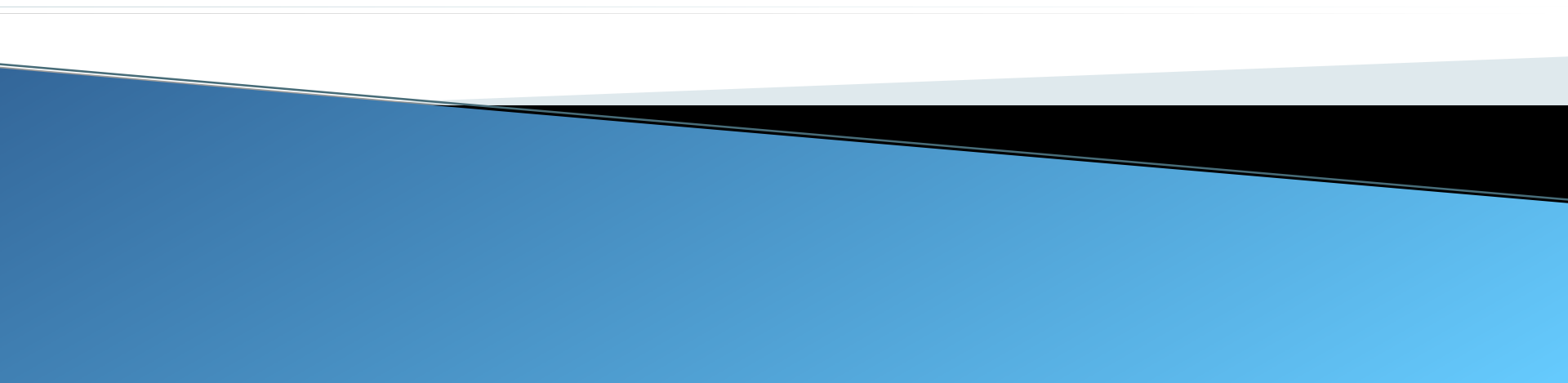


Gertaerei bideratutako programazioa

SOFTWARE INGENIARITZA



Gertaerei bideratutako programazioa

- ▶ Programaren egitura eta exekuzioa, sistemako gertaeren menpe.
- ▶ Erabiltzaileak edo programak berak sor ditzake gertaerak.
- ▶ Progaman definituko dira **gertaera** baten aurrean egin beharrekoak; hots, **ekintzak**.

Gertaeren kudeaketa

- ▶ Interfaze grafiko batetan:

Erabiltzaile ekintzak ➡ **gertaerak**

Gertaerak prozesatzeko ➡ **ekintzak**

Gertaeren kudeaketa

► Gertaera bat

- erabiltzailearen elkarrekintzak sortua
- interfazearen osagaien batekin lotua
- Adibideak: tekla bat sakatu, xagua mugitu, leiho bat itxi, leiho bat minimizatu, botoi bat sakatu...

Gertaeren kudeaketa

```
import javax.swing.*;

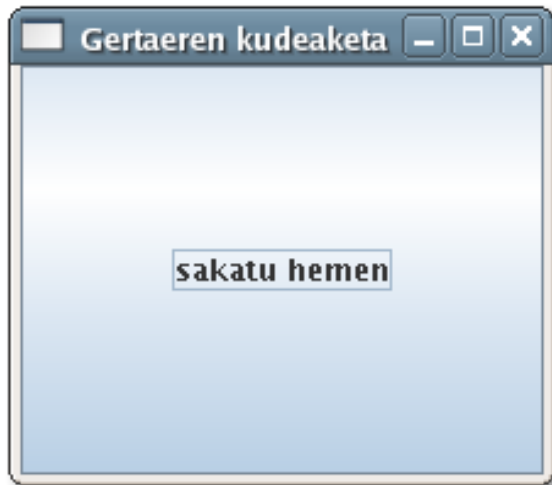
public class SimpleGUI extends JFrame {

    JButton button;

    public void ekin(){
        button = new JButton("sakatu hemen");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().add(button);
        setSize(300, 300);
        setVisible(true);
    }

    public static void main(String[] args){
        SimpleGUI frame = new SimpleGUI();
        frame.setTitle("gertaeren kudeaketa");
        frame.ekin();
    }
}
```

Gertaeren kudeaketa



Erabiltzaileak botoia noiz sakatzen duen jakin nahi dugu.

