

JCheckBoxMenuItem

Constructor	Περιγραφή
JCheckBoxMenuItem()	Δημιουργεί στο μενού με μη επιλεγμένο με check box χωρίς καθορισμένο κείμενο ή εικονίδιο.
JCheckBoxMenuItem(Action a)	Δημιουργεί στο μενού του οποίου οι ιδιότητες του παίρνονται από την ενέργεια a
JCheckBoxMenuItem(Icon icon)	Δημιουργεί στο μενού με μη επιλεγμένο με check box με εικονίδιο.
JCheckBoxMenuItem(String text)	Δημιουργεί στο μενού με μη επιλεγμένο με check box με κείμενο.
JCheckBoxMenuItem(String text, boolean b)	Δημιουργεί στο μενού ένα check box με κείμενο και επιλογή.
JCheckBoxMenuItem(String text, Icon icon)	Δημιουργεί στο μενού με μη επιλεγμένο με check box με κείμενο και εικονίδιο
JCheckBoxMenuItem(String text, Icon icon, boolean b)	It creates a check box menu item with the specified text, icon, and selection state.

ΠΑΡΑΔΕΙΓΜΑ

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import javax.swing.AbstractButton;
import javax.swing.JCheckBoxMenuItem;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;

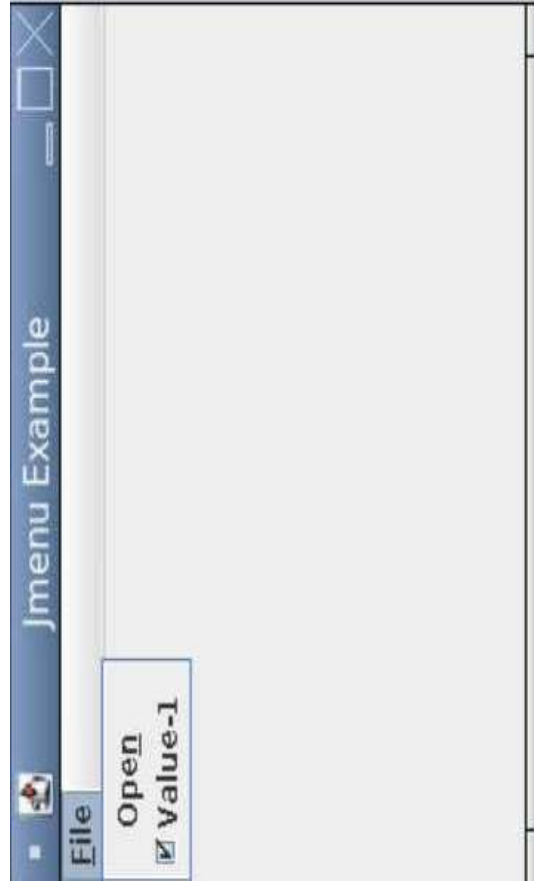
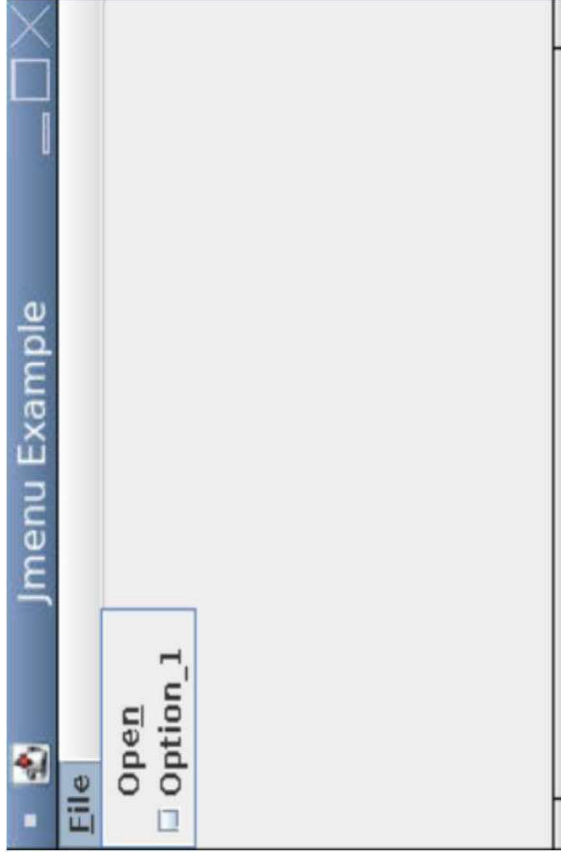
public class Main {
    public static void main(final String args[]) {
        JFrame frame = new JFrame("Jmenu Example");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JMenuBar menuBar = new JMenuBar();
        // File Menu, F - Mnemonic
        JMenu fileMenu = new JMenu("File");
        fileMenu.setMnemonic(KeyEvent.VK_F);
        menuBar.add(fileMenu);
        // File->New, N - Mnemonic
        JMenuItem menuItem1 = new JMenuItem("Open", KeyEvent.VK_N);
        fileMenu.add(menuItem1);
        JCheckBoxMenuItem caseMenuItem = new
        JCheckBoxMenuItem("Option_1");
        caseMenuItem.setMnemonic(KeyEvent.VK_C);
        fileMenu.add(caseMenuItem);
    }
}
```

