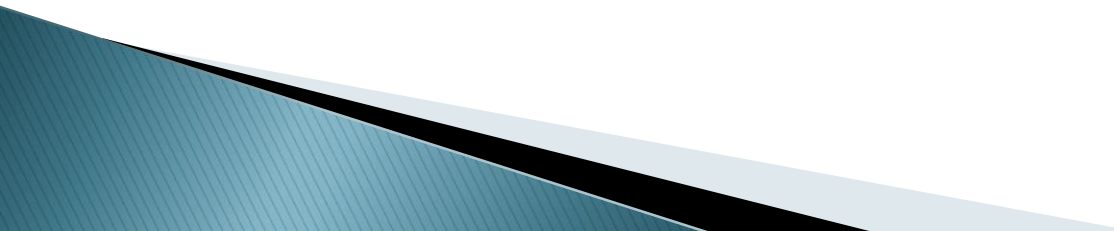


Interfaze Grafikoak

SOFTWARE INGENIARITZA

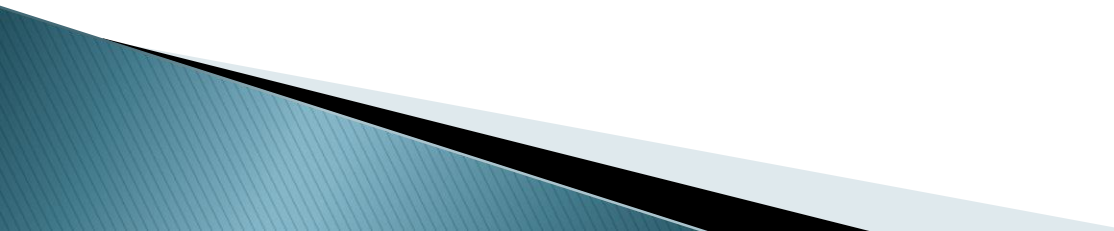
EDUKIAK

- ▶ Motivazioa eta helburuak
 - ▶ GUI osagaiak
 - ▶ Objektu grafikoen hierarkiak (AWT, Swing)
 - ▶ Layout kudeatzaileak
- 

Motibazioa

- ▶ Edozein programazio lengoaia modernoak Erabiltzaile Interfaze Grafikoak (GUI – Grafical User Interface) garatzeko erremintak eskeintzen ditu.
- ▶ Javak hurrengoa egiten du posible:
 - GUI-en diseinu eta programazio azkar eta sinplea.
 - AWT (Abstract Window Toolkit) klase paketea.
 - Swing klase paketea: AWT-ren eboluzioa, klase eta malgutasun gehiago eskeintzen ditu.

Helburuak

- ▶ GUIak eraikitzeko erabiltzen diren Java klaseen hierarkiaren diseinua ulertzea.
 - ▶ Gertaeren kudeaketa nola egiten den ulertzea.
 - ▶ GUIak eraikitzeko Eclipse programazio ingurunea erabiltzen ikastea.
- 

Java programa motak

▶ Aplikazioak

Zuzenean Java ingurune baten exekutatu daitezke

Motak:

- Konsola modua
 - ✓ Teklatu bitarteko interakzioa
 - ✓ Testuan oinarritutako interfazea
- Interfaze grafikodun aplikazioak (GUI)
 - ✓ Datu sarrera eta irteerarako leiho grafikoak
 - ✓ Ikonoak
 - ✓ Sarrera gailuak (arratoia, teklatua)
 - ✓ Interakzio zuzena

▶ Applet-ak

Nabigatzaile batetan (edo applet bistaratzailean - Appletviewer) exekutatzen diren aplikazio txikiak

Interfaze grafikoa

Segurtasun aldetik mugatuak

GUIrentzako osagai bibliotekak

- ▶ Abstract Windowing Toolkit (AWT)
 - Plataformaren menpe dagoen “Look & Feel”: leihoen, menuen, ... itxura Windows, Mac, Motif eta beste sistemetan ezberdina da.
 - Plataformaren menpe ez dagoen funtzionalitatea.
 - JDK 1.1.5 bertsiorarte estandarra.

GUIrentzako osagai bibliotekak

- ▶ Swing / Java Foundation Classes (JDK 1.1.5tik aurrera)
 - “Look & Feel” eta funtzionalitatea ez daude plataformaren menpe (“Java Look & Feel”)
 - ✓ Menuak eta kontrolak aplikazio “natibo”-arenak bezalakoak dira.
 - ✓ Plataforma konketuaren arabera aplikazioei itxura ezberdina eman diezaieke.
 - Funtzionalitate berriak
 - ✓ Irisgarritasunerako APlA beharrian espezikoak dituzten pertsonentzat

Adibidea

```
import javax.swing.*;

public class ProbaVentana{
// Leihua objetua
public static void main (String args[]) {
    static JFrame Leihua = new JFrame("Leiharen izenburua");
    Leihua.setBounds(50, 100, 400, 150); // kokapena eta tamaina ezarri
    Leihua.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
    Leihua.setVisible(true); // Leihua erakutsi
}
}
```

setDefaultCloseOperation metodoaren argumentuak:

DISPOSE_ON_CLOSE – Leihua “deusezten” du

DO_NOTHING_ON_CLOSE – Leiharen aukerak desgaitu

HIDE_ON_CLOSE – Leihua ezkutatzen du

Oinarrizko osagaiak

- ▶ GUI osagaiak (widgets): interfazeko objektu bisualak
 - ✓ Programa grafiko bat osagai anidatuen multzo bat da: Leihoak, kontendoreak, menuak, barrak, botoiak, testu eremuak, eta abar.
- ▶ Diseinu edo kokapen kudeatzaileak (*layout managers*)
 - ✓ Interfazaren osagai grafikoen antolaketa kudeatzen dute
- ▶ Grafiko eta testu sorrera – Graphics klasea
 - ✓ Testuak marrazten ditu
 - ✓ Marrak, irudiak, koloreztatzea,...
- ▶ Interaktibitatea: ebentoen kudeaketa
 - ✓ Teklatua
 - ✓ Sagua

