



Programazioa

Interfaze grafikoak: Swing I

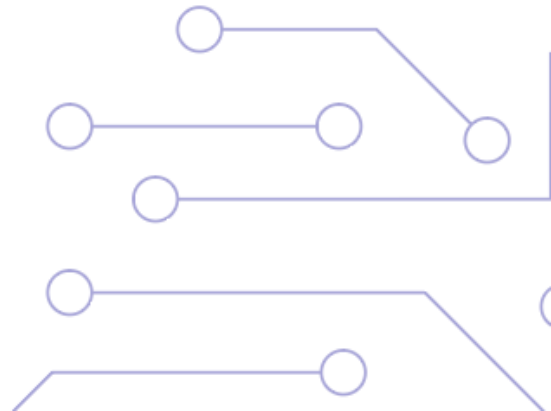
(Kontainer eta elementuak)

Interfaze grafikoak

- *Graphical User Interface (GUI)*: Erabiltzaileak aplikazioarekin elkar eragiteko balio duen goi-mailako adierazpide bisuala duen interfazea
 - Leiho, botoi, menu, barra, etab.-etaz osatuta dago
- Beraien egitura eta portaeragatik Objektuetara Bideratutako Programazioaren printzipioak jarraitzen dituzte definizioz
- Exekuzioa ez da lineala, erabiltzaileak aktibatzen dituen gertaeretan oinarritzen da
 - Gertaerak asinkronoak dira, erabiltzailearen ekintzek aktibatuta
 - Botoi bat sakatzea, elementu batean klikatzea, tekla bat sakatzea, leihoaren tamaina aldatzea...

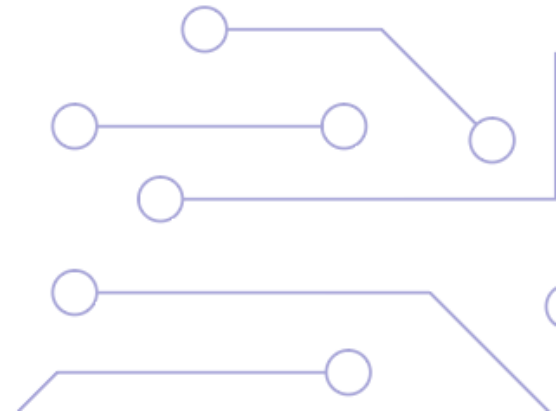
Swing

- Javan interfaze grafikoak sortzeko tresnen liburutegia
- Bere itxura aplikazioa erabiltzen den plataformatik independentea da
 - Beste liburutegi batzuekin itxura aldatu egiten da sistema eragilearen arabera
- Elementu sinpleez gain (botoiak, etiketak, check box-ak...) goi-mailako beste elementu batzuk eskaintzen ditu: pestainak, taulak, zerrendak...



Kontainerrak eta elementuak

- Swing-en bi objektu mota desberdindu ditzakegu.
 - Kontainerrak: Bere barnean elementu sinpleak maketazio (*layout*) bat jarraituta antolatzeko erabiltzen diren objektuak dira. Besteak beste, leihoak dira kontainerrak.
 - Elementuak: Interfaze grafikoko objektu sinpleak, kontainerretan antolatzen direnak. Hala nola, botoiak, etiketak, menuak...



Kontainerrak: JFrame

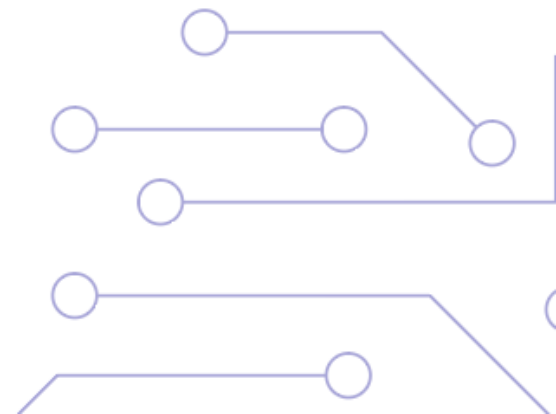
- Ertza, titulua eta handitu, txikitu eta ixteko ikonoak dituen goi-mailako leihoa da
- Aplikazio grafikoek gutxienez mota honetako kontainer bat dute
- JFrame batek modu lehenetsian beste elementu batzuk kokatzeko edukiontzi (*content pane*) bat dauka
- Zenbait metodo:
 - **JFrame()** eta **JFrame(String izena)**: Leihoa sortzeko
 - **getContentPane()**: Leihoaren edukiontzia jasotzeko
 - **setDefaultCloseOperationa(int aukera)**: Leihoa ixterakoan izan behar duen portaera (DO_NOTHING_ON_CLOSE | HIDE_ON_CLOSE | DISPOSE_ON_CLOSE | EXIT_ON_CLOSE)

Kontainerrak: JFrame

```
//Sorrera
JFrame frame = new JFrame("FrameDemo");
//Ixtarakoan duen portaera
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
//Tamaina esleitzea
frame.setPreferredSize(new Dimension(300, 200));
JLabel etiketa = new JLabel("Kaixo");
frame.getContentPane().add(etiketa, BorderLayout.CENTER);
//Bistaratzeta
frame.pack();
frame.setVisible(true);
```

Kontainerrak: JFrame

- Leiho konplexu bat sortu behar denean praktika ona da klase JFrame hedatzen duen klase berri bat sortzea
 - Hedatzen duen klase horren eraikitzailean gehitu (*add*), kendu (*remove*) eta maketazio mota ezartzeko (*setLayout*) metodoak berridatzita daude
 - Ez da beharrezkoa edukiontzia jasotzea (*getContentPane*) metodo horiek erabiltzeko



Kontainerrak: JFrame

```
public class NireLeiho extends JFrame
{
    NireLeiho()
    {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setPreferredSize(new Dimension(300,200));
        setLayout(new BorderLayout());
        JButton botoi = new JButton("Nire botoia");
        add(botoi);
        pack();
        setVisible(true);
    }
}
```


Kontainerrek: JPanel

- Normalean ikusten ez den kontainer bat da, non elementuak kokatzen dira
- Modu lehenetsian JFrame baten barruan horrelako bat dago
- Metodoak:
 - JPanel(): Lehenetsitako eraikitzailea
 - JPanel(LayoutManager maketazioa): Beste eraikitzaile bat non maketazioa (*layout*) adierazten den
 - void setLayout(LayoutManager maketazioa): Kontainerren maketazioa ezartzen du

Kontainerrak: JPanel

- Metodoak:
 - void add (Component elementua): Parametro bezala pasatako elementua gehitzen du
 - void add(Component elementua, int posizioa): Parametro bezala pasatako elementua adierazitako posizioan gehitzen du
 - void remove (Component elementua): Parametro bezala pasatako elementua ezabatzen du
 - void remove(int posizioa): Parametro bezala pasatako posizioan dagoen elementua ezabatzen du
 - Beste metodo batzuk: getIcon, setIcon, getWidth, getHeight, setPreferredSize, setBackground.

