

# Bölüm 22

## Java GUI Kütüphanesi

### 22.1 SWT GUI Kütüphanesi

Java programlama dili için bir başka GUI kütüphanesi daha vardır ve SWT(The Standard widget toolkit) olarak adlandırılır. SWT kütüphanesi ilk olarak IBM tarafından geliştirilmeye başlanmıştır. Şimdilerde IBM'in desteklediği açık kaynak kodlu bir projedir. SWT ağırsiklet araçların bir örneğidir. Altta çalışan işletim sisteminin GUI yaratmasına izin verir. SWT; çalışmak için yerel java arayüzünü kullanır. SWT'nin ana avantajları sürat, alışılmış görünüm ve hissiyattır. Swing kadar güçlü değildir ve işletim sisteminin kütüphanesi bağımlıdır.

### 22.2 SWING GUI Kütüphanesi

Swing kütüphanesi Sun Microsystems tarafından piyasaya sürülen resmi Java GUI aracıdır. Swing; %100 java ile yazılmıştır ve Java Foundation Classes(JFC) nin bir parçasıdır. Tam donanımlı masaüstü uygulamaları geliştirmek için kullanılan paketler bütünüdür. JFC; AWT ,Swing ,Ulaşılabilirlik,Java 2D ve Sürükle Bıraktan oluşur. 1997 yılında JDK 1.2 ile birlikte piyasaya sürülmüştür.

### 22.3 İlk Swing Programı

Bu bölümde, Swing kullanarak ilk programımızı oluşturacağız.Örnekler çok basit olacak. Örnek fonksiyonellikleri inceleyeceğiz.

İlk örnekte basit bir pencere yaratacağız.

## Alternative 1: JFrame'in Genişletilmesi

```
import javax.swing.JFrame;

public class Example extends JFrame {

    public Example() {

        setTitle("Simple example");
        setSize(300, 200);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String[] args) {

        Example ex = new Example();
        ex.setVisible(true);
    }
}
```

## Alternative 2: JFrame Sınıfının Kullanılması

```
import javax.swing.JFrame;

public class Example {

    public static void createGUI() {

        JFrame frame = new JFrame("Hello World");
        frame.setSize(300, 200);
        frame.setVisible(true);
    }

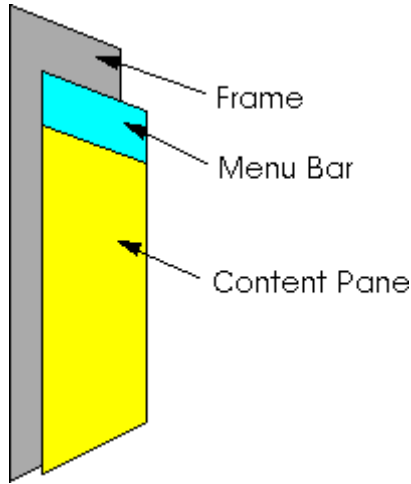
    public static void main(String[] args) {

        createGUI();
    }
}
```

## 22.3 JFrame

J ile başlayan bileşenlerin Swing isimlendirme kurallarına uyduğu Frame nesnesine JFrame denir.(JButton, JLabel)

JFrame 'Yüksek-Seviye' taşıyıcı olarak adlandırılır, yani JFrame görüntülenmek için kendini başka bir nesneye ekleme gereksinimi duymaz.



İçerik bölme; bütün text, buton ve grafiklerin konulduğu yerdir. Bu öğreticilerde İçerik bölmesiyle çalışacağız.

Bu örnekte gördüğünüz üzere JFrame birkaç bileşenden oluşur. Görüntü bileşenlerini eklemek için, içerik bölmesinde koyacağımız yerleri belirlemeliyiz.

İçerik bölmesi; kapsama hiyerarşisinde en üst sıradadır. Kapsama hiyerarşisi kök dizin olarak üst sıra taşıyıcısını(bu durumda JFrame) alan bileşenlerin oluşturduğu bir ağaç gibidir. Aşağı doğru indikçe, bileşenleri tutmak için JPanel gibi diğer taşıyıcıları kullanırız.