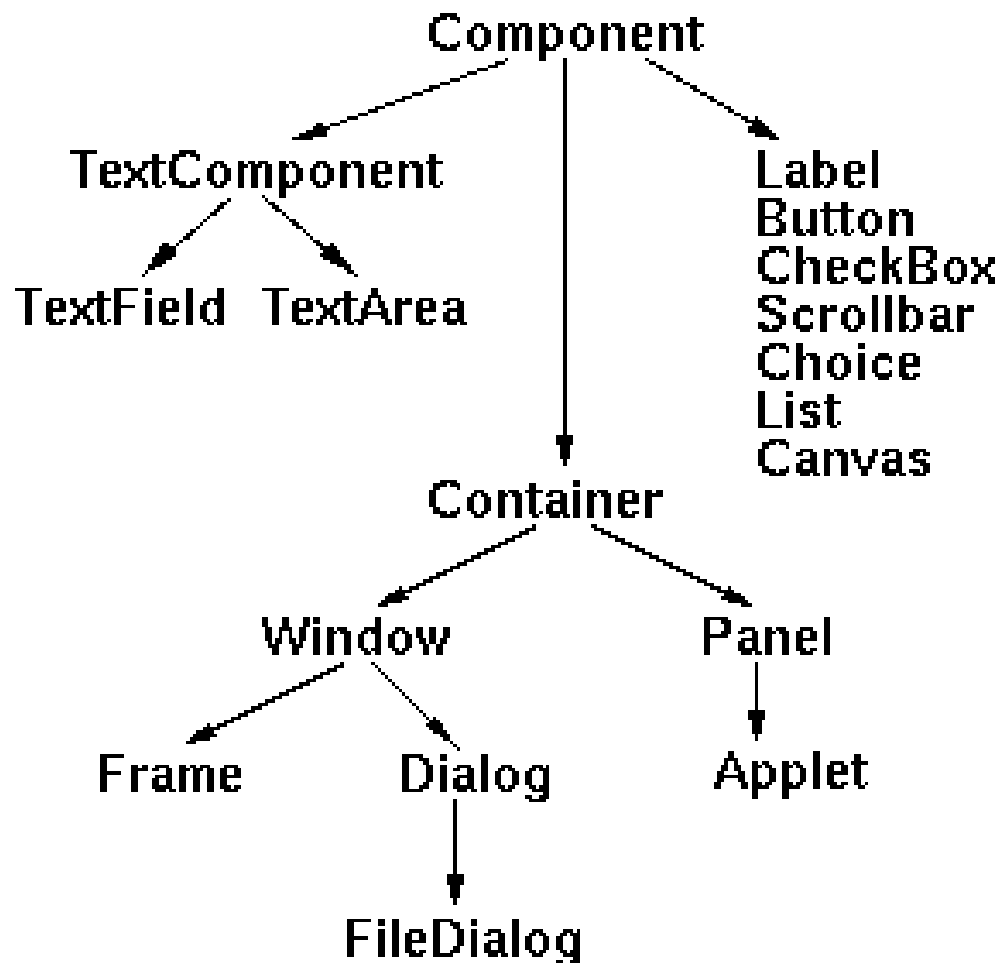
Edukiontziak eta osagaiak

- ▶ Java interfaze grafiko bat bi motako elementutan oinarritzen da:
 - ✓ Edukiontziak (containers): beste osagaiak taldekatu eta erakusteko erabiltzen dira.
 - ✓ Osagaiak: botoiak, etiketak, scrollbarak, edukiontziak, eta abar.
- ▶ Honela, edozein interfaze grafikok, behintzat container bat izango du. Honetarikoa bat izango da:
 - ✓ JFrame: programaren leiho nagusia
 - ✓ JDialog: elkarrizketentzat leihoa
 - ✓ JApplet: appletentzat leihoa

Edukiontziak eta osagaiak

- ▶ Java interfaze grafiko bat eraikitzeko, leiho baten eremuan osagaiak gehitzen dira (botoiak, menuak, testuak, taulak, listak, eta abar).
- ▶ Horregatik, leiho bat edukiontzi bat da: osagai multzo bat barnean hartzen duen elementua.

AWT osagaien hierarkia



Swing osagaiak

- ▶ Edukiontzia:

Beste osagai (edo edukiontzia) barnean izan ahal dituzte:

- ✓ Osagai hauek edukiontzia gehitu ahal dizkiogu eta, zenbait eragiketetan, oso bakarra bezala tratatu behar da.
- ✓ Diseinu kudeatzaile baten bitartez, osagaien antolamendua pantailan (layout) kontrolatzen da.
- ✓ Adibidez: JPanel, JFrame, JApplet

- ▶ Mihisea (*Canvas klasea*):

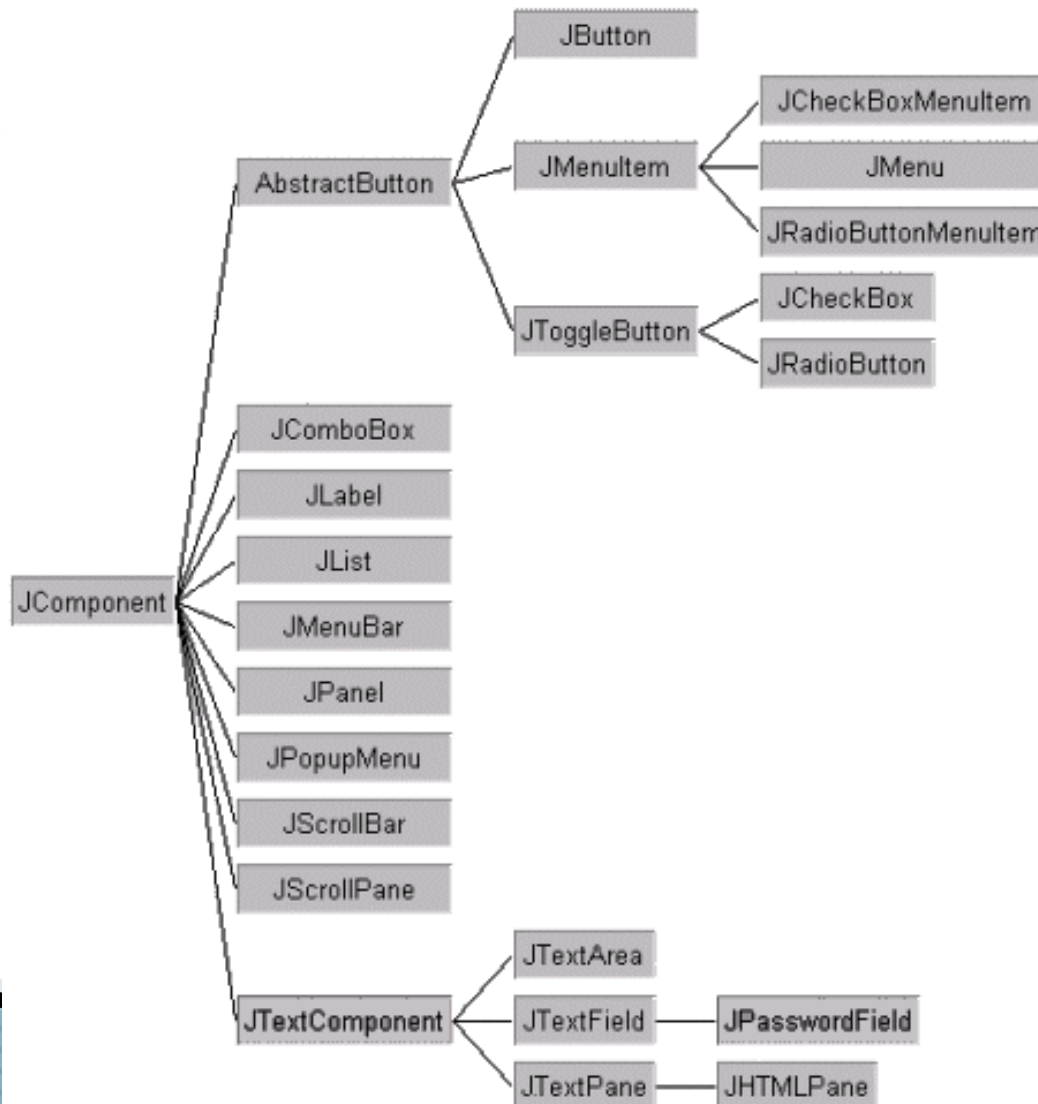
Marrazketa azalera sinplea

- ▶ Erabiltzaile-interfazearen osagaiak:

Botoiak, zerrendak, menuak, testu eremuak, e. a.