Kontainerrak: JScrollPane

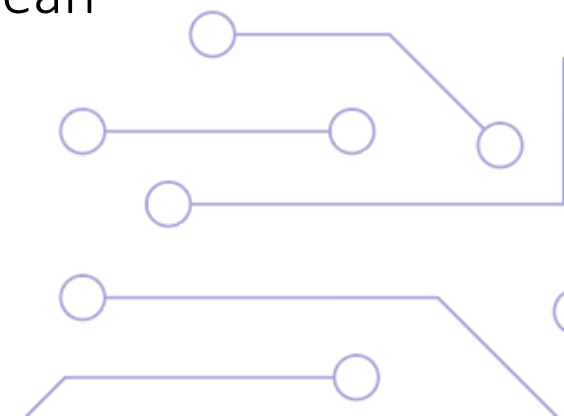
- Aurreko kontainerraren berdina da, baina *scroll* barra horizontal eta bertikalak izatea ahalbidetzen du
- Metodoak:
 - JScrollPane(Component kontainerra, int scroll_bertikala, int scroll_horizontala):
Eraikitzailea, zenbaki osoek scroll-aren politika ezartzen dute, beraien balioak hauek izan daitezkeelarik:
 - ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED // Edo HORIZONTAL
 - ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS
 - ScrollPaneConstants.VERTICAL_SCROLLBAR_NEVER
 - Beste metodo batzuk: setPreferredSize, setVerticalScrollBarPolicy, setHorizontalScrollBarPolicy, setVisibleRowCount.

Kontainerrek: JScrollPane

```
JFrame leiho = new JFrame();
JScrollPane panel1 = new JScrollPane();
JTextArea jt1 = new JTextArea("...", 10, 40);
leiho.setTitle("Nire aplikazioa");
leiho.setDefaultCloseOperation(leiho.EXIT_ON_CLOSE);
panel1.setViewportView(jt1);
panel1.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
leiho.getContentPane().add(panel1);
leiho.pack();
leiho.setVisible(true);
```

Kontainerrak: JFrame

- Guraso leiho baten barruan gehitu daitekeen kontainerra
- Aldi berean dokumentu bat baino gehiago erakutsi nahi denean erabiltzen da
- Sorrera hurrengo metodoak erabilita egin daiteke:
 - JFrame ()
 - JFrame (String izena)
 - JFrame (String izena, boolean tamainaAldatuDaiteke, boolean itxiDaiteke, boolean handituDaiteke)

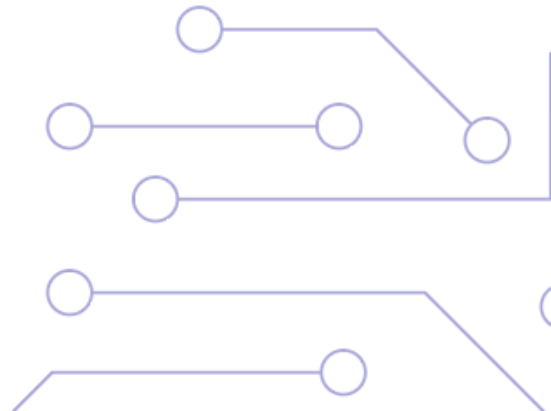


Kontainerrak: JFrame

```
NireLeiho()
{
    super("Internal frame sortzailea");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(450, 400);
    Container content = getContentPane();
    JDesktopPane desktop = new JDesktopPane();
    content.add(desktop, BorderLayout.CENTER);
    for(int i = 0; i < 5; i++) {
        JFrame frame = new JFrame("Azpileiho " + i), true,
        true, true, true);
        frame.setLocation(i * 50 + 10, i * 50 + 10);
        frame.setSize(200, 150);
        frame.setBackground(Color.white);
        frame.setVisible(true);
        desktop.add(frame);
        frame.moveToFront();
    }
    setVisible(true);
}
```

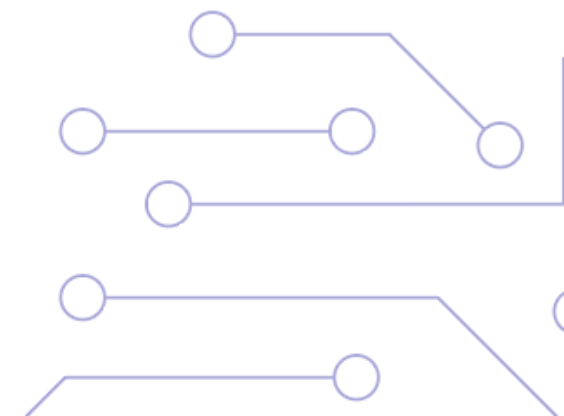
Kontainerrak: JDialog

- Erabiltzaileak daturen bat sartu dezan erabiltzen den leihoa
- Erabili ditzakeen elementu motak mugatuta daude, ezin du menurik eta tresna barrarik izan, adibidez
- JFrame batetara lotuta sortzen dira, guraso JFrame-a itxen denean bere JDialog guztiak ixten dira
- Sorrera hurrengo metodoak erabilita egin daitezke:
 - JDialog()
 - JDialog(Frame)
 - JDialog(Frame, String izena)
 - JDialog(Frame, String izena, boolean modalaDa)



Kontainerrak: JOptionPane

- Aurrez prestatutako Jdialog modalen bilduma
 - Leiho bat modala da hau itxi arte aplikazioan blokeatzen bada
- Erabiltzaileari abisuak emateko erabiltzen dira
- JOptionPane klaseak duen enumeratzailea erabiltzen da leiho mota adierazteko
 - JOptionPane.ERROR_MESSAGE
 - JOptionPane.INFORMATION_MESSAGE
 - JOptionPane.WARNING_MESSAGE
 - JOptionPane.QUESTION_MESSAGE
 - JOptionPane.PLAIN_MESSAGE