```
ActionListener aListener = new ActionListener()
{
    public void actionPerformed(ActionEvent event)
    {
        AbstractButton aButton = (AbstractButton) event.getSource();
        boolean selected = aButton.getModel().isSelected();
        String newLabel;
        if (selected) {
            newLabel = "Value-1";
        } else {
            newLabel = "Value-2";
        }
        aButton.setText(newLabel);
    }
};

caseMenuItem.addActionListener(aListener);
frame.setMenuBar(menuBar);
frame.setSize(350, 250);
frame.setVisible(true);
}
```

Η μέθοδος `setMnemonic` συσχετίζει

μνημονικά στοιχεία με το μενού. Τα μνημονικά στοιχεία επιτρέπουν στο χρήστη να αλληλεπιδρά με τα στοιχεία του μενού χρησιμοποιώντας τα πλήκτρα στο πληκτρολόγιο



Παράδειγμα

Δημιουργήστε μια αριθμομηχανή.

Σχεδιασμός

Τα αποτελέσματα εμφανίζονται στην κορυφή τα πλήκτρα από κάτω

1	2	3	+	
4	5	6	-	
7	8	9	*	
C	0	=	/	

Textbox για να εκτυπώνει το αποτέλεσμα

Κουμπιά για καθένα από τα πλήκτρα

Χρειαζόμαστε ένα border layout για να βάλουμε το textbox στην κορυφή. Στο κέντρο θα βάλουμε τα κουμπιά. Βάζουμε ένα panel με grid layout

ΥΛΟΤΥΠΩΝ

```
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.Border;
import java.awt.BorderLayout;
import java.awt.GridLayout;
import java.awt.FlowLayout;
import java.awt.Color;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class Main extends JFrame implements ActionListener
{
    public static final int WIDTH = 300;
    public static final int HEIGHT = 200;
    private JPanel topPanel;
    private JPanel centralPanel;
    private JPanel gridPanel;

    public Main()
    {
        super("Panel Demonstration");
        setSize(WIDTH, HEIGHT);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.Border;
import java.awt.BorderLayout;
import java.awt.GridLayout;
import java.awt.FlowLayout;
import java.awt.Color;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class Main extends JFrame implements ActionListener
{
    public static final int WIDTH = 300;
    public static final int HEIGHT = 200;
    private JPanel topPanel;
    private JPanel centralPanel;
    private JPanel gridPanel;

    public Main()
    {
        super("Panel Demonstration");
        setSize(WIDTH, HEIGHT);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        topPanel = new JPanel();
        JTextField top=new JTextField("");
        topPanel.setLayout( new FlowLayout());
        topPanel.add(top);
        add(topPanel, BorderLayout.NORTH);

        centralPanel=new JPanel();
        centralPanel.setLayout(new GridLayout(4,4));