## 22.4 SwingUtilities.invokeLater()

**invokeLater()** metodu uygulamayı Swing Olay Sırası(Swing Event Queue)'na sokar.GUI'nin belirli durumlarda takılmasını engellemek için kullanılır.Bu konu daha sonraki konulardan biridir ve şimdi bunu düşünmemeliyiz.

```
SwingUtilities.invokeLater(new Runnable() {  
    public void run() {  
  
        // gui codes should be placed here.  
  
    }  
});
```

### Örnek:

```
import javax.swing.JFrame;  
import javax.swing.SwingUtilities;
```

```

public class Example extends JFrame {

    public Example() {
        setTitle("Simple example");
        setSize(300, 200);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String[] args) {

        SwingUtilities.invokeLater(new Runnable() {
            public void run() {

                Example ex = new Example();
                ex.setVisible(true);
            }
        });
    }
}

```

Kod kısa olmasına rağmen uygulama ekranı çok iş yapabilir. Yeniden boyutlandırılabilir, tam ekran yapılabilir veya minimize edilebilir. Bütün karmaşıklık programcının uygulamasında saklıdır.

```

import javax.swing.JFrame;
import javax.swing.SwingUtilities;

```

Swing sınıfları şu şekilde içe aktarılır.

```

public class Example extends JFrame {

```

Example sınıfı JFrame bileşenlerinden miras alır. JFrame üst seviye taşıyıcıdır ve diğer bileşenleri tutmak için kullanılır.

```

setTitle("Simple example");

```

Burada pencerenin boyutunu `setTitle()` metodunu kullanarak ayarladık.

```

setSize(300, 200);

```

Bu kod pencereyi 300px genişliğinde 200px uzunluğunda ayarlayacaktır.

```

setLocationRelativeTo(null);

```

Pencerenin ortasında bu satır olacaktır.

```
setDefaultCloseOperation(EXIT_ON_CLOSE);
```

Bu metod, sağ üstteki kapatma ikonuna basıldığında pencereyi kapatacaktır. Ön tanımlı ayarında hiç birşey yapmayacaktır.

```
SwingUtilities.invokeLater(new Runnable() {  
    public void run() {  
        Example ex = new Example();  
        ex.setVisible(true);  
    }  
});
```

Kodumuzun bir örneğini alalım ve ekranda gösterelim.



## 22.5 JButton

Bir sonraki örneğimizde bir butonumuz olacaktır. Butona seçtiğimizde uygulama kapanacaktır.

```
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JPanel;  
import javax.swing.SwingUtilities;  
  
public class Example extends JFrame {  
  
    // constructor  
  
    public Example() {  
        initUI();  
    }  
}
```

```

// initialize GUI components
public final void initUI() {

    JPanel panel = new JPanel();
    getContentPane().add(panel);
    panel.setLayout(null);

    JButton quitButton = new JButton("Quit");
    quitButton.setBounds(50, 60, 80, 30);

    //Button Event
    quitButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent event) {

            System.exit(0);

        }
    });

    panel.add(quitButton);

    setTitle("Quit button");
    setSize(300, 200);
    setLocationRelativeTo(null);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
}

// main method
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {

            Example ex = new Example();
            ex.setVisible(true);

        }
    });
}
}

```

JButton'u pencereye yerleştirdik. Buraya bir action listener ekleyeceğiz.

```

public Example() {
    initUI();
}

```

Bu ,GUI'yi metodun içinde yaratan iyi bir programlama uygulamasıdır

```

JPanel panel = new JPanel();
getContentPane().add(panel);

```

Jpanel bileşeni oluşturalım. Bu üretken hafif bir taşıyıcıdır. JFrame'i Jpanel'e ekleyelim

```

panel.setLayout(null);

```

Normal olarak JPanel FlowLayout yöneticisine sahiptir. Bu görünüm yöneticisi bileşenleri taşıyıcılara yerleştirmek için kullanılır. Eğer setLayout(null) metodunu

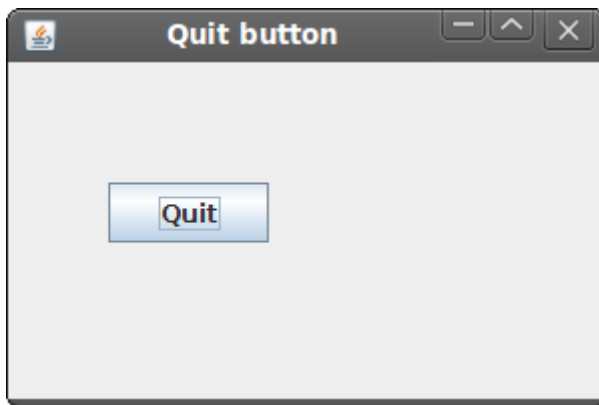
çağırırsak kendimiz bileşenleri yerleştirebiliriz. Bunun için `setBounds()` metodu kullanılır.

```
JButton quitButton = new JButton("Quit");
quitButton.setBounds(50, 60, 80, 30);
quitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) {
        System.exit(0);
    }
});
```