Kontainerrak: JOptionPane

```
JOptionPane.showMessageDialog(null, "Kontuz!", "Errorea",  
JOptionPane.ERROR_MESSAGE);
```

```
int egindakoAukera = JOptionPane.showConfirmDialog(null, "Aukeratu bat", "Aukera",  
JOptionPane.YES_NO_OPTION);
```

```
String sarrerakoBalioa = JOptionPane.showInputDialog("Sartu balio bat:");
```

Kontainerrak: JOptionPane

```
Object[] opt = { "OK", "CANCEL" };  
JOptionPane.showOptionDialog(null, "OK sakatu jarraitzeko", "Abisua",  
JOptionPane.DEFAULT_OPTION, JOptionPane.WARNING_MESSAGE, null, opt, opt [0]);
```

```
Object[] values = {"Bat", "Bi", "Hiru"};  
Object aukeratutakoBalioa = JOptionPane.showInputDialog(null, "Aukeratu bat",  
"Sarrera",JOptionPane.INFORMATION_MESSAGE, null, values, values[0]);
```

Kontainerrak: JFileChooser

- Fitxategi sistematik karpeta eta fitxategiak aukeratzea ahalbidetzen duen kontainerra
- Metodoak:
 - JFileChooser(String fitxategi_izena): Eraikitzailea
 - int showOpenDialog(Component): Kargatzeko leihoa erakusten du
 - int showSaveDialog(Component): Gordetzeko leihoa erakusten du
 - int showDialog(Component, String leiholzena): Leiho pertsonalizatua erakusten du
 - Hiru metodo hauetan irteera kodea jasotzen da, aukerak JFileChooser.APPROVE_OPTION, JFileChooser.CANCEL_OPTION eta JFileChooser.ERROR_OPTION direlarik

Kontainerrak: JFileChooser

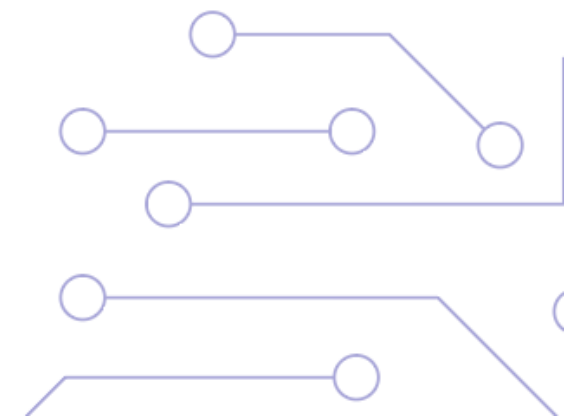
- Metodoak:
 - File getSelectedFile(): Aukeratutako fitxategia itzultzen du
 - File[] getSelectedFiles(): Aukeratutako fitxategiak itzultzen ditu
 - void setSelectedFile(File): Aukeratutako fitxategia parametro bezala jasotakoarekin aldatzen du
 - File getCurrentDirectory(): Unekeo karpeta itzultzen du
 - void setMultiSelectionEnabled(boolean): Fitxategi anitzak aukeratzea ahalbidetzen du

Kontainerrak: JFileChooser

```
JFileChooser fc = new JFileChooser("C:\\");
fc.setMultiSelectionEnabled(true);
int result = fc.showOpenDialog(new JFrame());
if (result == JFileChooser.APPROVE_OPTION)
{
    File file = fc.getSelectedFile();
    ...
}
```

Kontainerrak: JSplitPane

- Bi kontainer leiho berean bistartzeko balio du, bai horizontalki bai bertikalki
- Metodoak:
 - JSplitPane(int horizontalaEdoBertikala, Component kontainer1, Component kontainer2): Parametro bezala pasatako bi kontainerrekin JsplittedPane bat sortzen du. Zenbakizko parametroak horizontalki edo bertikalki banatu behar den zehazten du: JSplitPane.HORIZONTAL_SPLIT edo JSplitPane.VERTICAL_SPLIT.



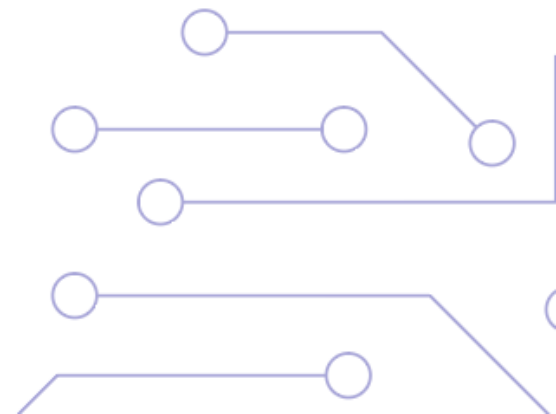
Kontainerrek: JTabbedPane

- Hainbat kontainer pestainatan antolatzeko balio duen kontainerra
- Metodoak:
 - JTabbedPane(): Lehenetsitako eraikitzailea
 - JTabbedPane(int kokapena, int maketazioa): Lehen parametroak adierazten duen posizioan ezartzen ditu pestainak eta bigarren parametroak adierazten duen maketazioa erabiltzen du
 - JTabbedPane.TOP, JTabbedPane.BOTTOM, JTabbedPane.LEFT eta JTabbedPane.RIGHT
 - JTabbedPane.SCROLL_TAB_LAYOUT edo JTabbedPane.WRAP_TAB_LAYOUT
 - Beste metodo batzuk: void addTab(String izena, Component kontainerra), void setSelectedIndex(int aukeratutakoPestaina)...