- ▶ Leihoak eraikitzeko osagaiak:

Leihoak, markoak, menu barrak, elkarrizketa leihoak, e. a.



Osagaiak

- ▶ **TextField (JTextField) klasea:**
 - Sarrera area bat generatzen du lerro bakarrekoa.
 - Informazioa ateratzeko ere balio dezake.

Osagaiak

- ▶ **TextArea (JTextArea) klasea:**
 - Klase hau erabiliz testu lerroak (bat baino gehiago!!!!) idatz eta irakur ditzakegu.

Adibidea

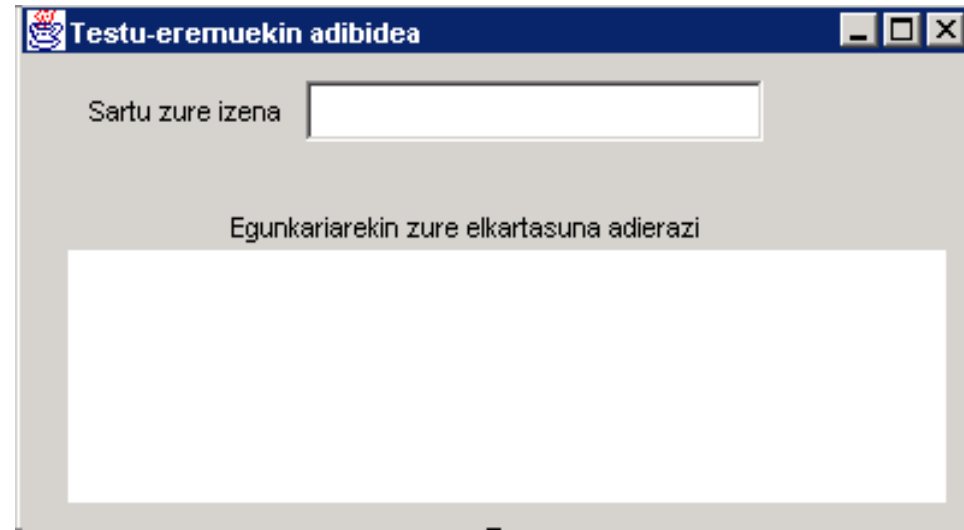
```
import javax.swing.*;
import java.awt.*;

public class TestuEremuak extends JFrame {
    JLabel jLabel1 = new JLabel();
    JTextField jTextField1 = new JTextField();
    JLabel jLabel2 = new JLabel();
    JTextArea jTextArea1 = new JTextArea();

    TestuEremuak(){
        try {
            jbInit();
        }
        catch (Exception e) {
            e.printStackTrace();}
    }

    public jbInit(){
        this.setTitle("Testu-eremuekin adibidea");
        jLabel1.setText("Sartu zure izena");
        jTextField1.setColumns(25);
        jLabel2.setText("Egunkariarekin zure elkartasuna adierazi");
        jTextArea1.setColumns(50);
        jTextArea1.setRows(10);
        this.getContentPane().add(jTextArea1, null);
        this.getContentPane().add(jLabel2, null);
        this.getContentPane().add(jTextField1, null);
        this.getContentPane().add(jLabel1, null);
    }

    public static void main(String[] args){
        Frame frame = new TestuEremuak();
        frame.setVisible(true);
    }
}
```



Osagaiak

- ▶ **Button (JButton) klasea:**
 - Botoiak sortzeko eta ondoren ekintza burutzeko balio du klase honek.

Osagaiak

► Label (JLabel) klasea:

- Etiketak(label) erabiltzaileari informazioa emateko jarriko ditugu. Beste osagaien (TextField,...) lagungarriak dira.
- Etiketak kontenedorearen (container) barruan hiru era desberdinetan lerroka daitezke:
 - Label.LEFT: Ezker lerrokatzea.
 - Label.CENTER: Erdian.
 - Label.RIGHT: Eskuin lerrokatzea.

Adibidea

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

public class Botoiak extends JFrame
{
    JButton jButton1 = new JButton();
    JButton jButton2 = new JButton();
    JButton jButton3 = new JButton();

    public Botoiak() {
        this.setTitle("Botoien adibidea");
        jButton1.setText("Ireki");
        jButton2.setText("Gorde");
        jButton3.setText("Ezeztatu");
        this.getContentPane().add(jButton3, BorderLayout.EAST);
        this.getContentPane().add(jButton2, BorderLayout.CENTER);
        this.getContentPane().add(jButton1, BorderLayout.WEST);
    }

    public static void main(String[] args){
        Frame frame = new Botoiak();
        frame.setVisible(true);
    }
}
```



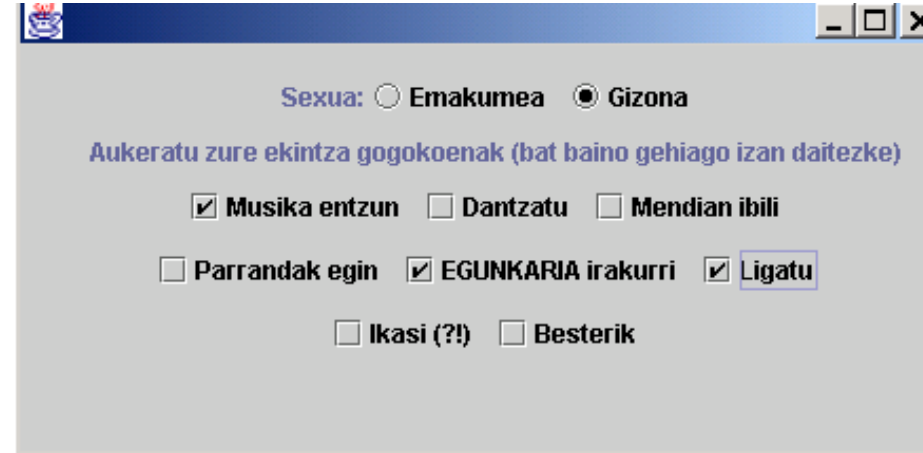
Osagaiak

- ▶ **Checkbox (JCheckBox/JRadioButton) klasea:**
 - Klase hau erabiliz erabiltzaileak aukera desberdinen artean bat aukera dezake.
 - Aukera aktibatu eta desaktibatu egin daiteke.
 - **Oharra:**
 - Checkbox-ak karratuak direnean bat baino gehiago aukera daiteke.
 - Borobilak direnean aukera bat bakarra egin behar da.

Adibidea

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

public class Aukerak extends JFrame{
    JPanel jPanel1 = new JPanel();
    JLabel jLabel1 = new JLabel();
    JRadioButton jButton1 = new JRadioButton();
    JRadioButton jButton2 = new JRadioButton();
    ButtonGroup g = new ButtonGroup();
    JLabel jLabel2 = new JLabel();
    JCheckBox jCheckBox1 = new JCheckBox();
    JCheckBox jCheckBox2 = new JCheckBox(); //Besteak eraiki
    public Aukerak() {
        jLabel1.setText("Sexua:");
        jButton1.setText("Gizona");
        jButton2.setText("Emakumea");
        jLabel2.setText("Aukeratu zure ekintza gogokoenak (bat baino gehiago izan daitezke)");
        jCheckBox1.setText("Musika entzun");
        jCheckBox2.setText("Dantzatu");//Besteenak testuak gehitu
        jPanel1.add(jLabel1, null);
        jPanel1.add(jButton2, null);
        jPanel1.add(jButton1, null);
        jPanel1.add(jLabel2, null);
        jPanel1.add(jCheckBox1, null); //Beste JCheckBox-ak gehitu
        g.add(jButton1);
        g.add(jButton2);
        this.getContentPane().add(jPanel1, null);
    }
}
```



The screenshot shows a Java Swing window titled "Aukerak" with a standard Mac OS X title bar. The window contains a form with the following elements:

- A label "Sexua:" followed by two radio buttons: "Emakumea" (unselected) and "Gizona" (selected).
- A label "Aukeratu zure ekintza gogokoenak (bat baino gehiago izan daitezke)".
- Three checkboxes: "Musika entzun" (checked), "Dantzatu" (unchecked), and "Mendian ibili" (unchecked).
- Three checkboxes: "Parrandak egin" (unchecked), "EGUNKARIA irakurri" (checked), and "Ligatu" (checked).
- Two checkboxes: "Ikasi (?!)" (unchecked) and "Besterik" (unchecked).

