

Bölüm 23

Gelişmiş Java GUI

23.1 JList Bileşeni

Jlist nesnelerin listesini gösteren bir bileşendir. Kullanıcının bir veya birden fazla seçim yapabilmesini sağlar.

```
import java.awt.Color;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JList;
import javax.swing.ListSelectionModel;
import javax.swing.event.ListSelectionListener;
import javax.swing.event.ListSelectionEvent;

public class JListExample extends JFrame {

    private JPanel contentPane;

    String[] listColorNames = { "black", "blue", "green", "yellow", "white" };
    Color[] listColorValues = { Color.BLACK, Color.BLUE, Color.GREEN, Color.YELLOW, Color.WHITE };

    public JListExample() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        final JList list = new JList(listColorNames);
        list.addListSelectionListener(new ListSelectionListener() {
            public void valueChanged(ListSelectionEvent e) {

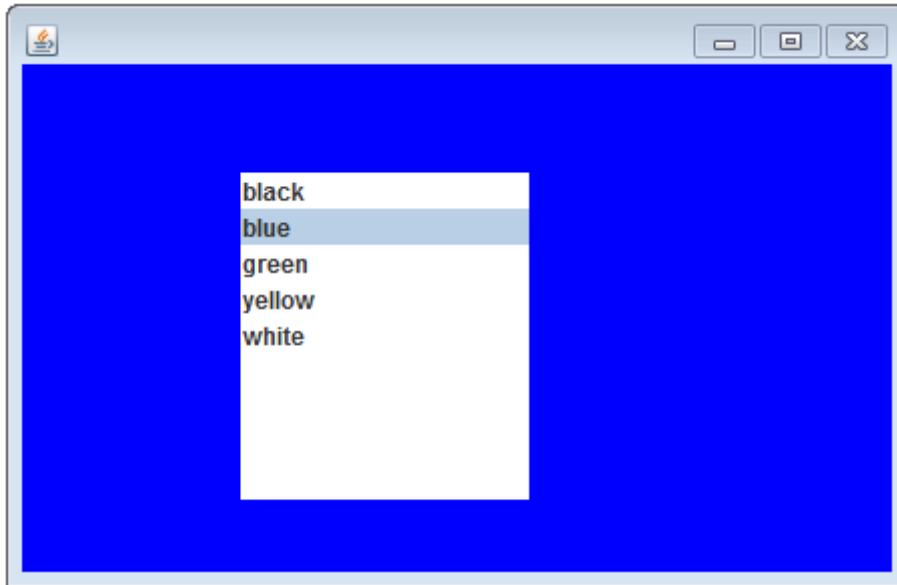
                contentPane.setBackground(listColorValues[list.getSelectedIndex()]);

            }
        });
        list.setSelectionModel(ListSelectionModel.SINGLE_SELECTION);
        list.setSelectedIndex(0);
        list.setBounds(109, 54, 144, 163);
        contentPane.add(list);
    }
}
```

```

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

```



23.2 JTextArea Bileşeni

JTextArea yazıların gösterildiği çok satırlı, hafifsiklet bir bileşendir. Bileşen sayfanın aşağı yukarı kaydırılmasını desteklemez. Bunun için JScrollPane kullanırız.

```

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JOptionPane;
import javax.swing.JTextArea;
import javax.swing.JButton;
import javax.swing.JScrollPane;
import javax.swing.SwingUtilities;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class JTextAreaExample extends JFrame {

    private JPanel contentPane;

    public JTextAreaExample() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("Independence hymn");
        setBounds(100, 100, 450, 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
    }
}

```

```

        contentPane.setLayout(null);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(56, 11, 300, 177);
        contentPane.add(scrollPane);

        final JTextArea textArea = new JTextArea();
        textArea.setWrapStyleWord(true);
        textArea.setLineWrap(true);
        scrollPane.setViewportViewView(textArea);

        JButton btnNewButton = new JButton("Show Me Text");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {

                JOptionPane.showMessageDialog(null, textArea.getText());
            }
        });
        btnNewButton.setBounds(156, 216, 141, 23);
        contentPane.add(btnNewButton);
    }

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

```

Bu örnek basit bir JTextArea bileşenini temsil eder.

```
JTextArea textArea = new JTextArea();
```

JTextArea bileşeninin constructor metodu aşağıdaki gibidir.

```
textArea.setLineWrap(true);
```

Eğer yazılar sığmayacaksa, aşağıdan devam etmesini sağlar.

```
textArea.setWrapStyleWord(true);
```

Burada satırın ne şekilde alt taraftan devam edeceği özelleştirilir. Bizim durumumuzda satırlar kelimelerin bittiği boşluklardan devam edecektir.