Kontainerrak: JToolBar

- Hainbat elementu (normalean botoiak ikonoekin) horizontalki edo bertikalki barra batean erakusten dituen kontainerra
- Barra dagoen kontainerren barruan posizio desberdinetara mugitu daiteke



Beste metodo erabilgarri batzuk

Metodoa	Esanahia
<code>void setVisible(boolean)</code> <code>boolean isVisible()</code>	Kontainerra bistaratzen du Ikusgai dagoen itzultzen du
<code>void setBounds(int, int, int, int)</code> <code>Rectangle getBounds()</code>	Adierazitako mugak esleitzen dizkio Mugak itzultzen ditu
<code>void setSize(int, int)</code> <code>Dimension getSize()</code>	Adierazitako tamaina esleitzen dio Tamaina itzultzen du
<code>void setBorder(Border)</code> <code>Border getBorder()</code>	Adierazitako ertza esleitzen dio Duen ertza itzultzen du
<code>void setForeground(Color)</code> <code>Color getForeground()</code>	Aurreko kolorea (letra) esleitzen du Aurreko kolorea itzultzen du
<code>void setBackground(Color)</code> <code>Color getBackground()</code>	Atzeko kolorea esleitzen du Atzeko kolorea itzultzen du

Beste metodo erabilgarri batzuk

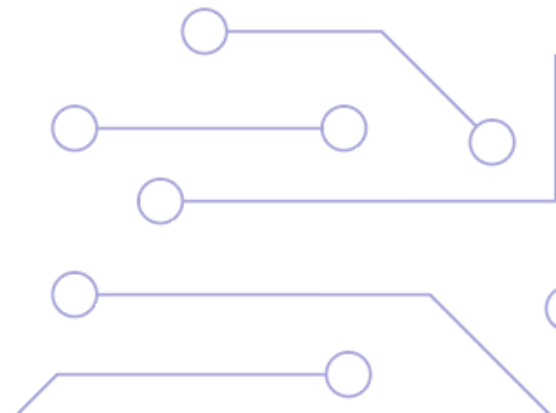
Metodoa	Esanahia
void setFont(Font) Font getFont()	Letra mota ezartzen du Letra mota itzultzen du
void setCursor(Cursor) Cursor getCursor()	Kurtsore mota ezartzen du Kurtsorea itzultzen du
void setToolTipText(String) String getToolTipText()	Sagua elementu gainean kokatzean ikusten den testua ezartzen du Sagua elementu gainean kokatzean ikusten den testua itzultzen du
void pack()	Kontainerraren tamaina barnean dituen elementuen tamainara egokitzen du
Point getLocation() Point getLocationOnScreen()	Kontainerraren posizioa itzultzen du Kontainerrak pantailan duen posizioa itzultzen du

Beste klase erabilgarri batzuk

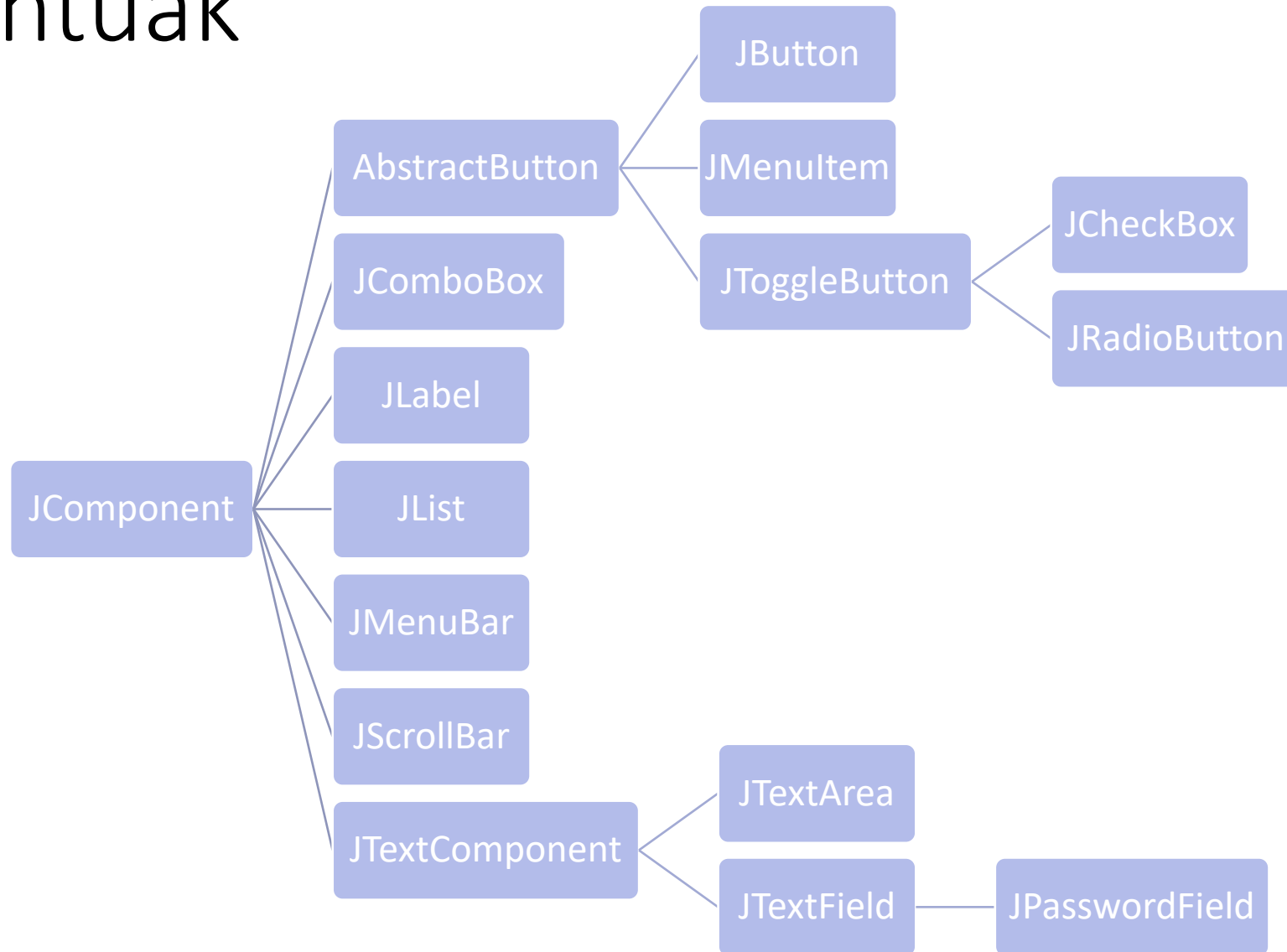
Klasea	Eraikitzailerik / Metodoak
Dimension	<code>new Dimension(zabalera, altuera)</code>
Point	<code>new Point(x, y)</code>
Rectangle	<code>new Rectangle(x, y, zabalera, altuera)</code>
Color	<code>new Color(int gorria, int berdea, int urdina)</code> : 0-255 arteko zenbakiak <code>new Color(int konstantea)</code> : Kolore konstante bat pasatzen zaio, <code>Color.WHITE...</code>
Font	<code>new Font("Helvetica", Font.BOLD + Font.ITALIC, 18)</code> Konstanteak: <code>Font.BOLD</code> , <code>Font.ITALIC</code> eta <code>Font.PLAIN</code> Beste metodo batzuk: <code>getName()</code> , <code>getStyle()</code> , <code>getSize()</code>
Cursor	<code>new Cursor(Cursor.HAND_CURSOR)</code>
Border	<code>new BorderFactory.createLineBorder(Color.black)</code>
ImageIcon	<code>new ImageIcon(fitxategi_izena)</code>