Burada bir buton yaratıyoruz. `setBounds()` metodunu çağırarak yerini ayarlıyoruz ve sonra bir action listener ekliyoruz. Butonu seçtiğimizde action listener çalışacaktır. Bu durumda butonu işaretlediğimizde uygulama kapanacaktır

```
panel.add(quitButton);
```

Quit butonunu göstermek için panele mutlaka eklemeliyiz.



## 22.6 Tooltip

Tooltip'ler yardım sisteminin birer parçalarıdır. Eğer fare imlecini bir objenin üstüne getirirsek Swing bunu dikdörtgen bir pencerede gösterir.

```
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;

public class Example extends JFrame {
```

```

public Example() {
    initUI();
}

public final void initUI() {

    JPanel panel = new JPanel();
    getContentPane().add(panel);

    panel.setLayout(null);
    panel.setToolTipText("A Panel container");

    JButton button = new JButton("Button");
    button.setBounds(100, 60, 100, 30);
    button.setToolTipText("A button component");

    panel.add(button);

    setTitle("Tooltip");
    setSize(300, 200);
    setLocationRelativeTo(null);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
}

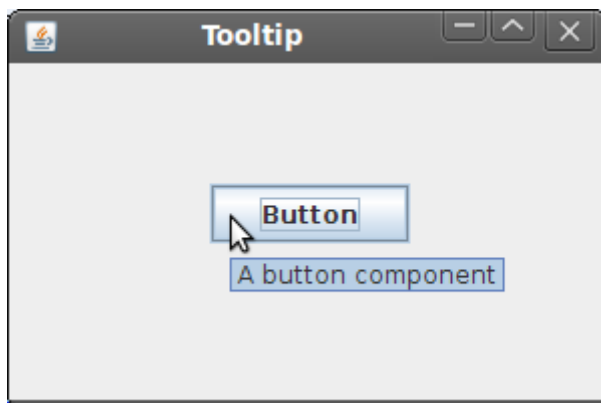
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            Example ex = new Example();
            ex.setVisible(true);
        }
    });
}
}

```

Bu örnekte frame ve buton için tooltip ayarlıyoruz.

```
panel.setToolTipText("A Panel container");
```

Tooltip'ı etkinleştirmek için `setToolTipText()` metodunu çağırıyoruz.





## 22.7 JLabel Bileşeni

JLabel yazı ve resim göstermek için kullanılan basit bir bileşendir. Girdi olaylarıyla etkileşim kurmaz.

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Font;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class MyLabel extends JFrame {

    public MyLabel() {

        setTitle("No Sleep");

        String lyrics = "<html>It's way too late to think of<br>" +
            "Someone I would call now<br>" +
            "And neon signs got tired<br>" +
            "Red eye flights help the stars out<br>" +
            "I'm safe in a corner<br>" +
            "Just hours before me<br></html>";

        JPanel panel = new JPanel();
        panel.setLayout(new BorderLayout(10, 10));

        JLabel label = new JLabel(lyrics);
        label.setFont(new Font("Georgia", Font.PLAIN, 14));
        label.setForeground(new Color(50, 50, 25));
        label.setBounds(105, 129, 46, 14);

        panel.add(label, BorderLayout.CENTER);
        add(panel);
        pack();

        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String[] args) {
        MyLabel mylabel = new MyLabel();
        mylabel.setVisible(true);
    }
}
```

Bu örnekte Cardigans'ın No Sleep Song isimli şarkısının sözlerini göstereceğiz.

JLabel bileşeninde HTML tagleri kullanılabilir. Satırları ayırmak için <br> kullanılır.

```
JPanel panel = new JPanel();
panel.setLayout(new BorderLayout(10, 10));
```

-- BorderLayout kullanmak için bir panel yaratıyoruz.