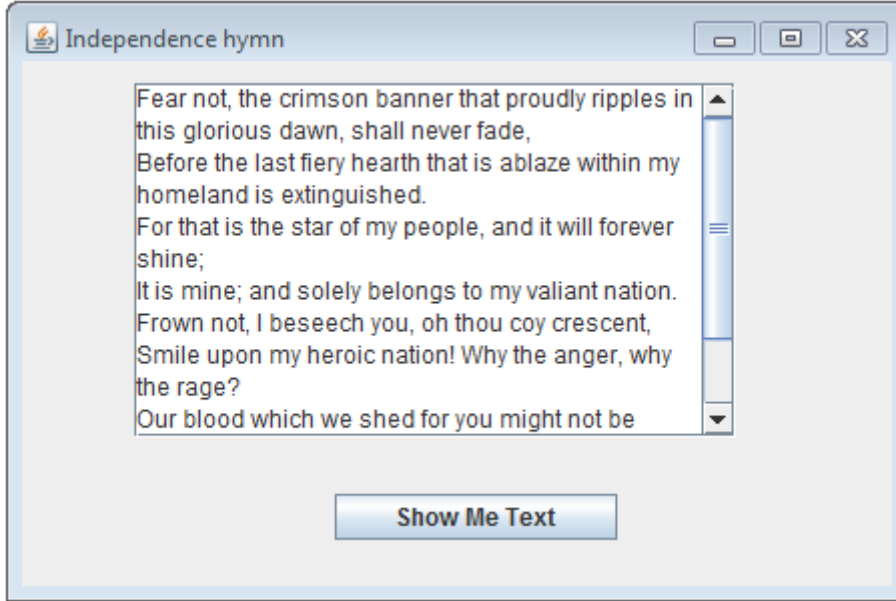
```

JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(56, 11, 300, 177);
contentPane.add(scrollPane);

final JTextArea textArea = new JTextArea();
textArea.setWrapStyleWord(true);
textArea.setLineWrap(true);
scrollPane.setViewportViewView(textArea);

```

Yazının kaydırılabilmesi için JTextArea bileşenini JScrollPane bileşeninin içine koyarız.



23.3 JEditorPane Bileşeni

JEditorPane farklı yazı formatları gösterebilen bir textArea'dır. Öntanımlı olarak HTML ve RTF destekler.

Pratikte, JEditorPane HTML göstermek için kullanılır. JEditorPane artı olarak RTF de desteklemektedir ancak sınırlıdır. JEditorPane içeriğini aşağıdaki şekilde belirleyebilirsiniz.

- URL nesnesini veya string'ini JEditorPane constructor metoduna göndererek
- setPage() metodunu kullanarak JEditorPane içeriğini çalışma zamanında ayarlayarak
- içeriği setText() metoduyla string olarak göndererek

JEditorPane'ı yalnız okunabilir kılmak için setEditable(false) metodu kullanılır. HTML dökümanları gösterilirken kullanıcının tıkladığı HTML linkleri JEditorPane tarafından farkedilebilir. Fare tıklamalarını kontrol edebilmek için **HyperlinkListener** event'i kullanılmalıdır.