Elementuak

- Interfaze grafikoko objektu sinpleak
- Kontainer baten barruan kokatu behar dira bistaratu ahal izateko
- javax.swing.JComponent paketetik eratortzen dira denak
 - Kontainerrak ere

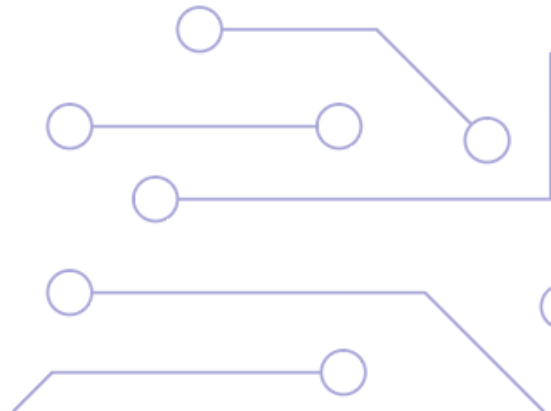
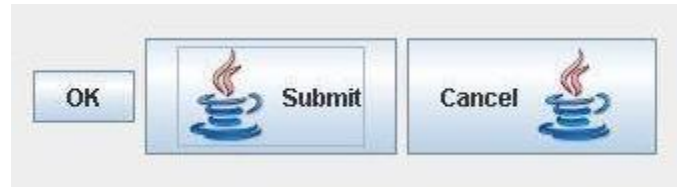


Elementuak



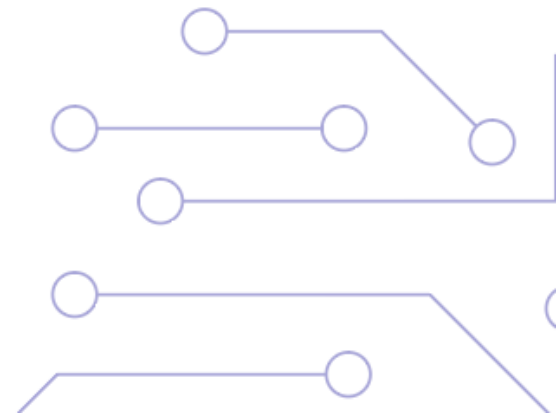
Elementuak: JButton

- Testu bat, irudi bat edo biak izan ditzakeen botoia
- Metodoak:
 - JButton()
 - JButton(Icon)
 - JButton(String)
 - JButton(String, Icon)
 - void setText(String)
 - String getText()
 - setToolTipText(String)
 - String getToolTipText()
 - setIcon(Icon)
 - Icon getIcon()



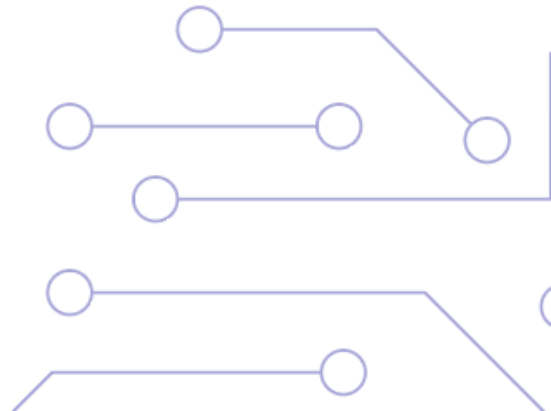
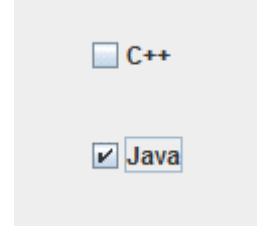
Elementuak: JToggleButton

- Bi egoera dituen botoia, sakatuta eta sakatu gabe
- Botoi normalak dituen metodoak eta bereak diren beste batzuk ditu
- Metodoak:
 - JToggleButton()
 - JToggleButton(Icon)
 - JToggleButton(Icon, boolean)
 - JToggleButton(String)
 - JToggleButton(String, boolean)
 - JToggleButton(String, Icon, boolean)
 - boolean isSelected()
 - void setSelected(boolean)



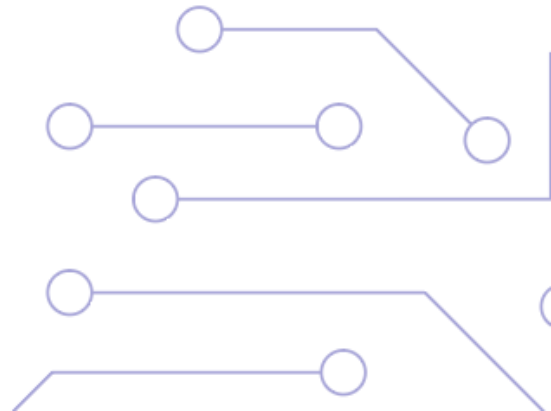
Elementuak: JCheckBox

- Aukera anitzak adierazteko erabiltzen da
- Metodoak:
 - JCheckBox()
 - JCheckBox(String)
 - JCheckBox(String, boolean)
 - boolean isSelected()
 - void setSelected(boolean)
 - String getLabel()
 - void setLabel(String)
 - boolean getState()
 - void setState(boolean)



Elementuak: JRadioButton

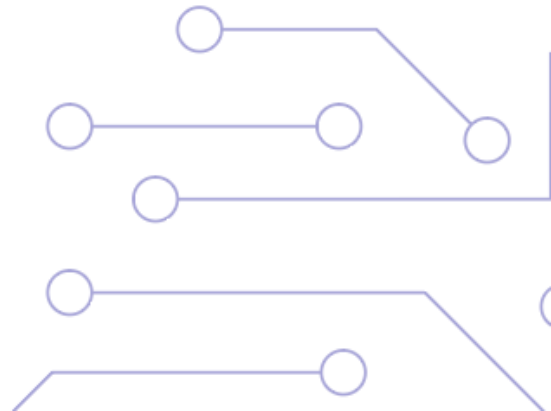
- Botoi multzo batean aukera bakarra hautatzen duten botoiak adierazteko erabiltzen da
- Metodoak:
 - JRadioButton()
 - JRadioButton(String)
 - JRadioButton(String, boolean)
 - JRadioButton(Icon)
 - JRadioButton(Icon, boolean)
 - JRadioButton(String, Icon)
 - JRadioButton(String, Icon, boolean)
 - boolean isSelected()
 - void setSelected(boolean)