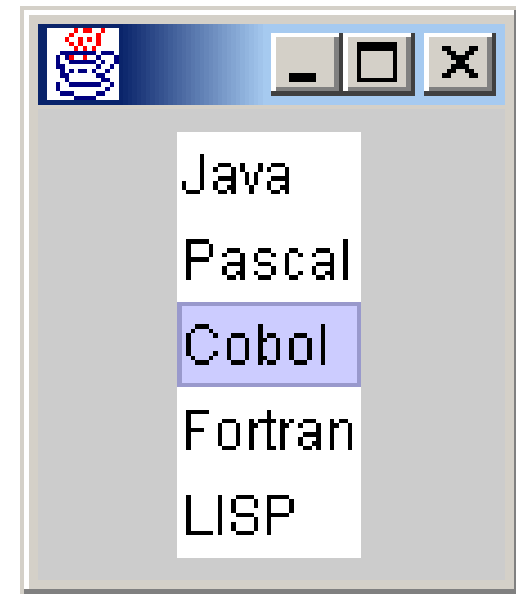
Osagaiak

► List (JList) klasea:

- Klase hau erabiliz aukera desberdinak eskaintzen zaizkio erabiltzaileari honek aukera egin dezan.
- Aukerak scroll erako pantaila txiki batean azaltzen dira espazioa aurrezteko leiho nagusian.

Adibidea

```
import javax.swing.*;
import java.awt.*;
import java.util.*;
public class Listak extends JFrame
{
    JList jList1; //new gero egingo da
    Vector elementuak = new Vector();
    JPanel jPanel1 = new JPanel();
    public Listak(){
        this.setTitle("Listen adibidea");
        elementuak.addElement("Java");
        elementuak.addElement("Pascal");
        elementuak.addElement("Cobol");
        elementuak.addElement("Perl");
        jList1 = new JList(elementuak);
        jPanel1.add(jList1, null);
        this.getContentPane().add(jPanel1, null);
        elementuak.addElement("LISP"); //elementua
            aurkezten da oraindik framea
            aurkeztu ez delako
    }
    public static void main(String[] args){
        Frame frame = new Listak();
        frame.setVisible(true);
    }
}
```

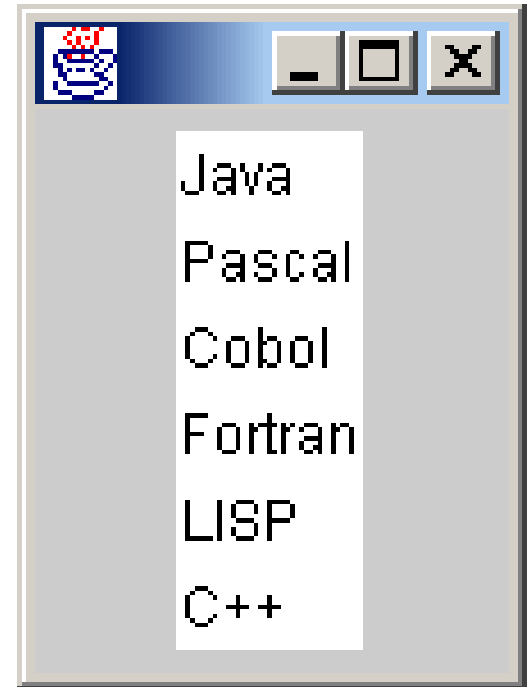


**frame.elementuak.addElement(
“C++”); ez da aurkeztuko**

Vector ez dabil egitura “dinamikoa” bada

Adibidea

```
import javax.swing.*;
import java.awt.*;
import java.util.*;
public class Listak extends JFrame
{
    JList jList1; //new gero egingo da
    DefaultListModel elementuak = new DefaultListModel();
    JPanel jPanel1 = new JPanel();
    public Listak(){
        this.setTitle("Listen adibidea");
        elementuak.addElement("Java");
        elementuak.addElement("Pascal");
        elementuak.addElement("Cobol");
        elementuak.addElement("Perl");
        jList1 = new JList(elementuak);
        jPanel1.add(jList1, null);
        this.getContentPane().add(jPanel1, null);
        elementuak.addElement("LISP"); //Vector-en
            gehitze, JLIST-a aldatzen da!!
    }
    public static void main(String[] args){
        Frame frame = new Listak();
        frame.setVisible(true);
        frame.elementuak.addElement("C++");
        frame.setVisible(true);
    }
}
```

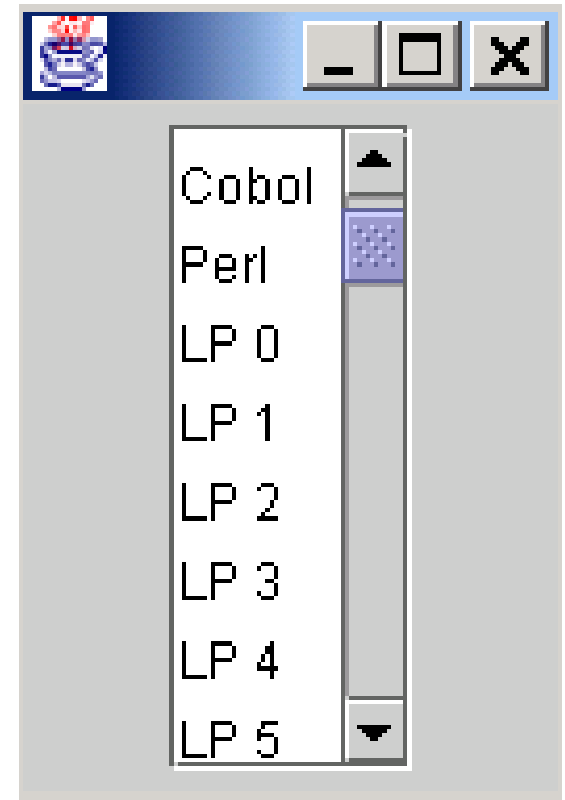


javax.Swing DefaultListModel
klasea eskaitzen du Vector
klasearen metodo berdinekin

