ageModel);
 ageList.setPreferredSize(new Dimension(20, 20));

 ageList.setBorder(BorderFactory.createRaisedBevelBorder());
 ageList.setSelectedIndex(1);
 westPanel.add(ageList);

 genderLabel = new JLabel("Gender: ");
 westPanel.add(genderLabel);
 maleRadio = new JRadioButton("male");
 femaleRadio = new JRadioButton("female");
 maleRadio.setActionCommand("male");
 femaleRadio.setActionCommand("female");
 genderGroup = new ButtonGroup();
 maleRadio.setSelected(true);
 genderGroup.add(maleRadio);
 genderGroup.add(femaleRadio);
 JPanel genderPanel = new JPanel();
 genderPanel.add(maleRadio);
 genderPanel.add(femaleRadio);
 genderPanel.setLayout(new BoxLayout(genderPanel,
 BoxLayout.Y_AXIS));
 westPanel.add(genderPanel);
```

# MVC Example 5 – Person Information – View (cont.)

```
empLabel = new JLabel("Employment Category: ");
westPanel.add(empLabel);
empCombo = new JComboBox();
// Set up combo box.
DefaultComboBoxModel empModel = new
DefaultComboBoxModel();
empModel.addElement("Employed");
empModel.addElement("Self Employed");
empModel.addElement("Unemployed");
empCombo.setModel(empModel);
empCombo.setSelectedIndex(0);
empCombo.setEditable(true);
westPanel.add(empCombo);

addBtn = new JButton("Add");
exportBtn = new JButton("Export");
westPanel.add(addBtn);
westPanel.add(exportBtn);

westPanel.setLayout(new GridLayout(7, 2,
2, 2));
setLayout(new BorderLayout());
add(textPanel, BorderLayout.EAST);
add(westPanel, BorderLayout.WEST);
}

public void addListener(ActionListener al) {
 addBtn.addActionListener(al);
}

public void addExportListener(ActionListener
al) {
 exportBtn.addActionListener(al);
}
```

```
public void appendText(String text) {
 textArea.append(text);
}

public Database getDatabase() { return this.db; }

public String getPersonName() { return nameField.getText();}

public void setPersonName(String name) {
 this.nameField.setText(name);
}

public String getOccupation() {
 return occupationField.getText();
}

public void setOccupation(String occupation) {
 occupationField.setText(occupation);
}

public String getAgeCategory() {
 return ageList.getSelectedValue().toString();
}

public Gender getGender() {
 Gender g;
 if (genderGroup.getSelection().getActionCommand() ==
Model.Gender.male.toString()) {
 g = Gender.male;
 } else {
 g = Gender.female;
 }
 return g;
}

public int getEmploymentCategory() {
 return empCombo.getSelectedIndex();
}
```



# MVC Example 5 – Person Information – View (cont.)

```
public Person setPersonInfo() {
 Person p = new Person();
 p.setName(getPersonName());
 p.setOccupation(getOccupation());

 p.setAgeCategory(Model.AgeCategory.valueOf(
 getAgeCategory()));
 p.setGender(getGender());

 p.setEmpCat(Model.EmploymentCategory.values
 () [getEmploymentCategory()]);
 return p;
}
```

```
class AgeCategory {

 private int id;
 private String text;

 public AgeCategory(int id, String text) {
 this.id = id;
 this.text = text;
 }

 public String toString() {
 return text;
 }

 public int getId() {
 return id;
 }
}
```