Elementuak: JRadioButton

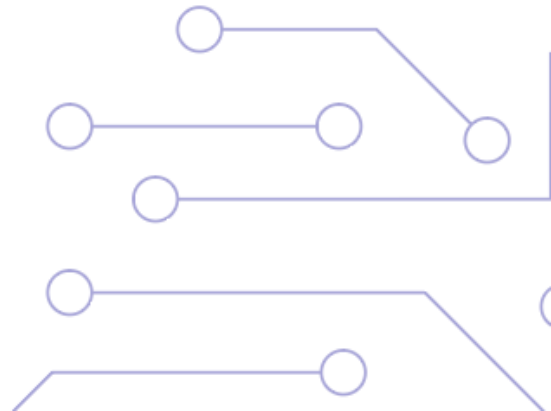
- Botoi multzoak ButtonGroup izena dauka
 - Ertza eta izena dauka
 - Multzoan JRadioButton-ak gehitzeko *add* metodoa erabiltzen da

```
ButtonGroup group = new ButtonGroup();  
group.add(radiobutton1);  
group.add(radiobutton2);  
group.add(radiobutton3);
```



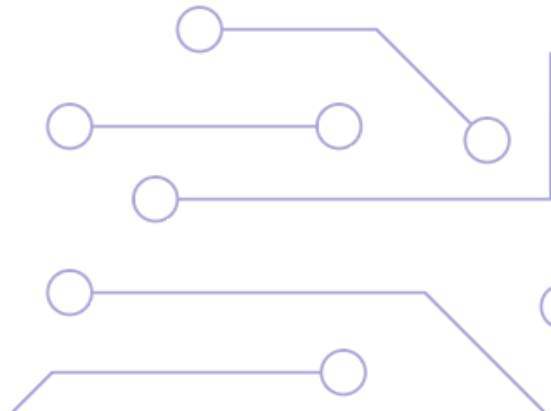
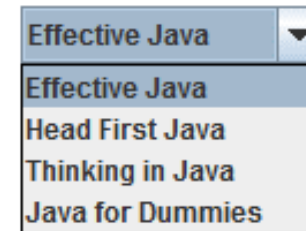
Elementuak: JComboBox

- Goitik behera irekitzen den eta aukera desberdinak erakusten dituen menua sortzen du
 - Aukera bakarra hautatu daiteke eta editagarria izan daiteke ala ez
- Metodoak:
 - JComboBox()
 - JComboBox(Object[])
 - JComboBox(ComboBoxModel)
 - Object getSelectedItem()
 - int getSelectedIndex()
 - void addItem(Object)



Elementuak: JComboBox

- Metodoak:
 - void remove(int)
 - void removeAll()
 - void setSelectedIndex(int)
 - Object getItemAt(int)
 - int getItemCount()
 - void setEditable(boolean)
 - void setModel(ComboBoxModel)



Elementuak: JComboBox

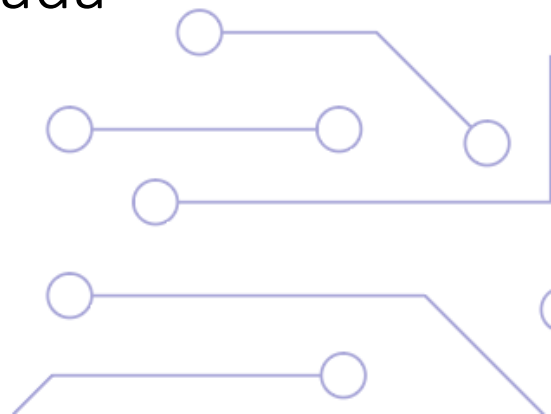
```
NireLeiho()  
{  
    super("Combo box");  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    setPreferredSize(new Dimension(300,200));  
    setLayout(new FlowLayout(FlowLayout.CENTER));  
    String[] months={"urtarrila", "otsaila", "martxoa", "apirila", "maiatza",  
"ekaina", "uztaila", "abuztua", "iraila", "urria", "azaroa", "abendua"};  
    JComboBox combo = new JComboBox(months);  
    combo.setSelectedIndex(4);  
    add(combo);  
    pack();  
    setVisible(true);  
}
```

Elementuak: JComboBox

```
ComboKlase(){
    JComboBox combo = new JComboBox();
    DefaultComboBoxModel model = new DefaultComboBoxModel();
    combo.setModel(model);
    model.addElement(new Talde("Aukeratu", "noId"));
    model.addElement(new Talde("Metallica", "met"));
    model.addElement(new Talde("Megadeth", "meg"));
    model.addElement(new Talde("Sepultura", "sep"));
    model.addElement(new Talde("Pantera", "pan"));
    combo.setBounds(10, 10, 120, 30);
    this.getContentPane().add(combo);
    pack();
    setVisible(true);
}
```

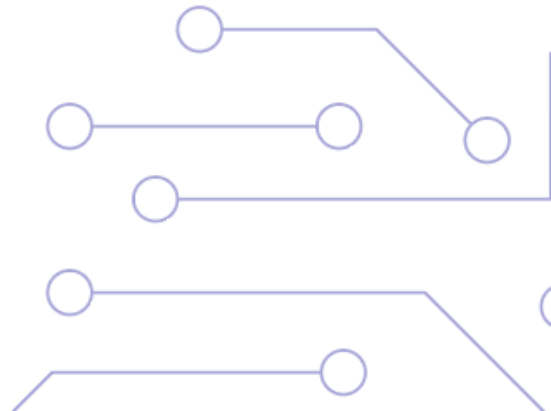
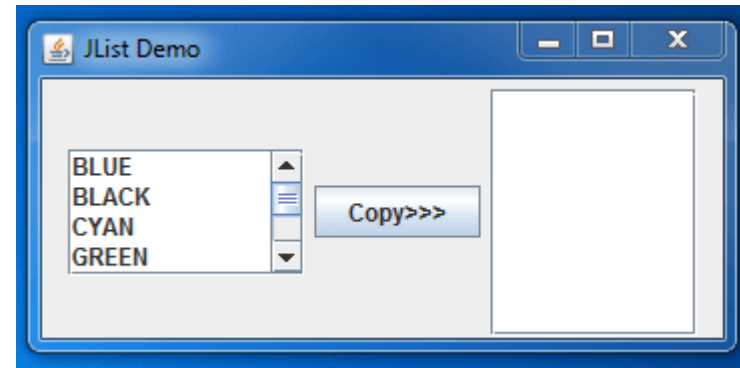
Elementuak: JList