Bu örnekte JEditor Pane kullanarak herhangi bir sayfayı açmak için basit bir web tarayıcısı yapacağız. Artı olarak JButton ve JTextField bileşenleri de kullanılacak.

```

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.JEditorPane;
import javax.swing.JScrollPane;
import javax.swing.SwingUtilities;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class JEditorPaneExample extends JFrame {

    private JPanel contentPane;
    private JTextField textField;

    public JEditorPaneExample() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 850, 517);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblUrl = new JLabel("URL");
        lblUrl.setBounds(10, 11, 36, 14);
        contentPane.add(lblUrl);

        textField = new JTextField();
        textField.setBounds(56, 8, 688, 20);
        contentPane.add(textField);
        textField.setColumns(10);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(10, 39, 800, 416);
        contentPane.add(scrollPane);

        final JEditorPane editorPane = new JEditorPane();
        scrollPane.setViewportView(editorPane);
        JButton btnNewButton = new JButton("GO");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {

                try {
                    editorPane.setPage(textField.getText());
                } catch (Exception ex) {
                    ex.printStackTrace();
                }

            }
        });
        btnNewButton.setBounds(754, 7, 56, 23);
        contentPane.add(btnNewButton);
    }

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

```



23.4 Swing Dialog Pencereleeri

Diyalog pencereleri veya diyaloglar modern GUI uygulamalarının olmazsa olmazlarıdır. Diyalog iki veya daha fazla kişi arasında geçen konuşma olarak tanımlanmaktadır. Bilgisayar uygulamasında ise; uygulamayla konuşmak için kullanılır. Diyalog; veri girişi, verinin değiştirilmesi, uygulama ayarlarının değiştirilmesi vb.'nde kullanılır. Diyaloglar program ve kullanıcı arasındaki iletişimin kilit noktasıdır.

Java Swing de iki tür diyalog üretilebilir.

- özel diyalog
- standart diyalog

Özel diyaloglar programcı tarafından üretilir. JDialog sınıfını baz alır.

Standart diyaloglar Swing'de bulunan önceden tanımlı olanlardır. Örnek olarak `MessageDialog`, `InputDialog`, `JColorChooser` `JFileChooser` verilebilir. Bu diyaloglar metni gösterme, input alma, dosyaları yükleme ve kaydetme gibi genel programlama

görevleri için kullanılır. Bunlar programcıya zaman kazandırır ve bazı standart davranışların kullanımını arttırır.

İki tür basit diyalog tipi vardır. Kipli ve kipsiz

- **Kipli diyaloglar** açık olan pencerenin arkasındaki pencerelere müdahaleye izin vermez. Dosya açma penceresi buna iyi bir örnektir. Bir dosya açarken başka bir işleme izin verilmez.
- **Kipsiz diyaloglar** diğer pencerelere erişime izin verir. Arama diyalogu buna bir örnektir.

23.5 JFileChooser

JFileChooser sistemden dosya seçmek için kullanılan standart bir diyalogdur.

```
import java.awt.EventQueue;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.filechooser.FileFilter;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.io.File;
import javax.swing.JEditorPane;

public class JFileChooserExample extends JFrame {

    private JPanel contentPane;

    public JFileChooserExample() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
    }
}
```

```

final JEditorPane editorPane = new JEditorPane();
editorPane.setBounds(20, 50, 387, 187);
contentPane.add(editorPane);

JButton btnNewButton = new JButton("Open File");
btnNewButton.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        JFileChooser fileopen = new JFileChooser();
        FileFilter filter = new FileNameExtensionFilter("txt files", "txt");
        fileopen.addChoosableFileFilter(filter);

        int answer = fileopen.showDialog(null, "Open file");

        if (answer == JFileChooser.APPROVE_OPTION) {
            File file = fileopen.getSelectedFile();
            try {
                editorPane.setPage("File: //" + file);
            } catch (Exception e1) {
                e1.printStackTrace();
            }
        }

        btnNewButton.setBounds(10, 11, 89, 23);
        contentPane.add(btnNewButton);
    }
});

public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                JFileChooserExample frame = new JFileChooserExample();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
}

```

Bu örnek yazı alanı bileşenine dosya içeriklerini koyabilmek için dosya seçme diyalogunun nasıl kullanılacağını açıklamaktadır.

```
JFileChooser fileopen = new JFileChooser();
```

Dosya seçme diyalogunun constructor metodu.

```
FileFilter filter = new FileNameExtensionFilter("txt files", "txt");
fileopen.addChoosableFileFilter(filter);
```