        JButton n1 ,n2,n3,n4,n5,n6,n7 ,n8,n9,n0,add,minus,mul,div,eq,c;
        n1= new JButton("1");
        n1.addActionListener(this);
        n2= new JButton("2");
        n2.addActionListener(this);
        n3= new JButton("3");
        n3.addActionListener(this);
        n4= new JButton("4");
        n4.addActionListener(this);
        n5= new JButton("5");
        n5.addActionListener(this);
```

ΣΥΝΕΧΕΙΑ

```
n6= new JButton("6");
n6.addActionListener(this);
n7= new JButton("7");
n7.addActionListener(this);
n8= new JButton("8");
n8.addActionListener(this);
n0= new JButton("0");
n8.addActionListener(this);
n9= new JButton("9");
n9.addActionListener(this);
add= new JButton("+");
add.addActionListener(this);
minus= new JButton("-");
minus.addActionListener(this);
mul= new JButton("*");
mul.addActionListener(this);
div= new JButton("/");
div.addActionListener(this);
c= new JButton("C");
c.addActionListener(this);
eq= new JButton("=");
eq.addActionListener(this);
}
```

```
centralPanel.add(n1);
centralPanel.add(n2);
centralPanel.add(n3);
centralPanel.add(add);
centralPanel.add(n4);
centralPanel.add(n5);
centralPanel.add(n6);
centralPanel.add(minus);
centralPanel.add(n7);
centralPanel.add(n8);
centralPanel.add(n9);
centralPanel.add(mul);
centralPanel.add(c);
centralPanel.add(n0);
centralPanel.add(eq);
centralPanel.add(div);

add(centralPanel, BorderLayout.CENTER);

setVisible(true);
}

public static void main(String [] args)
{ Main m=new Main();}

public void actionPerformed(ActionEvent e) {
    System.exit(0); }
```

Υλοποίηση actionPerformed