- Hainbat elementu erakusten dituen lista bat da eta erabiltzaileak elementu bat edo gehiago aukeratu ditzake
- Hiru aukeraketa modu daude:
 - Bakarra aukeratzea
 - Tarte bakarra aukeratzea
 - Hainbat tarte aukeratzea
- Bere tamaina dituen elementuen kopuruarentzat txikiegia bada JScrollPane batean gehitu behar da scroll-a izan dezan



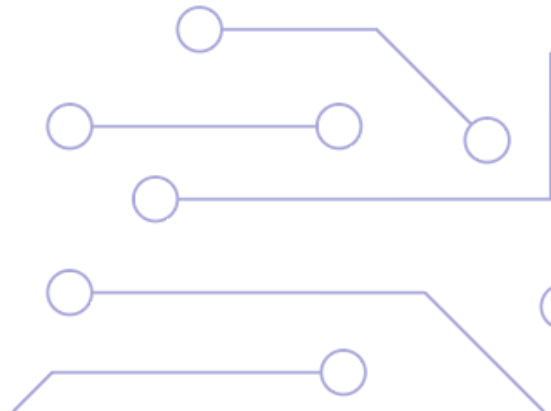
Elementuak: JList

- Metodoak:
 - JList()
 - JList(Object[])
 - JList(Vector)
 - JList(ListModel)
 - void setVisibleRowCount(int)
 - void setSelectedIndex(int)
 - void setSelectedIndices(int[])
 - boolean isIndexSelected(int)
 - void setListData (Vector)
 - void setModel(ListModel)



Elementuak: JList

- Metodoak:
 - void setSelectionMode(int selectionMode)
 - Non aukerak ListSelectionModel.SINGLE_SELECTION, ListSelectionModel.SINGLE_INTERVAL_SELECTION eta ListSelectionModel.MULTIPLE_INTERVAL_SELECTION diren
 - void remove(int)
 - void removeAll()
 - int getSelectedIndex()
 - Object getSelectedValue()
 - int[] getSelectedIndices()
 - Object[] getSelectedValues()
 - Vector getListData()
 - ListModel getModel()



Elementuak: JList

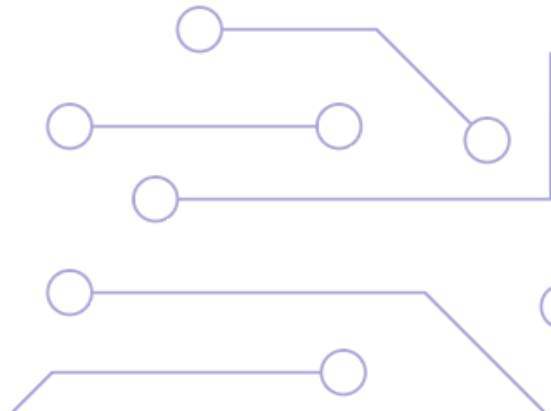
```
ListaKlase()  
{  
    DefaultListModel<String> estatuZerrenda = new DefaultListModel<>();  
    estatuZerrenda.addElement("Frantzia");  
    estatuZerrenda.addElement("Alemania");  
    estatuZerrenda.addElement("Espainia");  
    estatuZerrenda.addElement("Italia");  
    estatuZerrenda.addElement("Grezia");  
    estatuZerrenda.addElement("Portugal");  
    estatuZerrenda.addElement("Belgika");  
    estatuZerrenda.addElement("Irlanda");  
  
    add(new JList<>(estatuZerrenda));  
  
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    this.setTitle("JList adibidea");  
    this.setSize(400, 150);  
    this.setVisible(true);  
}
```

Elementuak: JList

```
ListaKlase()  
{  
    DefaultListModel<String> estatuZerrenda = new DefaultListModel<>();  
    estatuZerrenda.addElement("Frantzia");  
    estatuZerrenda.addElement("Alemania");  
    estatuZerrenda.addElement("Espainia");  
    estatuZerrenda.addElement("Italia");  
    estatuZerrenda.addElement("Grezia");  
    estatuZerrenda.addElement("Portugal");  
    estatuZerrenda.addElement("Belgika");  
    estatuZerrenda.addElement("Irlanda");  
    JList<String> zerrenda = new JList<>(estatuZerrenda);  
    add(new JScrollPane(zerrenda));  
    zerrenda.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);  
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    this.setTitle("JList adibidea");  
    this.setSize(400, 150);  
    this.setVisible(true);  
}
```

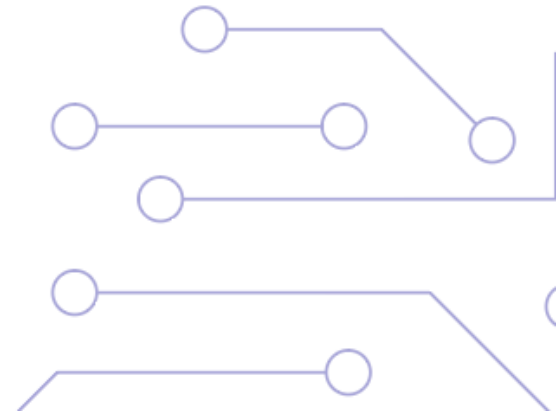
Elementuak: JLabel

- Editatu ezin daitekeen testu bat, irudi bat edo biak adierazteko elementua
 - Etiketa izena hartzen du
- Metodoak:
 - JLabel()
 - JLabel(String)
 - JLabel(String, LEFT|RIGHT|CENTER)
 - JLabel(Icon)
 - JLabel(String,Icon)
 - String getText()
 - void setText(String text)
 - void setIcon(Icon)
 - Icon getIcon()