**ListModel erabili zerrenda “dinamikoa”
denean**

Adibidea

```
import javax.swing.*;
import java.awt.*;
import java.util.*;
public class ListakScrollekin extends JFrame {
    JPanel jPanel1 = new JPanel();
    JList jList1;
    Vector elementuak = new Vector();
    public ListakScrollekin() {
        this.getContentPane().add(jPanel1, null);
        elementuak.addElement("Java");
        elementuak.addElement("Pascal");
        elementuak.addElement("Cobol");
        elementuak.addElement("Perl");
        jList1 = new JList(elementuak);
        JScrollPane j = new JScrollPane(jList1);
        //Lista sartzen dugu scroll-a duen panel batean
        jPanel1.add(j,null);
        for (int i=0;i<50;i++) elementuak.addElement("LP "+i);
        pack();
    }
    public static void main(String[] args){
        Frame frame = new ListakScrollekin();
        frame.setVisible(true);
    }
}
```



Adibidea

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

public class ComboBoxak extends JFrame {
    DefaultComboBoxModel elementuak = new DefaultComboBoxModel ();
    JPanel jPanel1 = new JPanel();
    JComboBox jComboBox1;//new gero egingo da

    public ComboBoxak(){
        elementuak.addElement("Java");
        elementuak.addElement("Pascal");
        elementuak.addElement("Cobol");
        elementuak.addElement("Perl");
        jComboBox1 = new JComboBox(elementuak);
        jPanel1.add(jComboBox1, null);
        this.getContentPane().add(jPanel1, null);
        elementuak.addElement("LISP");
        pack();
    }

    public static void main(String[] args){
        Frame frame = new ComboBoxak();
        frame.setVisible(true);
    }
}
```

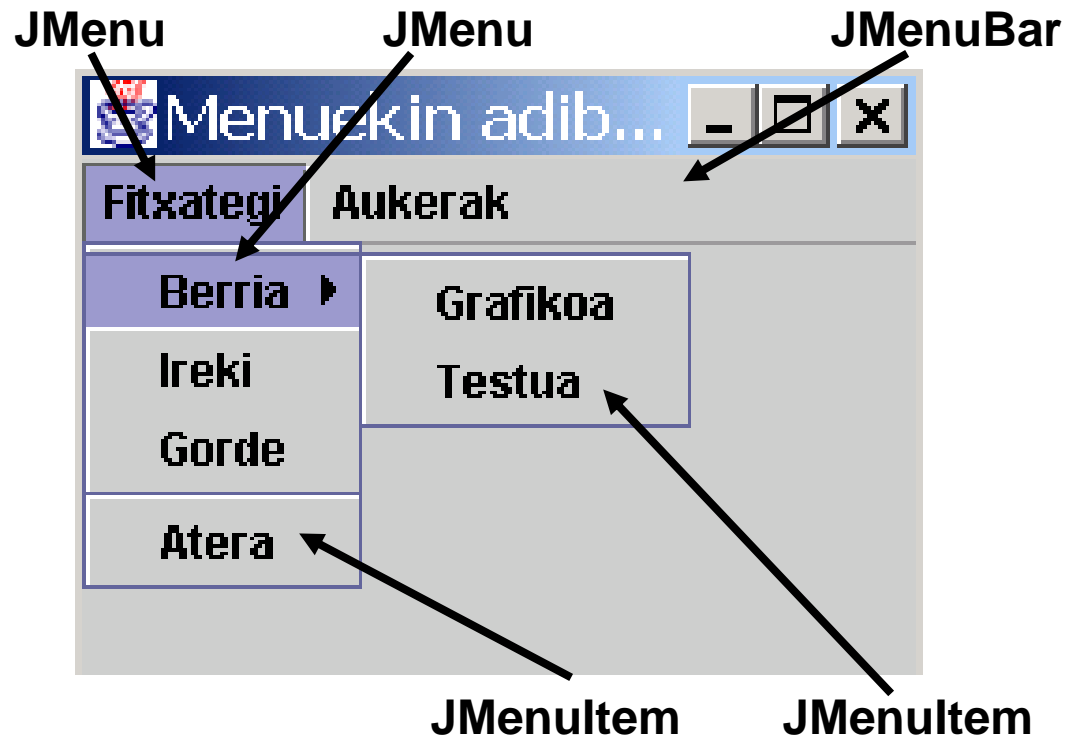


Osagaiak

- ▶ **Menu (JMenu/JMenuItem/JMenuBar) klasea:**
 - Aplikazioak garatzerakoan asko erabiltzen den aukera.
 - Frame klasea da menuak sortzeko balio duen edukiontzi (container) bakarra.
 - Orain arte ikusitako osagaiek baino egitura konplexuagoa du.

Adibidea

► Menuen sorkuntza Swing-en



Adibidea

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

public class Menuak extends JFrame {
    JMenuBar menuBarra = new JMenuBar();
    JMenu fitxategi = new JMenu(); JMenu aukerak = new JMenu();
    JMenu berria = new JMenu(); JMenuItem ireki = new JMenuItem();
    JMenuItem gorde = new JMenuItem(); JMenuItem atera = new JMenuItem();
    JMenuItem koloreak = new JRadioButtonMenuItem();
    JMenuItem txuriBeltz = new JRadioButtonMenuItem();
    ButtonGroup bg = new ButtonGroup();
    JMenuItem testua = new JMenuItem(); JMenuItem grafikoa = new JMenuItem();

    public Menuak() {
        this.setJMenuBar(menuBarra); this.setTitle("Menuekin adibidea");
        fitxategi.setText("Fitxategi"); aukerak.setText("Aukerak");
        berria.setText("Berria"); grafikoa.setText("Grafikoa");
        testua.setText("Testua"); ireki.setText("Ireki");
        gorde.setText("Gorde"); atera.setText("Atera");
        koloreak.setText("Koloreak"); txuriBeltz.setText("Txuri-beltz");
        testua.setText("Testua");
        berria.add(grafikoa); berria.add(testua); berria.add(testua);
        fitxategi.add(berria); fitxategi.add(ireki); fitxategi.add(gorde);
        fitxategi.addSeparator(); fitxategi.add(atera);
        menuBarra.add(fitxategi);
        aukerak.add(txuriBeltz); aukerak.add(koloreak);
        bg.add(txuriBeltz); bg.add(koloreak);
        menuBarra.add(aukerak);
    }

    public static void main(String[] args){
        Frame frame = new Menuak();
        frame.setVisible(true);
    }
}
```