Botoia sakatzerakoan gertaera bat sortuko da.

Zer egin botoia sakatzerakoan ekintza burutu gura badugu?

- 1) Metodo bat programatu, gertaera kudeatzeko
- 2) Gertaera noiz sortu den jakin

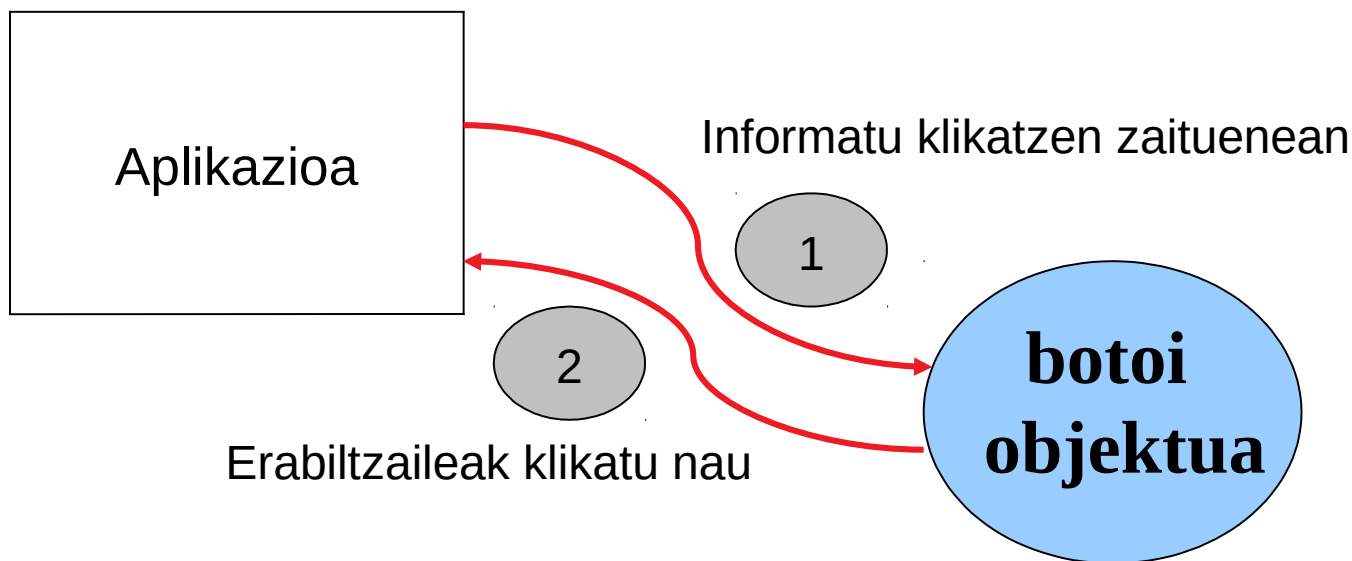
Gertaeren kudeaketa

1 Gertaera kudeatzeko metodoa (ekintza):

```
public void textuaAldatu(){  
    button.setText("ok! botoia sakatu duzu");  
}
```

2 Nola jakin daiteke erabiltzaileak botoia noiz sakatu duen?

Gertaeren kudeaketa



Gertaeren kudeaketa

Aplikazioa

Informatu klikatzen
zaituenean

1

botoi
objektua

```
public class SimpleGUI extends JFrame{
```

```
    JButton button;
```

```
    public void ekin(){
```

```
        button = new JButton("sakatu hemen");
```

```
        button.addActionListener(this);
```

```
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        getContentPane().add(button);
```

```
        setSize(300,300);
```

```
        setVisible(true);
```

```
    }
```

```
}
```

Gertaera “entzun”:
- Botoia noiz ikutu da?