```
public void actionPerformed(ActionEvent e) {
    String buttonText = e.getActionCommand();
    if (buttonString.equals("1")|| buttonString.equals("2") ||
        buttonString.equals("3")|| buttonString.equals("4") ||
        buttonString.equals("5")|| buttonString.equals("6") ||
        buttonString.equals("7")|| buttonString.equals("8") ||
        buttonString.equals("9")|| buttonString.equals("0") ||
        buttonString.equals("+")||buttonString.equals("-")||
        buttonString.equals("*")||buttonString.equals("/"))
        {top.setText(top.getText()+buttonString);
         setVisible(true);}
    else if (buttonString.equals("C"))
        {top.setText("");
         setVisible(true);}
    else if (buttonString.equals("="))
        {top.setText("calculate the result");
         setVisible(true);}
    else
        System.exit(0); }
```

1	2	3	+	
4	5	6	-	
7	8	9	*	
C	0	=	/	

1588-69*3+47				
1	2	3	+	
4	5	6	-	
7	8	9	*	
C	0	=	/	

calculate the result				
1	2	3	+	
4	5	6	-	
7	8	9	*	
C	0	=	/	

1	2	3	+	
4	5	6	-	
7	8	9	*	
C	0	=	/	

JLIST

- Ένα στοιχείο που εμφανίζει μια λίστα αντικειμένων και επιτρέπει στο χρήστη να επιλέξει ένα ή περισσότερα αντικείμενα. Ένα ξεχωριστό μοντέλο, `ListModel`, διατηρεί τα περιεχόμενα της λίστας.
- Είναι εύκολο να εμφανίσετε έναν πίνακα ή ένα `Vector` αντικειμένων, χρησιμοποιώντας τον κατασκευαστή `JList` που δημιουργεί αυτόματα μία εμφάνιση `ListModel` μόνο για ανάγνωση για εσάς:

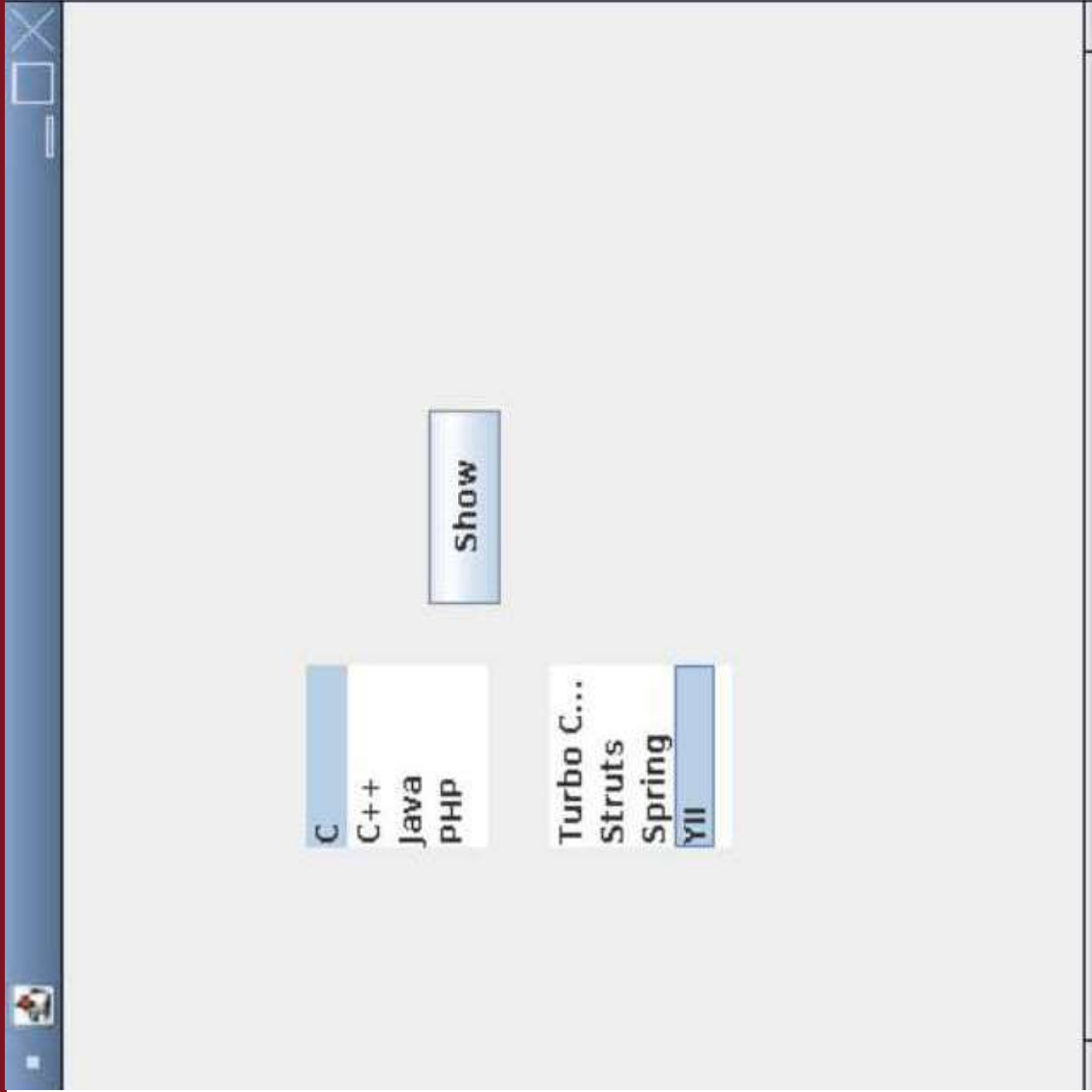
```
// Create a JList that displays strings from an array
String[] data = {"one", "two", "three", "four"};
JList<String> myList = new JList<String>(data);
```

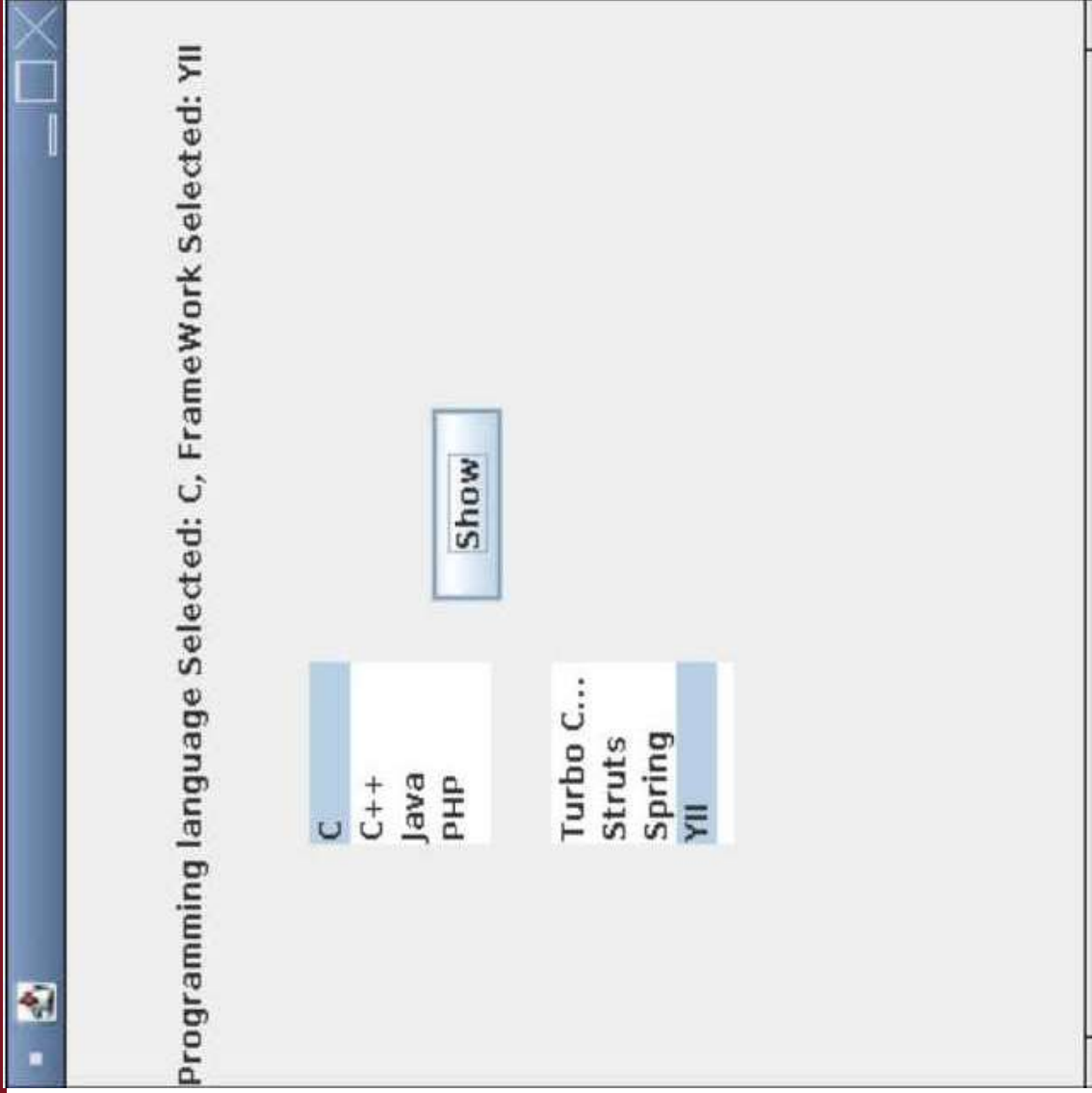
ΠΑΡΑΔΕΙΓΜΑ JList