```
JLabel label = new JLabel(lyrics);
label.setFont(new Font("Georgia", Font.PLAIN, 14));
label.setForeground(new Color(50, 50, 25));
```

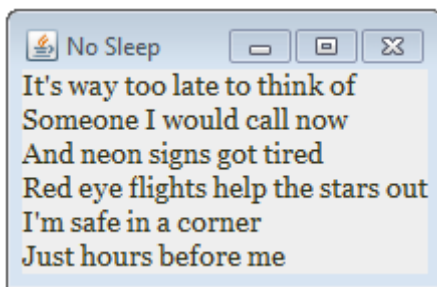
Burada bir label bileşeni oluşturuyoruz. Fontunu Georgia,14 px olarak ayarlıyoruz. Ön rengini de değiştiriyoruz.

```
panel.add(label, BorderLayout.CENTER);
```

Label'ı panelin merkezine koyuyoruz.

```
add(panel);  
pack();
```

Panel frame bileşenine eklendi. Bütün bileşenlerin görünebilmesi ve pencerenin boyutunun değiştirilebilmesi için metodunu `pack()` çağırıyoruz.



## 22.8 JCheckBox Bileşeni

JCheckBox iki durumu olan bir bileşendir. Bu durumlar Açık ve kapalıdır. İsmi olan bir kutudur. Eğer kutu işaretlenmişse, içinde bir tick(✓) ile gösterilir. Bir check box göster/gizle gibi durumlarda kullanılabilir.

```
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import javax.swing.JCheckBox;  
import javax.swing.JFrame;  
import javax.swing.SwingUtilities;  
  
public class CheckBoxExample extends JFrame {  
  
    public CheckBoxExample() {  
        initUI();  
    }  
  
    public final void initUI() {  
  
        final JCheckBox checkbox = new JCheckBox("Show Title", true);  
        checkbox.setBounds(37, 23, 127, 23);  
  
        getContentPane().setLayout(null);  
        getContentPane().add(checkbox);  
  
        checkbox.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent e) {  
  
                boolean state = checkbox.isSelected();  

```

```

        if (state) {
            setTitle("JCheckbox example");
        } else {
            setTitle("");
        }
    }
});

setSize(280, 200);
setTitle("JCheckBox example");
setLocationRelativeTo(null);
setResizable(false);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            CheckBoxExample ex = new CheckBoxExample();
            ex.setVisible(true);
        }
    });
}
}

```

Örneğimizde duruma göre pencere belirir veya saklanır.

```
JCheckBox checkbox = new JCheckBox("Show Title", true);
```

Burada checkbox için bir constructor metodumuz var. Yazıyı ve durumu sağlıyoruz.

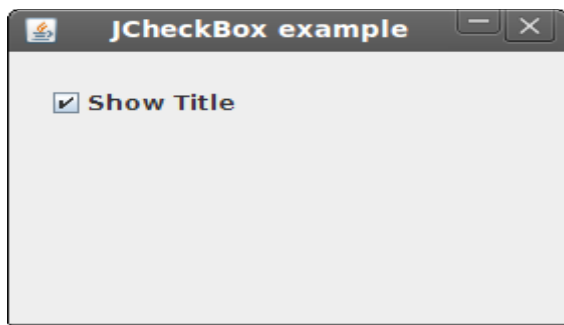
```

boolean state = checkbox.isSelected();

if (state) {
    setTitle("JCheckbox example");
} else {
    setTitle("");
}

```

Checkbox'un durum seçimi olduğunu biliyoruz. Checkbox'un durumuna bağlı olarak pencereyi gösterebilir veya gizleyebiliriz.



## 22.9 JSlider

JSlider kullanıcının belirli bir aralıkta sürükleme çubuğunu kaydırarak bir değere ulaşmasını sağlar. Aşağıdaki örnek, ses kontrolünü göstermektedir.

```
import java.awt.BorderLayout;
import java.awt.Dimension;
import javax.swing.BorderFactory;
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JSlider;
import javax.swing.SwingUtilities;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;

public class SliderExample extends JFrame {

    private JSlider slider;
    private JLabel label;

    public SliderExample() {
        initUI();
    }

    public final void initUI() {

        JPanel panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.X_AXIS));
        panel.setBorder(BorderFactory.createEmptyBorder(40, 40, 40, 40));
        panel.setLayout(new BorderLayout());

        panel.add(Box.createHorizontalGlue());
        slider = new JSlider(0, 150, 0);

        slider.setPreferredSize(new Dimension(150, 30));

        slider.addChangeListener(new ChangeListener() {
            public void stateChanged(ChangeEvent event) {
                int value = slider.getValue();
                if (value == 0) {
                    label.setText("<|");
                } else if (value > 0 && value <= 30) {
                    label.setText("<|");
                } else if (value > 30 && value < 80) {
                    label.setText("<|)");
                } else {
                    label.setText("<|))");
                }
            }
        });