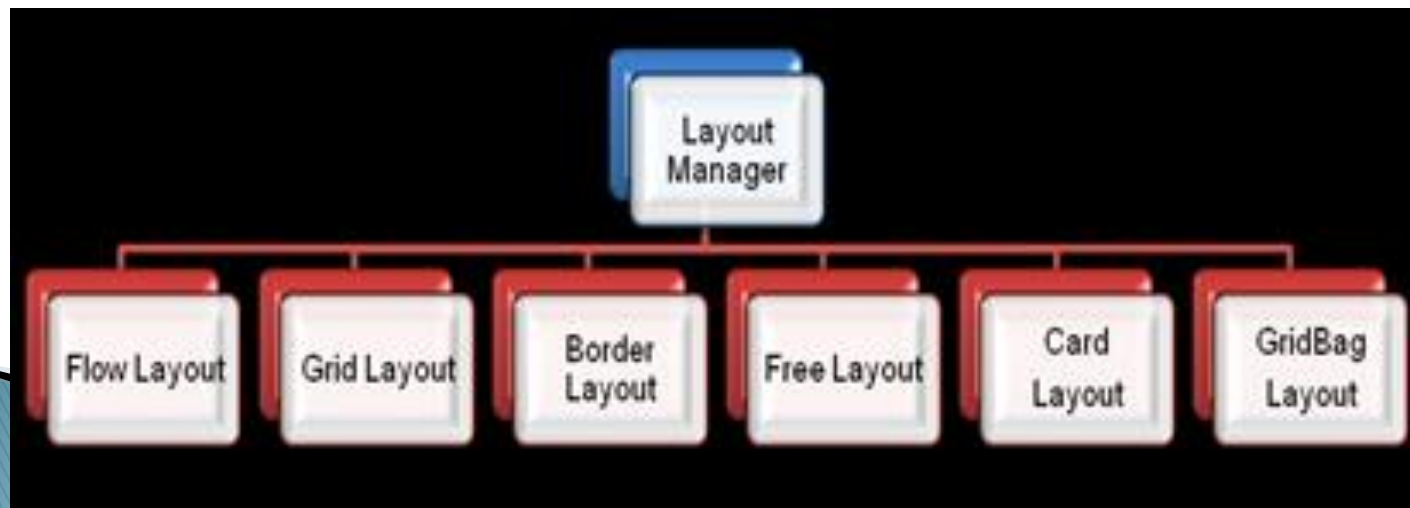
Layout kudeatzaileak: Motibazioa

Zer hartu behar da kontutan interface grafiko bat diseinatzeko?

- ▶ Objektuen berdimentsionatzea Leihoaren tamaina aldatzen denean.
- ▶ Leihoaren bermarraztea, beste leiho bat gainean ezartzen denean.
- ▶ Objektu grafikoak tamaina ezberdinekoak izan daitezke.
- ▶ ...

Deskribapena

- ▶ Layout kudeatzaileek objektua grafikoak kontenedore baten antolatzen dituzte.
- ▶ Osagaiek izango duten itxura espezifikatzen dute, bai tamaina baita posizioa ere kontrolatuz.



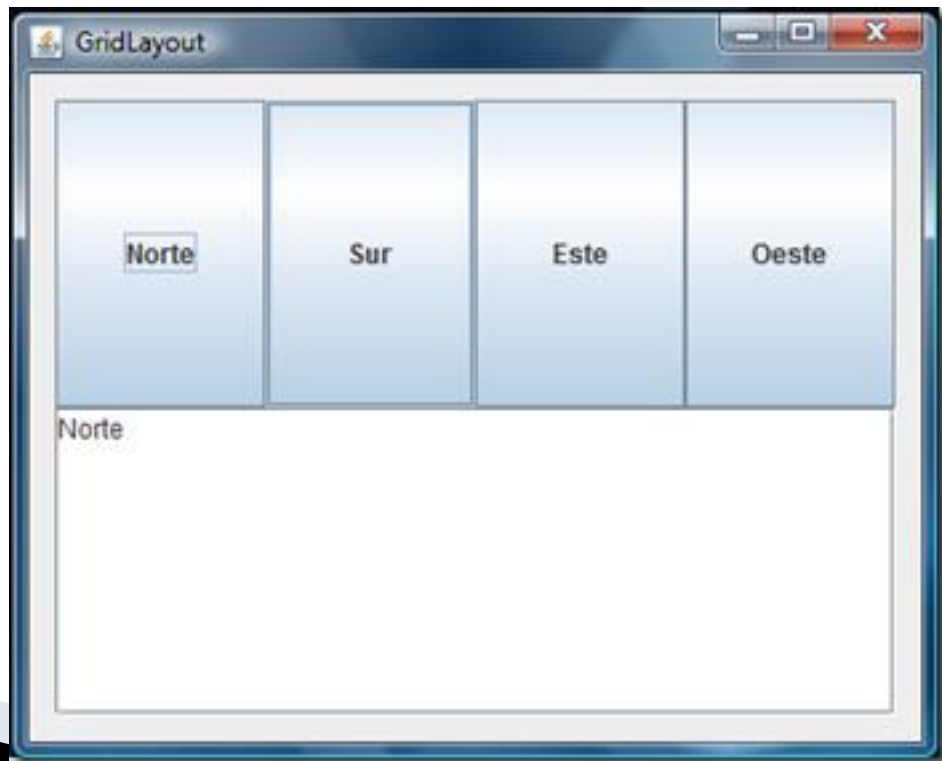
FlowLayout

Osagaiak lerroetan edo zutabetan antolatzen dira, eraikitzailean adierazitakoaren arabera. Lerroa edo zutabea betetzen denean, berri bat hasiko da.



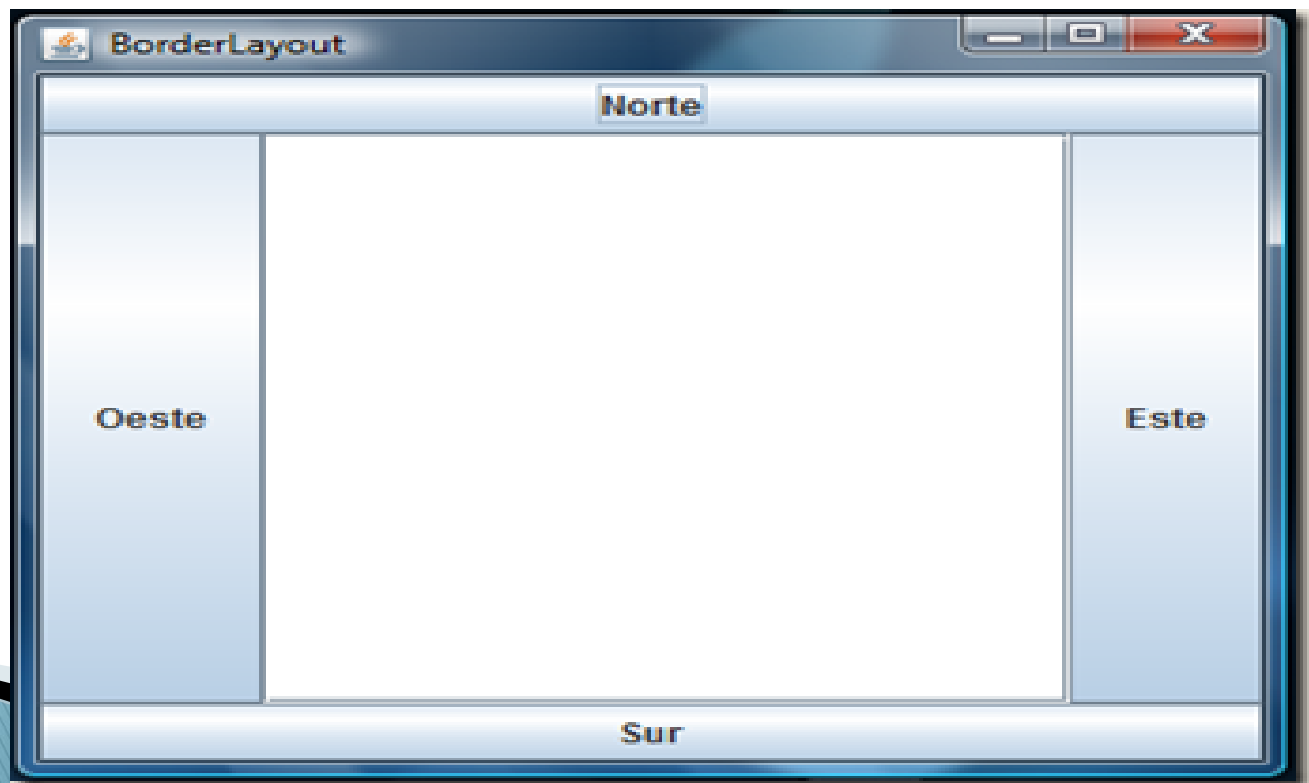
GridLayout

Osagaiak taula baten (lerro eta zutabeekin) banatzen dira. Kontenedorea laukizuzenetan banatzen da eta laukizuzen bakoitzean osagai bat kokatzen da.