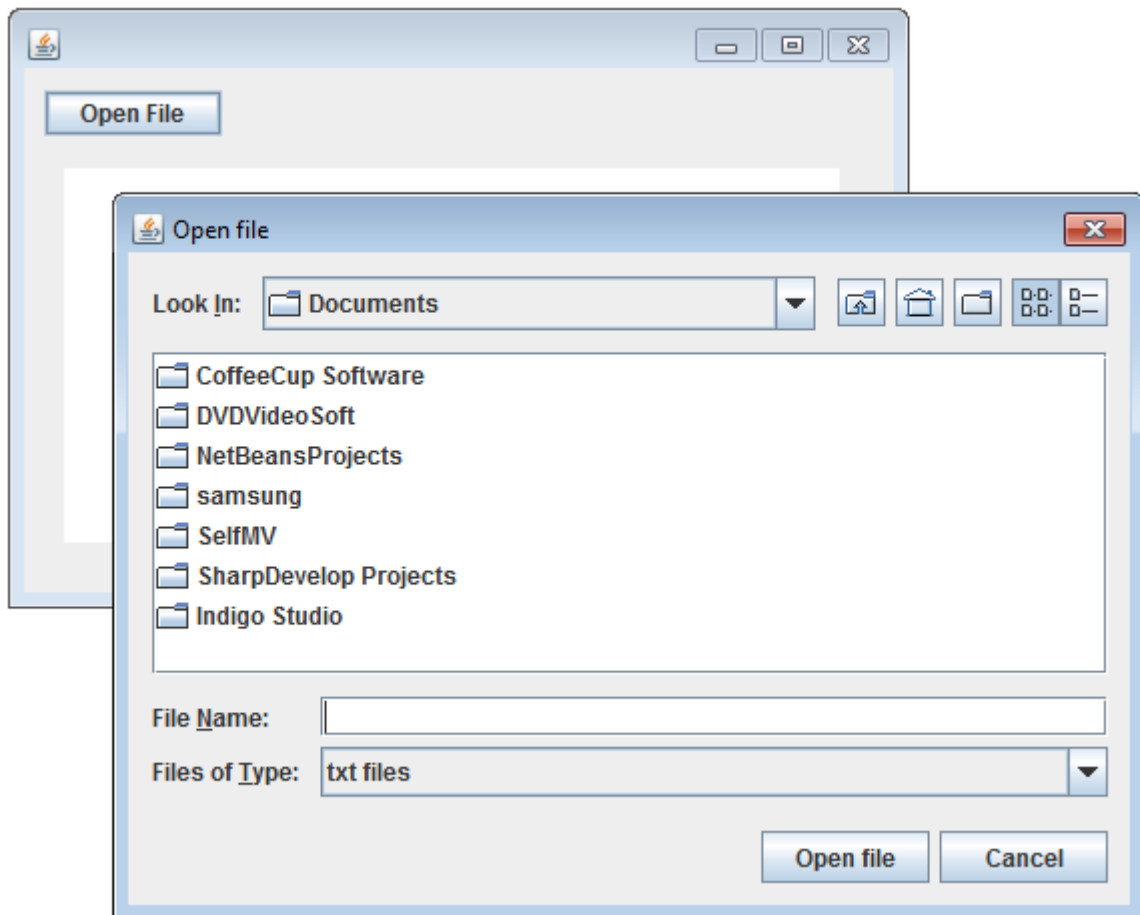
Burada bir filtre oluşturuyoruz. Şu durumda txt uzantılı text dosyalarına erişeceğiz. Öntanımlı olarak “Bütün dosyalar” seçeneği vardır.

```
int answer = fileopen.showDialog(null, "Open file");
```


Burada dosya seçme diyalogunu gösteriyoruz. Dosyayı aç butonuna basınca geri dönen değer `JFileChooser.APPROVE_OPTION`. değeridir.

```
if (ret == JFileChooser.APPROVE_OPTION) {  
    File file = fileopen.getSelectedFile();  
    try {  
        editorPane.setPage("File:///"+ file);  
    } catch (Exception e1) {  
        e1.printStackTrace();  
    }  
}
```

Burada seçilen dosyanın ismini alıyoruz. Dosyanın içeriklerini okuyoruz ve ismini editorPane'e atıyoruz.