        panel.add(slider);

        label = new JLabel("<|", JLabel.CENTER);
        panel.add(label);
        panel.add(Box.createHorizontalGlue());
        add(panel, BorderLayout.CENTER);

        pack();
    }
}
```

```

        setTitle("JSlider");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
    }

    public static void main(String[] args) {

        SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                SliderExample ex = new SliderExample();
                ex.setVisible(true);
            }
        });
    }
}

```

Bu örnekte; JSlider ve JLabel gösterdik. Sürükleme çubuğunu kaydırarak, label bileşenindeki yazıyı değiştirebiliriz.

```

panel.setLayout(new BoxLayout(panel, BoxLayout.X_AXIS));
panel.setBorder(BorderFactory.createEmptyBorder(40, 40, 40, 40));

```

Panelin etrafına 40px'lik bir çerçeve koyduk.

```

panel.add(Box.createHorizontalGlue());

```

JSlider'ın anormal boyutlara ulaşmasını engellemek için sağa ve sola ayarlanabilir boşluklar bıraktık.

```

slider = new JSlider(0, 150, 0);

```

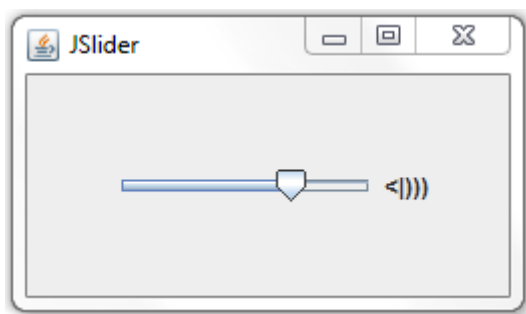
Bu JSlider constructor'udur. Parametreleri; minimum,maximum ve şu andaki değerdir.

```

slider.addChangeListener(new ChangeListener() {
    ...
});

```

Sürgüye ChangeListener ekliyoruz. Listener'ın içinde sürgü değerini ve yazıyı ayarlıyoruz.



## 22.10 JComboBox

Combobox butonu, alanı veya drop-down list bileşenlerini kombine eder. Kullanıcı isteğine göre drop-down list'ten değerini seçebilir. Eğer combobox değiştirilebilir yapılırsa, kullanıcının içine veri girebileceği bir alan oluşur.

```
import javax.swing.ImageIcon;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.border.EmptyBorder;
import java.awt.event.ItemListener;
import java.awt.event.ItemEvent;
import javax.swing.border.LineBorder;
import java.awt.Color;

public class ComboBoxExample extends JFrame {

    private JPanel contentPane;
    private JLabel display = null;
    private JComboBox combobox = null;

    final String[] authors = {"Leo Tolstoy", "John Galsworthy", "Honore de Balzac"};
    final String[] images = { "tolstoy.jpg", "galsworthy.jpg", "balzac.jpg", };

    public ComboBoxExample() {

        setTitle("JComboBox");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 315, 300);

        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        display = new JLabel();
        display.setBorder(new LineBorder(new Color(0, 0, 0)));
        display.setBounds(50, 26, 148, 162);
        contentPane.add(display);

        combobox = new JComboBox(authors);

        //Combobox Event declared here: itemStateChanged

        combobox.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {

                if (e.getStateChange() == ItemEvent.SELECTED) {
                    int index = combobox.getSelectedIndex();
                    display.setIcon(new ImageIcon(
                        ClassLoader.getResource(images[index])
                    ));
                }
            }
        });
        combobox.setSelectedIndex(-1);
        combobox.setBounds(50, 199, 148, 20);
        contentPane.add(combobox);
    }
}
```

```

    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                ComboBoxExample frame = new ComboBoxExample();
                frame.setVisible(true);
            }
        });
    }
}

```

Örneğimizde iki tane bileşen var. Label ve combobox. Combobox'un içinde 3 tane ünlü roman yazarı var. İsmi seçtiğimizde; resmi label'in içinde belirecek.

