

JAVA: ERRONKA 3

Aurkibidea

1. Interfaze-grafikoak
 - a. Sarrera
 - b. Elementuak
 - c. Elementuak antolatu
 - d. JDialog
2. Datu-baseak
 - a. Konexioa
 - b. Kotsultak

1. Interfaze-grafikoak

Java Swing Java plataforman sartzen diren erabiltzaile-interfaze grafikoko liburutegien multzoa da. Sun Microsystems-ek (orain Oracle) garatu zuen, AWTren (Abstract Window Toolkit) alternatiba bat eskaintzeko, aplikazio moderno askok behar zuten pertsonalizatzeko gaitasunik ez zuena.

Java Swing-en ezaugarri nagusiak hauek dira:

- **Erabat pertsonalizagarria:** Swing-ek aplikazioaren beharren arabera pertsonalizatu daitezkeen osagai ugari eskaintzen ditu.
- **Multiplataforma:** Swing aplikazioak bateragarriak dira hainbat sistema eragilerekin, hala nola Windows, Linux eta macOS.
- **Modularra:** Swing aplikazioak egitura modular bat erabiliz eraiki daitezke, eta horrek esan nahi du osagaiak beharrezkoa den heinean gehitu eta ezabatu daitezkeela.
- **Oso bisuala:** Swing-ek oso bisualak eta erakargarriak diren erabiltzaile-interfazeak sortzeko aukera ematen du, ikonoak, grafikoak, animazioak eta efektu bisualak erabiliz.

JFrame → [dokumentazioa](#)

Java Swing-en, JFrame-ak kontenedore bezala erabili daiteke, bertan beste konponente batzuk sartzeko aukera dago (JButton, JLabel, JTextField, ...)

Nola erabili:

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

```
public class OinarrizkoLehioa{  
    public static void main(String[] args) {  
        JFrame lehio = new JFrame("Lehenengo Interfaze-Grafikoa");  
        lehio.setSize(400, 300);  
        lehio.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        lehio.setVisible(true);  
    }  
}
```

JButton → [dokumentazioa](#)

JButton Swing liburutegiko konponente bat da, erabiltzaile-interakzioei erantzuteko

Adib:

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

```
public class VentanaConBoton {  
    public static void main(String[] args) {  
        JFrame ventana = new JFrame("Mi ventana");  
        ventana.setSize(400, 300);  
        ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        JButton boton = new JButton("Haz clic");  
        boton.addActionListener(e -> System.out.println("¡Hola mundo!")); → interakzioa ahalbidetzen du  
        ventana.add(boton);  
        ventana.setVisible(true);  
    }  
}
```

JPanel → Dokumentazioa

JPanel Swing liburutegiko konponente bat da, beste konponente batzuk barnean izan ditzakena.

Adibidea:

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

public class JPanelAdib {
    public static void main(String[] args) {
        // Sortu botoiak
        JButton button1 = new JButton("Botoi 1");
        JButton button2 = new JButton("Botoi 2");
        JButton button3 = new JButton("Botoi 3");

        // Sortu panela eta gehitu botoiak
        JPanel panel = new JPanel();
        panel.add(button1);
        panel.add(button2);
        panel.add(button3);

        // Sortu JFrame bat eta gehitu panela
        JFrame frame = new JFrame("JPanel adibidea");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().add(panel);
        frame.pack();
        frame.setVisible(true);
    }
}
```

JTextArea → [Dokumentazioa](#) - JTextField → [Dokumentazioa](#)

JTextField Java Swing klase bat da, erabiltzaile-interfaze grafiko batean (GUI) testu-eremuaren osagai bat sortzeko erabiltzen dena.

Osagai horrek aukera ematen dio erabiltzaileari testua sartzeko, lerro bakarra edo hainbat lerro, konfigurazioaren arabera. Oso erabilgarria da erabiltzailearen datuak sartzeko aplikazioetan, hala nola inprimaketan eta bilaketa-sarreretan, besteak beste.

Adib:

```
JTextFiel jtf = new JTextField ("Idatzi nahi duzuna");
```

Elementuan Idatzitakoa jasotzeko → `getText();`

Elementuan zeozer idazteko → `setText();`

JLabel → [Dokumentazioa](#)

JLabel erabiltzaile-interfaze grafiko batean (GUI) etiketa-osagai bat sortzeko erabiltzen den Java Swing klase bat da.

Osagai hau edukiontzi baten barruko posizio jakin batean testua edo irudi bat erakusteko erabiltzen da.

Etiketak oso erabilgarriak dira informazio deskribatzailea bistaratzea eskatzen duten aplikazioetan, hala nola datuen sarrera-eremu baten izena, aplikazioaren atal baten izenburua, etab.

Menu zabalgarria → [Dokumentazioa](#)

Java swing klase bat da, aukeren goitibeherako zerrenda bat sortzeko aukera ematen duena.

// behar dugu edukinen string array bat