Gertaeren kudeaketa

Botoi baten gertaerei buruz informatuta egoteko
ActionListener interfazea implementatu

```
public class SimpleGUI extends JFrame implements ActionListener {
```

```
    JButton button;
```

```
    public void ekin(){  
        button = new JButton("sakatu hemen");  
        button.addActionListener(this);  
    }
```

```
    public void actionPerformed(ActionEvent e) {  
        button.setText("ados! botoia sakatu duzu!");  
    }
```

```
}
```

Aplikazioa

Erabiltzaileak
klikatu egin nau!

2

botoi
objektua

Ekintza gauzatu:
- Testua aldatu

Listener interfazeak

- ▶ Gertaerak kudeatzeko, javak **Listener** interfaze “entzuleak” ditu
 - Implementatu beharreko metodoak.
- ▶ Metodo bakoitzaren implementazioak gertaera bakoitzari erantzun egokia eman.
- ▶ Objektu grafiko baten gaineko hainbat gertaera kontrolatzeko, *listener* bat esleitu:

objGraf.addXXXListener(objListener)



Gertaeren kudeaketa: 1. bertsioa

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

```
public class SimpleGUI extends JFrame implements ActionListener {
    JButton button;
```

```
    public void ekin(){
        button = new JButton("sakatu hemen");
        button.addActionListener(this);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().add(button);
        setSize(300,300);
        setVisible(true);
    }
```

```
    public static void main(String[] args){
        SimpleGUI frame = new SimpleGUI();
        frame.ekin();
    }
```

```
    public void actionPerformed(ActionEvent e){ button.setText("ados! botoia sakatu duzu!"); }
}
```

JFrame implements ActionListener

**Ekintza klase nagusiaren
metodo bat da**

Gertaeren kudeaketa: 2. bertsioa

```
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class Aukerak2 extends JFrame{

    JLabel jLabel1 = new JLabel("Palkoa:");
    JLabel jLabel2 = new JLabel("Aukeratu duzun palkoa:");
    JLabel emaitza = new JLabel();
    JRadioButton goian = new JRadioButton("Goian", true);
    JRadioButton behean = new JRadioButton("Behean", false);
    ButtonGroup bg = new ButtonGroup();

    public Aukerak2() {
        super("Gertaerak kudeatzen...");
    }
}
```

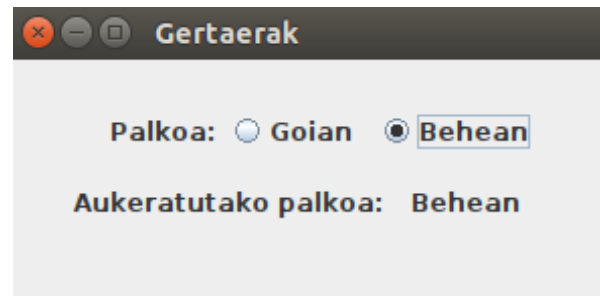
```
public void go(){
    bg.add(goian);
    bg.add(behean);
    goian.addActionListener(new Kontroladore());
    behean.addActionListener(new Kontroladore());
    this.getContentPane().setLayout(new FlowLayout());
    getContentPane().add(jLabel1, null);
    getContentPane().add(goian, null);
    getContentPane().add(behean, null);
    getContentPane().add(jLabel2, null);
    getContentPane().add(emaitza, null);
    setSize(300, 200);
    setVisible(true);
}
```

```
public class Kontroladore implements ActionListener {

    public void actionPerformed(ActionEvent e) {
        emaitza.setText(e.getActionCommand());
    }
}
```

```
public static void main(String[] args){
    Aukerak2 proba = new Aukerak2();
    proba.go();
    proba.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
}
```