```
display = new JLabel();
```

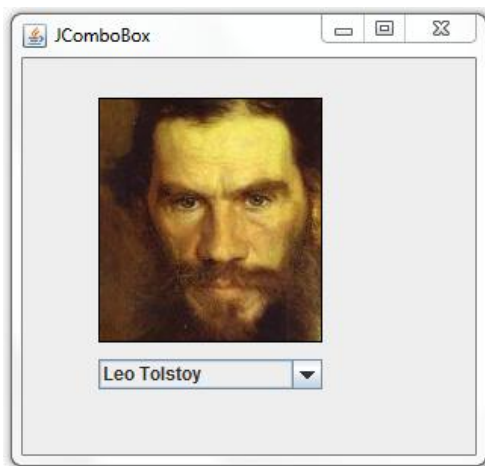
Gösterilecek alan basit bir JLabel'dir.

```
combobox = new JComboBox(authors);
combobox.setSelectedIndex(-1);
```

JComboBox'un constructor metodu yazarların isimlerini string dizisi olarak alır. Eğer setSelectedIndex() metodunu -1 olarak verirse hiç birşey seçilemeyecektir.

```
combobox.addItemListener( ... );
```

Combobox'a ItemListener ekliyoruz.Event handler'da combobox'da seçilmiş indexi ve label için uygun resmi alıyoruz. Seçilen isim; görüntülerin olduğu bir dizinin de indexidir.



## 22.11 JToggleButton

JToggleButton bir butondur ve iki durumu vardır. Basıldı veya basılmadı. Butona basarak bu durumu değiştirebilirsiniz. Bu fonksiyonelliğin kullanıldığı durumlar mevcuttur.

```
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.border.EmptyBorder;
import javax.swing.border.LineBorder;
import javax.swing.JToggleButton;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ToggleButtonSample extends JFrame {

    private JPanel contentPane;
    private JToggleButton redButton;
    private JToggleButton greenButton;
    private JToggleButton blueButton;
    private JPanel display;

    public ToggleButtonSample() {

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 302, 245);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        display = new JPanel();
        display.setBounds(155, 27, 121, 113);
        display.setBorder(LineBorder.createGrayLineBorder());
        display.setBackground(Color.black);
        contentPane.add(display);

        redButton = new JToggleButton("red");
        redButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                Color color = display.getBackground();
                int red = color.getRed();
                int green = color.getGreen();
                int blue = color.getBlue();

                if (e.getActionCommand().equals("red")) {
                    if (red == 0) {
                        red = 255;
                    } else {
                        red = 0;
                    }
                }
                Color setCol = new Color(red, green, blue);
                display.setBackground(setCol);
            }
        });
        redButton.setBounds(10, 27, 121, 23);
        contentPane.add(redButton);
    }
}
```



```

greenButton = new JToggleButton("green");
greenButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        Color color = display.getBackground();
        int red = color.getRed();
        int green = color.getGreen();
        int blue = color.getBlue();

        if (e.getActionCommand().equals("green")) {
            if (green == 0) {
                green = 255;
            } else {
                green = 0;
            }
        }
        Color setCol = new Color(red, green, blue);
        display.setBackground(setCol);

    }
});
greenButton.setBounds(10, 65, 121, 23);
contentPane.add(greenButton);

blueButton = new JToggleButton("blue");
blueButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        Color color = display.getBackground();
        int red = color.getRed();
        int green = color.getGreen();
        int blue = color.getBlue();

        if (e.getActionCommand().equals("blue")) {
            if (blue == 0) {
                blue = 255;
            } else {
                blue = 0;
            }
        }
        Color setCol = new Color(red, green, blue);
        display.setBackground(setCol);

    }
});
blueButton.setBounds(10, 100, 121, 23);
contentPane.add(blueButton);

}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            ToggleButtonSample frame = new ToggleButtonSample();
            frame.setVisible(true);
        }
    });
}
}

```

Örnekte bir panel ve üç tane buton vardır. Panelin arka plan rengini siyaha ayarlayacağız. Butonlar renk değerlerinin kırmızı mavi ve yeşil olanlarını tetikleyecek. Basılan butona göre arkaplan rengi değişecek.

```
Color color = display.getBackground();  
int red = color.getRed();  
int green = color.getGreen();  
int blue = color.getBlue();
```

`actionPerformed` metodunda o andaki kırmızı mavi ve yeşil rengi algılıyoruz.

```
if (e.getActionCommand().equals("red")) {  
    if (red == 0) {  
        red = 255;  
    } else {  
        red = 0;  
    }  
}
```

Hangi butona basıldığını ve o anki renk durumunu elde ediyoruz.

```
Color setCol = new Color(red, green, blue);  
display.setBackground(setCol);
```

Burada yeni bir renk oluşturuldu ve görüntü paneli o renkte görünmekte.