```
import javax.swing.*;
import java.awt.event.*;
public class ListExample
{
    ListExample(){
        JFrame f= new JFrame();
        final JLabel label = new JLabel();
        label.setSize(500,100);
        JButton b=new JButton("Show");
        b.setBounds(200,150,80,30);
        final DefaultListModel<String> l1 = new DefaultListModel<>();
        l1.addElement("C");
        l1.addElement("C++");
        l1.addElement("Java");
        l1.addElement("PHP");
        final JList<String> list1 = new JList<>(l1);
        list1.setBounds(100,100,75,75);
        DefaultListModel<String> l2 = new DefaultListModel<>();
        l2.addElement("Turbo C++");
        l2.addElement("Struts");
        l2.addElement("Spring");
        l2.addElement("YII");
        final JList<String> list2 = new JList<>(l2);
        list2.setBounds(100,200,75,75);
        f.add(list1); f.add(list2); f.add(b); f.add(label);
        f.setSize(450,450);
        f.setLayout(null);
        f.setVisible(true);
    }
}

b.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String data = "";
        if (list1.getSelectedIndex() != -1) {
            data = "Programming language Selected: " +
                list1.getSelectedValue();
            label.setText(data);
        }
        if(list2.getSelectedIndex() != -1){
            data += " Framework Selected: ";
            for(Object frame :list2.getSelectedValues()){
                data += frame + " ";
            }
        }
        label.setText(data);
    }
});
}