```
JComboBox <String> jcb = new JComboBox<>(array);
```

//definitu defektuz aukeratuta egongo den elementua

```
jcb.setSelectedIndex(0);
```

//gehitu leihora

//jakiteko aukeratutako elementua

```
jcb.addActionListener (...)
```



MenuBar → [Dokumentazioa](#)

Edozein programak duen goiko menu barra sortzeko liburutegia da.

Beharrezkoak dira ondorengo elementuak: **JMenuBar**, **JMenu** eta **JMenuItem**

JMenuBar → barra sortuko du

JMenu → Menu bloke bakoitza

JItem → Menu bakoitzaren barruan egongo diren elementuak

Adib:

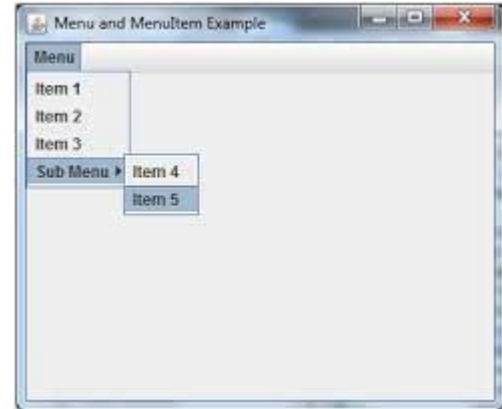
```
JMenu ma = new JMenu ("Fitxategia");
```

```
JMenuItem jmi1 = new JMenuItem ("Item1");
```

```
JMenuBar jmb = new JMenuBar ();
```

```
jmb.add(ma);
```

```
JFrame.setJMenuBar(jmb);
```



Beste elementu batzuk

JColorChooser	JSpinner
JList	JSplitPane
JPasswordField	JTabbedPane
JProgressbar	JTable
JSeparator	<u>JScrollPane</u>

Elementuak antolatu (JFrame eta JPanel) → [Dokumentazioa](#)

Hauek dira erabilienak:

- BorderLayout → lehenetsitako diseinua. Osagarriak 5 lekutan antolatzen dira. NORTH, SOUTH, EAST, WEST eta CENTER.
 - Pausuak:
 - `JFrame.setLayout (new BorderLayout());`
 - `JFrame.add("osagarria", BorderLayout.NORTH);`
- GridLayout → kuadrikulatan antolatzen ditu osagarriak, non gelaxka bakoitzak tamaina berdina dauka.
 - Pausuak:
 - `JFrame.setLayout (new GridLayout(2,2));`
 - `JFrame.add("osagarria");`
- FlowLayout → osagarriak horizaontalki antolatuko dira, eta lerroan ez direnean sartzen hurrengora pasatuko dira.
 - Pausuak:
 - `JFrame.setLayout (new FlowLayout());`
 - `JFrame.add("osagarria");`
- BoxLayout → osagarriak lerroetan edo zutabetan antolatzen ditu, osagarri guztiak tamaina berdina izango dute.
 - Pausuak:
 - `JFrame.setLayout (new BoxLayout(JFrame, BoxLayout.Y_AXIS));`
 - `JFrame.add("osagarria")`

JDialog → [Dokumentazioa](#)

JDialog bigarren mailako leiho bat da, informazio gehigarria erakusteko edo erabiltzaileari datuak eskatzeko erabiltzen dena, leiho nagusi bat (JFrame bat, esaterako) irekita dagoen bitartean.

JDialogak modalak izaten dira, eta horrek esan nahi du erabiltzaileak JDialog itxi behar duela leiho nagusiarekin elkarreragin ahal izan aurretik.

Adib:

```
import javax.swing.JDialog;
JDialog dialog = new JDialog(JFrame"izena");
//gehitu nahi ditugun elementuak, JFrame-n egiten den bezala
dialog.setVisible(true);
dialog.dispose(); //itxiko du leihoa
```

JFrame & JDialog: Extends

Herentzia erabiltzeak, JFramek Javan zabalduko duen klase bat sortzeko, aplikazio batean leihoak sortzeko eta administratzeko prozesua sinplifikatu dezake, eta, era berean, leihoaren itxura eta funtzionaltasuna pertsonalizatzea ahalbidetzen du, aplikazioaren behar espezifikoaren arabera.

JFrame & JDialog: Extends Adibidea

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

public class LeihioPertsonalizatua extends JFrame {

    private JButton boton;
    private JPanel panel;

    public VentanaPersonalizada() {
        setTitle("Leihio Pertsonalizatua");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        boton = new JButton("Klikat hemen");
        panel = new JPanel();
        panel.add(boton);

        add(panel);
    }

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

Ejer persona

Adib. organizar elementos y menuBar

Ejer calculadora

Ejer extens