**Ekintza edozein elementok
implementatu: barne klase bat da**

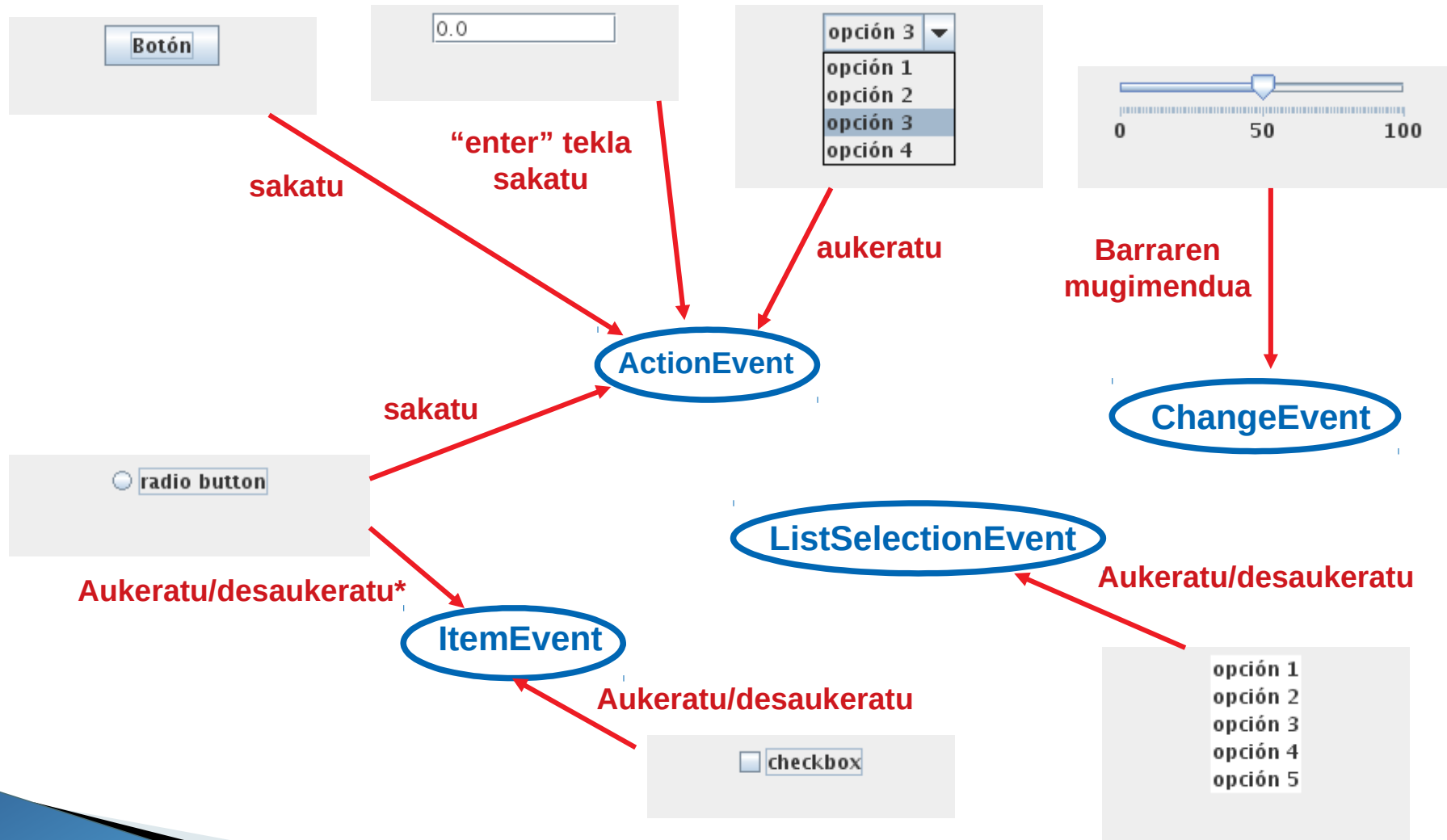


Gertaeren kudeaketa: 3. bertsioa

```
public void go() {  
    bg.add(goian);  
    bg.add(behean);  
    goian.addActionListener(new ActionListener() {  
        public void actionPerformed(ActionEvent e) {  
            emaitza.setText(e.getActionCommand());  
        }  
    });  
  
    behean.addActionListener(new ActionListener() {  
        public void actionPerformed(ActionEvent e) {  
            emaitza.setText(e.getActionCommand());  
        }  
    });  
    this.getContentPane().setLayout(new FlowLayout());  
    getContentPane().add(jLabel1, null);  
    getContentPane().add(goian, null);  
    getContentPane().add(behean, null);  
    getContentPane().add(jLabel2, null);  
    getContentPane().add(emaitza, null);  
    setSize(300, 200);  
    setVisible(true);  
}
```