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



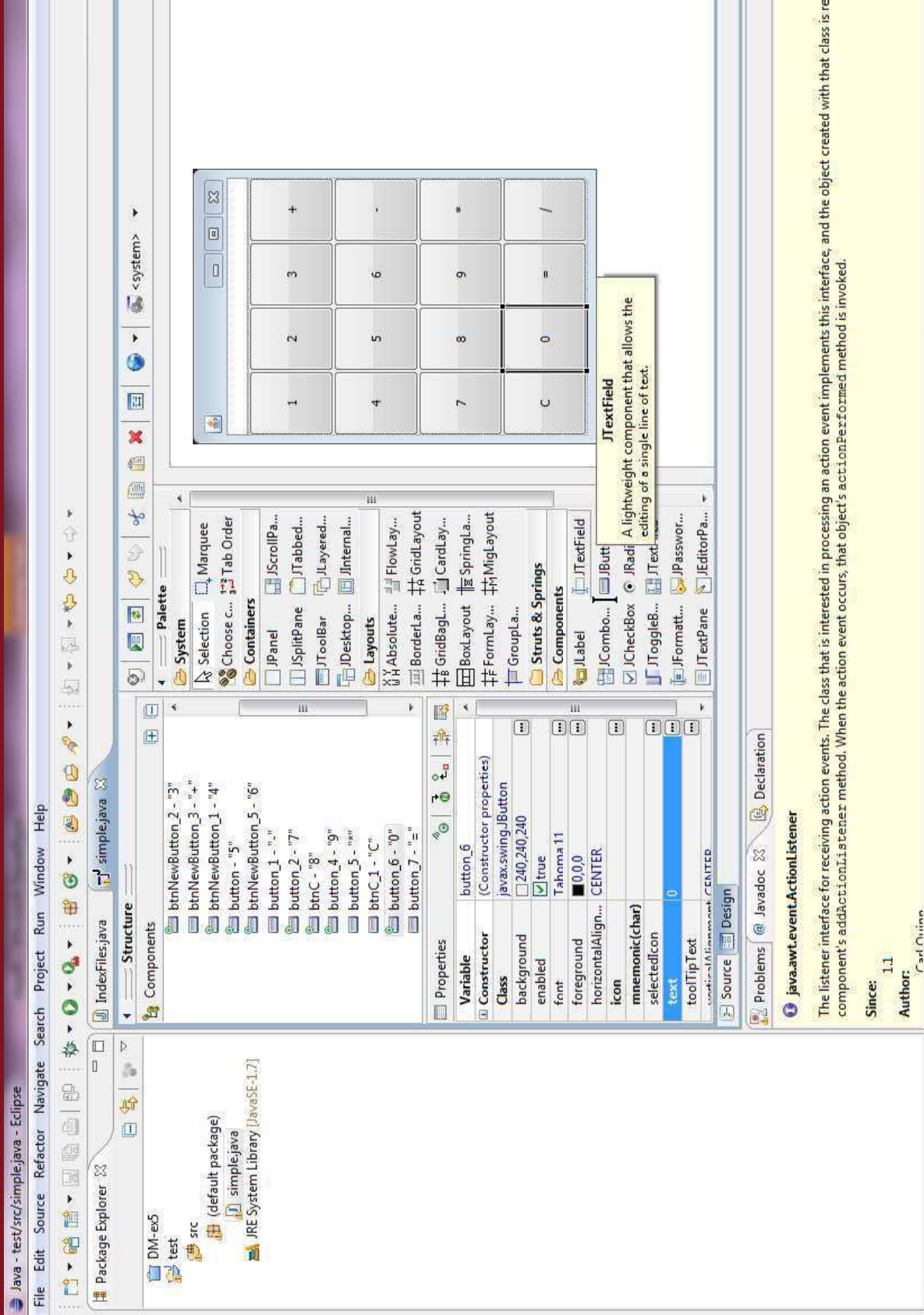


Eclipse

- Η eclipse (αλλά και άλλα IDEs) μας δίνει πολλά έτοιμα εργαλεία για την δημιουργία GUIs
- Εγκαταστήσετε το plug-in *Windows Builder Pro*
- **Παράδειγμα:** Δημιουργήστε μια αριθμομηχανή.

Εισαγωγή μίας διαφάνειας στο Eclipse

- Το Eclipse οργανώνει τον κώδικα σε **projects**.
- Οι **κλάσεις** στη συνέχεια προστίθενται μέσα στο project.
- Πρέπει να έχετε εγκαταστήσει το plugin **Windows Builder Pro**.
- Για να φτιάξετε ένα GUI
 - Αρχικά πρέπει να φτιάξετε ένα Java Project
 - Συνέχεια προσθέτετε στο project. Επιλέξετε **Other>WindowsBuilder>SWING>Application Window**.
- Στη συνέχεια θα έχετε ένα **μενού** από τα διάφορα components τα οποία μπορείτε να προσθέτετε στο στην εφαρμογή σας.
- Μπορείτε να δουλεύετε είτε με το **Design** είτε με τον **Source** κώδικα



The listener interface for receiving action events. The class that is interested in processing an action event implements this interface, and the object created with that class is registered with the component's addActionListener method. When the action event occurs, that object's actionPerformed method is invoked.

Since: 1.1
 Author: Carl Quinn

Δημιουργία κώδικα

- Τα IDEs μας επιτρέπουν να διαχωρίζουμε το **design** από τον **κώδικα**
 - Το πλεονέκτημα είναι ότι έχουμε ένα **WYSIWYG** interface με το οποίο μπορούμε να σχεδιάσουμε το GUI
 - Το μειονέκτημα είναι ότι δημιουργείται πολύς κώδικας **αυτόματα** ο οποίος δεν είναι πάντα όπως τον θέλουμε.
- Ο **διαχωρισμός** του σχεδιαστικού κομματιού από τις πράξεις που εκτελούν είναι γενικά μια καλή προγραμματιστική πρακτική.

Δημιουργία κώδικα

- Η δημιουργία ενός κουμπιού δημιουργεί αυτό τον κώδικα
- Αν πατήσουμε πάνω στο κουμπί (double-click) δημιουργείται ο ακροατής του κουμπιού αυτόματα ως μια ανώνυμη κλάση

```
JButton button_6 = new JButton("0");  
panel.add(button_6);
```

```
JButton button_6 = new JButton("0");  
button_6.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
    }  
});  
panel.add(button_6);
```

Δημιουργία κώδικα

- Η δημιουργία ενός κουμπιού δημιουργεί αυτό τον κώδικα
- Αν πατήσουμε πάνω στο κουμπί (double-click) δημιουργείται ο ακροατής του κουμπιού αυτόματα ως μια ανώνυμη κλάση
- Εμείς συμπληρώνουμε τον κώδικα

```
JButton button_6 = new JButton("0");  
panel.add(button_6);
```

```
JButton button_6 = new JButton("0");  
button_6.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        textField.setText(textField.getText() + "0");  
    }  
});  
panel.add(button_6);
```