23.6 JColorChooser

JcolorChooser renk seçmek için kullanılan bir diyalogdur.

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

public class JColorChooserExample extends JFrame {

    private JPanel contentPane;

    public JColorChooserExample() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        final JPanel panel = new JPanel();
        panel.setBorder(new LineBorder(new Color(0, 0, 0)));
        panel.setBounds(67, 74, 220, 145);
        contentPane.add(panel);

        JButton btnNewButton = new JButton("Choose Color");
        btnNewButton.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                Color color = JColorChooser.showDialog(null, "Choose Color", Color.white);
                panel.setBackground(color);

            }
        });

        btnNewButton.setBounds(10, 11, 111, 23);
        contentPane.add(btnNewButton);

    }

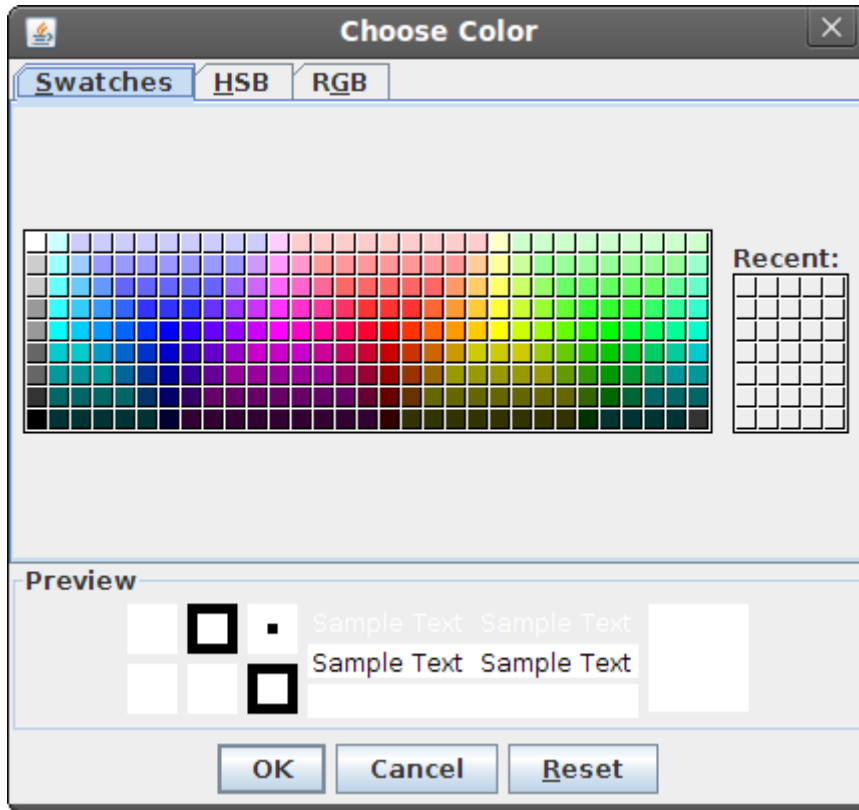
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                JColorChooserExample frame = new JColorChooserExample();
                frame.setVisible(true);
            }
        });
    }
}
```

```
}
```

Bu örnekte bir panelimiz var. Renk seçme diyalogunu kullanarak panelden bir renk seçip arka plan rengini değiştireceğiz.

```
Color color = JColorChooser.showDialog(null, "Choose Color", Color.white);  
panel.setBackground(color);
```

Bu kod renk seçme diyalogunu gösteriyor. `showDialog()` metodu seçilen rengi geri döndürür. Seçilen yeni renge göre arka planı değiştiriyoruz.



23.7 Java Swing Menüleri

Menubar GUI uygulamasında en çok görülen bileşendir. Bunlara farklı menülere yerleştirilmiş komutlar grubu da denebilir.

Java Swing'te, menubar uygulamak için, 3 adet nesne kullanırız. JMenuBar, JMenu ve JMenuItem.

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.SwingUtilities;

public class Example extends JFrame {

    public Example() {
        JMenuBar menubar = new JMenuBar();
        ImageIcon icon = new ImageIcon(getClass().getResource("exit.png"));

        JMenu filemenu = new JMenu("File");

        JMenuItem eMenuItem = new JMenuItem("Exit", icon);
        eMenuItem.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent event) {
                System.exit(0);
            }
        });

        filemenu.add(eMenuItem);
        menubar.add(filemenu);

        setJMenuBar(menubar);

        setTitle("Simple menu");
        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);
            }
        });
    }
}
```