BorderLayout

Osagaiak bost eskualdetan antolatzen dira: iparraldea, ekialdea, hegoaldea, mendebaldea eta erdia.



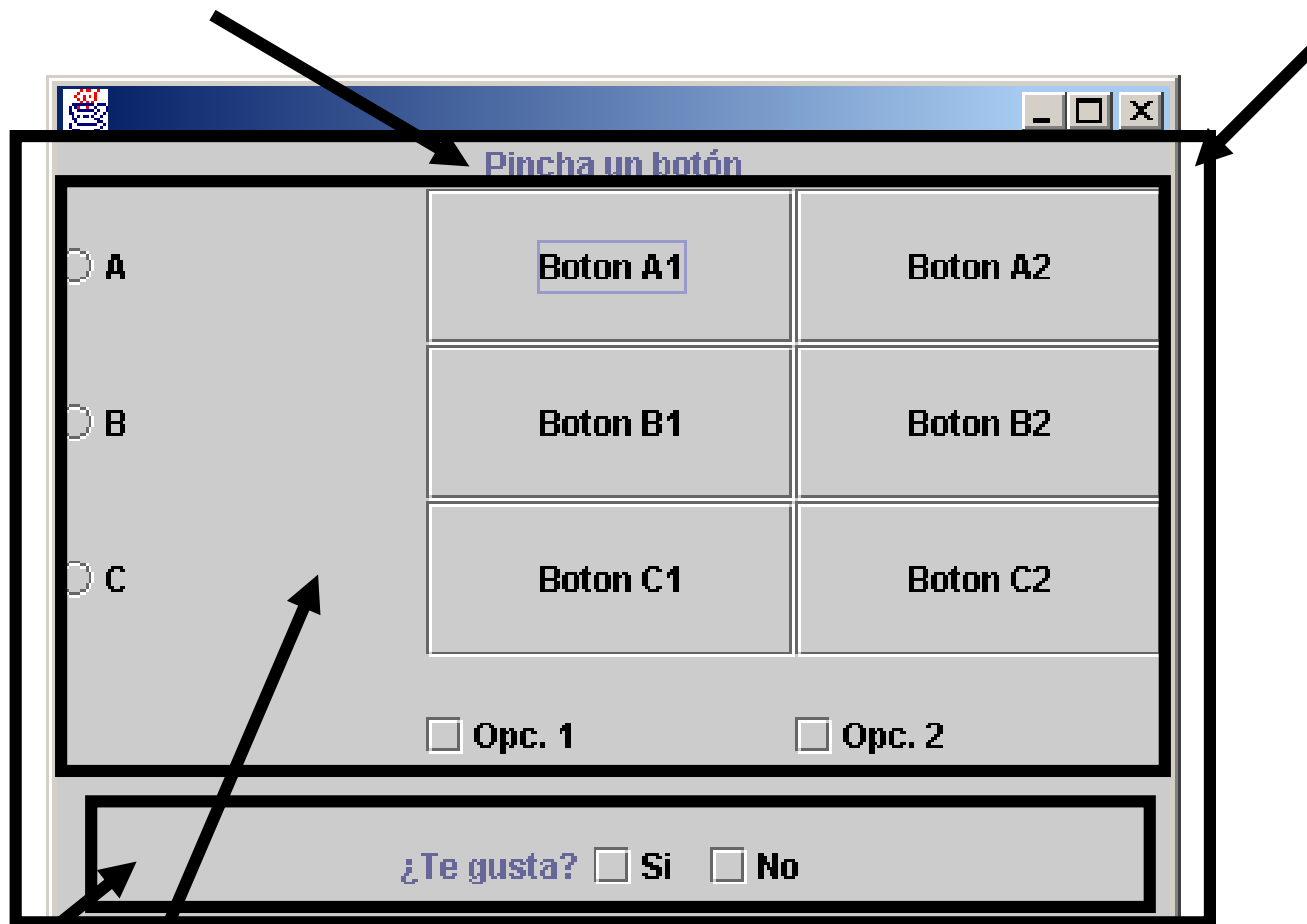
Layout

- ▶ Osagaia koordenatu zehatzetan jarri nahi bada diseinu administratzailea kendu egin behar da
`osagaia.setLayout(null);`
`// edo setLayout(new XYLayout());`
- ▶ `this` edukiontzi bat bada eta `textField1` bere osagai bat
`setLayout(null);`
`textField1.setBounds(15,20,50,60);`
kokatzen du :



Label
BorderLayout.NORTH

Panel
BorderLayout



Panel GridLayout(4,3) BorderLayout.CENTER-en kokatua

Panel FlowLayoutekin BorderLayout.SOUTH-en kokatua

Pincha un botón

☐ A

☐ B

☐ C

Boton A1	Boton A2
Boton B1	Boton B2
Boton C1	Boton C2

☐ Opc. 1 ☐ Opc. 2

¿Te gusta? ☐ Si ☐ No

Pincha un botón

☐ A

☐ B

☐ C

Boton A1	Boton A2
Boton B1	Boton B2
Boton C1	Boton C2

☐ Opc. 1 ☐ Opc. 2

¿Te gusta? ☐ Si ☐ No

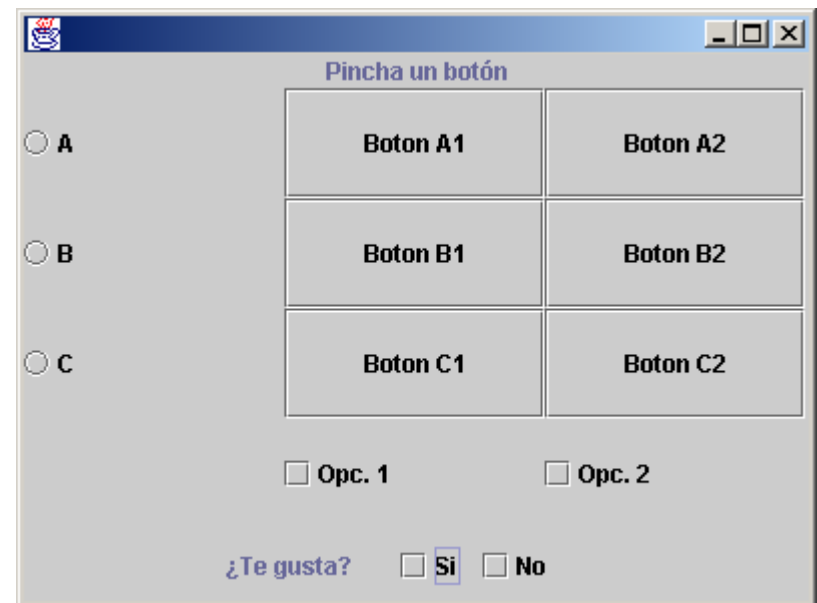
Abantaila: Diseinu administratzaileak erabiltzerakoan leihoaren forma aldatzen badugu osagaiak automatikoki egoera berrira egokitzen dira.

```

this.setSize(new Dimension(400, 300));
jPanel1.setBounds(new Rectangle(0, 0, 392, 273));
jPanel1.setLayout(null);
jCheckBox1.setLabel("Opc. 1");
jCheckBox1.setBounds(new Rectangle(130, 165, 130, 55));
jCheckBox2.setLabel("Opc. 2");
jCheckBox2.setBounds(new Rectangle(260, 165, 130, 55));
jPanel3.setBounds(new Rectangle(0, 165, 130, 55));
jPanel3.setLayout(null);
jPanel2.setBounds(new Rectangle(0, 238, 392, 35));
jPanel2.setLayout(null);

```

**Diseinu administratzaile
gabe: Osagai guztien
koordinatuak ematen
dira(tresna aurreratu
batekin ez da lan zaila)**



► Baina...