Ekintza edozein elementok
implementatu: **klase anonimo**
bat da

Gertaera motak



(*) gutxi erabiliak

Gertaera kudeatzaileak

Gertaerei erantzuteko, beraiei dagokien interfazea implementatzen duen **klase kudeatzaile** baten bidez

| Gertaera | Interfaze kudeatzailea | Metodo kudeatzailea |
|--------------------|------------------------|---|
| ActionEvent | ActionListener | void actionPerformed(ActionEvent e) |
| ChangeEvent | ChangeListener | void stateChanged(ChangeEvent e) |
| ItemEvent | ItemListener | void itemStateChanged(ItemEvent evt) |
| ListSelectionEvent | ListSelectionListener | void valueChanged(ListSelectionEvent evt) |

Gertaerek `java.util.EventObject`-tik gertaera sortutako osagaia itzultzen duen `Object getSource()` erabili.

Gertaerak

| Osagai Grafikoa | Gertaera | Entzulea | Metodoak |
|------------------------|----------------|-------------------|---|
| Jbutton, JtextField... | ActionEvent | ActionListener | actionPerformed(ActionEvent) |
| Osagaiak | ComponentEvent | ComponentListener | componentHidden(ComponentEvent) componentMoved(ComponentEvent) componentResized(ComponentEvent) componentShown(ComponentEvent) |
| Osagaiak | FocusEvent | FocusListener | focusGained(FocusEvent) focusLost(FocusEvent) |
| Osagaiak | KeyEvent | KeyListener | keyPressed(KeyEvent) keyReleased(KeyEvent) keyTyped(KeyEvent) |

Gertaerak

| Osagai Grafikoa | Gertaera | Entzulea | Metodoak |
|-----------------|------------|---------------------|--|
| Osagaiak | MouseEvent | MouseListener | MouseClicked(MouseEvent) MouseEntered(MouseEvent) MouseExited(MouseEvent) MousePressed(MouseEvent) MouseReleased(MouseEvent) MouseDragged(MouseEvent) MouseMoved(MouseEvent) |
| | | MouseListener | MouseClicked(MouseEvent) MouseEntered(MouseEvent) MouseExited(MouseEvent) MousePressed(MouseEvent) MouseReleased(MouseEvent) |
| | | MouseMotionListener | MouseDragged(MouseEvent) MouseMoved(MouseEvent) |

Gertaerak

Osagai Grafikoa

Gertaera

Entzulea

Metodoak

Edukiontzia

ContainerEvent

ContainerListener

ComponentAdded(ContainerEvent)
ComponentRemoved(ContainerEvent)

Leihoa

WindowEvent

WindowListener

windowActivated(WindowEvent)
windowClosed(WindowEvent)
windowClosing(WindowEvent)
windowDeactivated(WindowEvent)
windowDeiconified(WindowEvent)
windowIconified(WindowEvent)
windowOpened(WindowEvent)

Listener-en Adapterrak

- ▶ Kasu gehienetan, ez ditugu osagai baten gertaera posible guztiak kudeatu behar.
- ▶ Gertaera kudeatzaileen inplementazioa sinplifikatzeko, “Adapter”-ak erabili.
- ▶ Adapter klase abstaktu batek Listener interfaza inplementatu:
 - Beharreko metodoa soilik inplementatu
 - Besteak hutsik ditu

Listener-en Adapterrak

```
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.swing.JFrame;

public class Sketcher {
    JFrame window = new JFrame("Sketcher");
    public Sketcher() {
        window.setBounds(30, 30, 300, 300);
        window.addWindowListener(new WindowHandler());
        window.setVisible(true);
    }

    private class WindowHandler extends WindowAdapter {
        public void windowClosing(WindowEvent e) {
            System.out.println("closing");
            window.dispose(); // Release the window resources
            System.exit(0); // End the application
        }
    }

    public static void main(String[] args) {
        new Sketcher();
    }
}
```

**Beharrezko
metodoak soilik
implementatu**