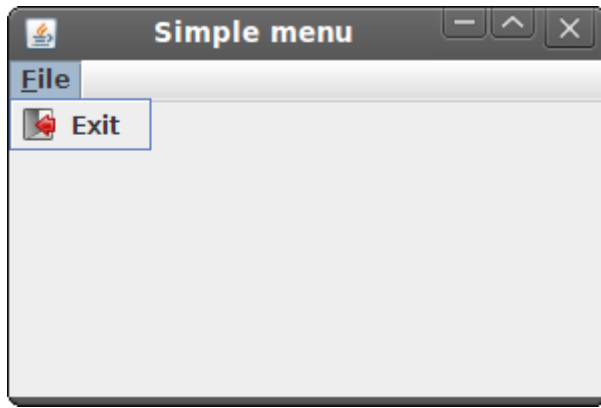
Örneğimizde bir elemanı olan bir menü göstereceğiz Çıkış seçeneğini seçerek uygulamayı kapatacağız.

```
JMenuBar menubar = new JMenuBar();
```

Burada menüyü oluşturduk

```
ImageIcon icon = new ImageIcon(getClass().getResource("exit.png"));
```

Menüde bir ikon gösterdik.



23.8 JTable

JTable çok boyutlu dizilerde tutulan verileri göstermek için kullanılır. Oyuncuları,müşterileri,farklı sensörlerden gelen verileri göstermek için uygundur. Öntanımlı modelinde veriler değiştirilebilir hücrelerin olduğu bir tabloya koyulur. Excel'de olduğu gibi bu hücreler eşit boyutludur.

Tabloyu doldurmak için veri dizilerine ihtiyac duyulur.Tablonun başlığı basit bir string dizisidir. Veriler ise 2 boyutlu bir nesne dizisidir.

```
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.border.EmptyBorder;
import javax.swing.JTable;
import javax.swing.JScrollPane;

public class JTableExample extends JFrame {

    private JPanel contentPane;
    private JTable table;

    // The data used as the titles for the table.
    String[] title = {"No.", "Country", "Player", "Position"};

    // The data used in the table, placed as a multi-dimensional array.
    Object[][] playerdata = {
        {4, "United States", "Sterling Davis", "Forward"},
        {6, "Germany", "Moritz Wohlers", "Forward/Centre"},
        {7, "United Kingdom", "Ross Hutton", "Centre"},
        {8, "Belgium", "Hugo Sterk", "Guard"},
        {10, "United Kingdom", "Andy Pearson", "Forward"},
        {11, "United States", "Robert Yanders", "Guard"},
        {12, "United Kingdom", "Graham Hunter", "Guard"},
        {14, "United Kingdom", "Julius Joseph", "Guard/Forward"},
        {15, "United Kingdom", "Gareth Murray", "Forward"},
        {21, "United States", "Maurice Hampton", "Guard"};
    }
```

```

public JTableExample() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 300);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JScrollPane scrollPane = new JScrollPane();
    scrollPane.setBounds(35, 45, 344, 119);
    contentPane.add(scrollPane);

    table = new JTable(playerdata, title);
    scrollPane.setViewportView(table);
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            try {
                JTableExample frame = new JTableExample();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
}

```

JTable'dan kolay bir şekilde instance alınabilir. 2 boyutlu diziyi ve başlığı tabloya göndeririz.

```
JTable table = new JTable(playerdata, title);
```

Kaydırma çubuğunu iki sebepten ötürü kullanırız. ilki aşağı yukarı erişim, ikincisi iste kaydırma çubuğu olmadan başlıkların gözükmemesidir.