- Framea berdimentsionatzean osagaian zeuden lekuan geratzen dira.

A screenshot of a software window with a blue title bar and standard Windows controls. The window contains a form with three radio buttons labeled A, B, and C. To the right of the radio buttons is a table with three rows and two columns. The first row contains 'A1' and 'A2', the second row contains 'B1' and 'B2', and the third row contains 'C1' and 'C2'. The 'C2' button is highlighted with a blue border. Below the table, there is a label 'Gustatzen' followed by two checkboxes labeled 'Bai' and 'Ez'.

<input type="radio"/> A	A1	A2
<input type="radio"/> B	B1	B2
<input type="radio"/> C	C1	C2

Gustatzen ☐ Bai ☐ Ez

A screenshot of a software window with a blue title bar and standard Windows controls. The window contains a form with two radio buttons labeled A and B. To the right of the radio buttons is a table with two rows and two columns. The first row contains 'A1' and 'A2', and the second row contains 'B1' and 'B2'. The 'B2' button is highlighted with a blue border. Below the table, there is a label 'Gustatzen' followed by two checkboxes labeled 'Bai' and 'Ez'.

<input type="radio"/> A	A1	A2
<input type="radio"/> B	B1	B2

Gustatzen ☐ Bai ☐ Ez

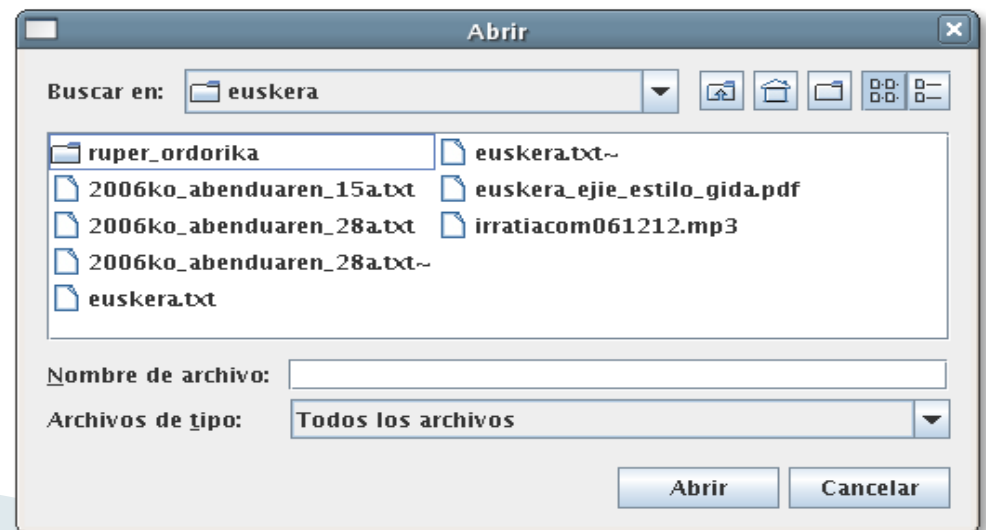
A screenshot of a software window with a blue title bar and standard Windows controls. The window contains a form with three radio buttons labeled A, B, and C. To the right of the radio buttons is a table with three rows and two columns. The first row contains 'A1' and an empty box, the second row contains 'B1' and an empty box, and the third row contains 'C1' and an empty box. The 'C1' button is highlighted with a blue border. Below the table, there is a label 'Gustatzen' followed by two checkboxes labeled 'Bai' and 'Ez'.

<input type="radio"/> A	A1	
<input type="radio"/> B	B1	
<input type="radio"/> C	C1	

Gustatzen ☐ Bai ☐ Ez

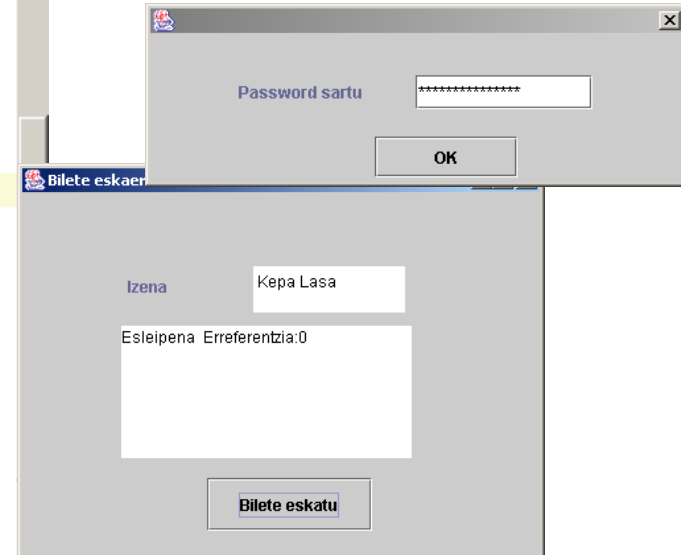
Beste edukiontzi batzuk

- ▶ **Dialog/JDialog klaseak**
 - Erabiltzailearen datuak irakurtzeko leihoa.
 - MODAL ezaugarria jartzen badiogu, aktiboa dagoen bitartean ezin izango da beste leiho batetara aldatu.
- ▶ **JFileChooser klasea**



Adibidea

```
C:\adEuskara\Project5\src\mypackage4\DialogPassword.java
10
11 public class DialogPassword extends JDialog
12 {
13     private JPasswordField jPasswordField1 = new JPasswordField();
14     private JLabel jLabel1 = new JLabel();
15     private JButton jButton1 = new JButton();
16
17     public DialogPassword(Frame parent, String title, boolean modal)
18     {
19         this.setSize(new Dimension(400, 135));
20         this.getContentPane().setLayout(null);
21         jPasswordField1.setText("jPasswordField1");
22         jPasswordField1.setBounds(new Rectangle(195, 30, 130, 25));
23         jLabel1.setText("Password sartu");
24         jLabel1.setBounds(new Rectangle(65, 35, 110, 15));
25         jButton1.setText("OK");
26         jButton1.setBounds(new Rectangle(165, 75, 105, 30));
27         this.getContentPane().add(jButton1, null);
28         this.getContentPane().add(jLabel1, null);
29         this.getContentPane().add(jPasswordField1, BorderLayout.CENTER);
30     }
31
32     public static void main(String args[]){
33         Frame1 a=new Frame1();
34         DialogPassword dp=new DialogPassword(a,"Password frogaketa",true);
35         a.setVisible(true);
36         dp.setVisible(true);
37     }
38 }
```



Modal lehioa, itxi arte ezin d
aurreko lehiora joan