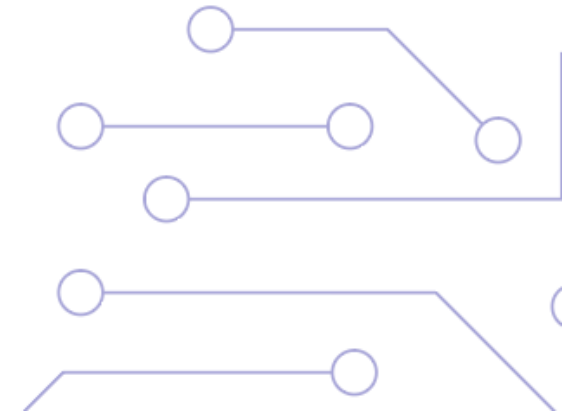
Elementuak: JTextComponent

- Bere barnean testu kutxa desberdinak sortzeko elementuak hartzen ditu
- Elementu guzti hauek zenbait metodo amankomun dituzte
 - String getText()
 - void setText(String)
 - boolean isEditable()
 - void setEditable(boolean)
 - .getSelectedText()
 - select(int,int)
 - selectAll()
 - getSelectionStart()
 - getSelectionEnd()
 - setSelectionStart(int)
 - setSelectionEnd(int)



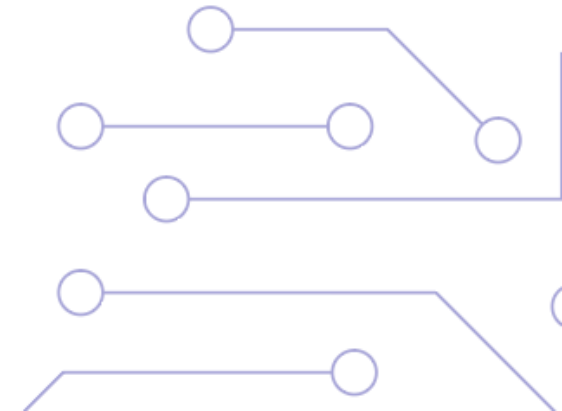
Elementuak: JTextField

- Editagarria den testu kutxa bat gehitzen du
- Metodoak:
 - JTextField()
 - JTextField(String)
 - JTextField(String, int)
 - String getText()
 - void setText(String text)
 - int getColumns()
 - void setColumns (int)



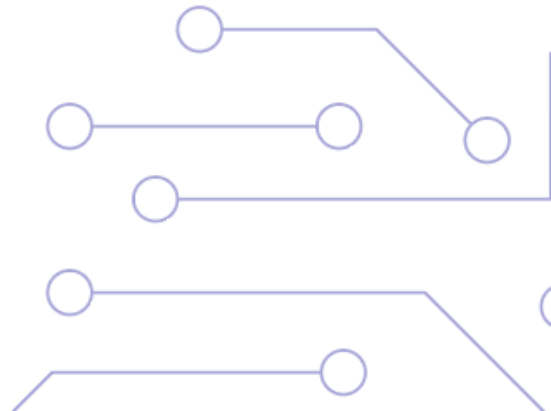
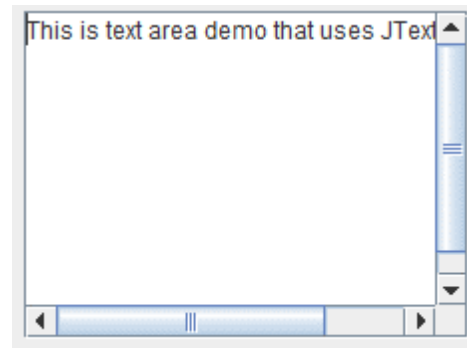
Elementuak: JPasswordField

- Aurrekoaren elementu berdina, baina idazten dena ezin da ikusi
- Metodoak:
 - `char[] getPassword()`: metodo hau `getText()` beharrean erabiltzen da
 - `setEchoChar('char')`



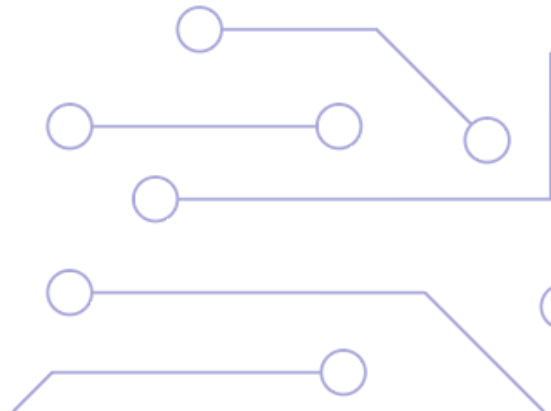
Elementuak: JTextArea

- Hainbat lerroko testu bat erakutsi edo idatzi behar denerako erabiltzen da
- Bere tamaina duen testuarentzat txikiegia bada JScrollPane batean gehitu behar da scroll-a izan dezan, bestela ez da ondo ikusiko
- Metodoak:
 - JTextArea()
 - JTextArea(String)
 - JTextArea(int, int)
 - JTextArea(String,int,int)
 - String getText()
 - void setText(String text)



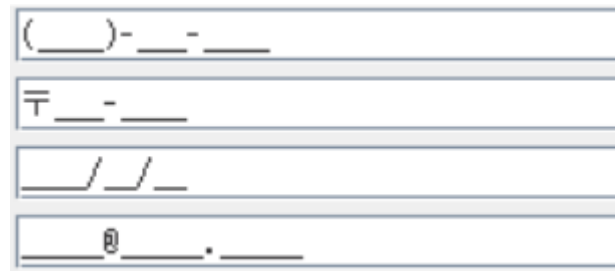
Elementuak: JTextArea

- Metodoak:
 - `int getColumns()`
 - `void setColumns(int)`
 - `int getRows()`
 - `void setRows(int)`
 - `void append(String)`
 - `void insert(String, int)`
 - `void replaceRange(String, int, int)`
 - `void setLineWrap(boolean)`



Elementuak: JFormattedTextField

- JTextField elementuaren antzekoa da, baina maskara edo formatu zehatz bat definitu daiteke sartu daitezkeen karaktereak mugatzeko
 - Egokia datak, portzentaiak, email helbideak, etab.



The image displays four examples of JFormattedTextField masks, each in a separate text box:

- Phone number mask: ()- - -
- Currency mask: ¤ - -
- Date mask: / / -
- Email mask: @ . - -

