



## Tópico 9: Intro GUI y Swing

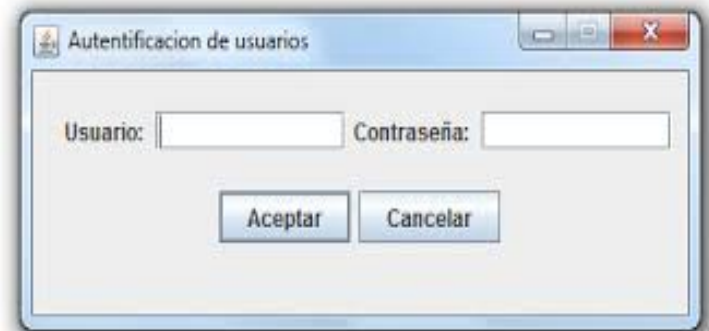
Semestre: 01/16

No. Serie

# Contenidos

---

- Concepto de GUI
- GUI y Java
- Java y el paquete Swing
- Modelo de Eventos
- Interfaces para gestión de Eventos



# Recordando...

---

- Concepto de herencia
- Utilidad de la herencia en la POO
- Implementación de la herencia en Java
- La herencia como relación entre clases con UML

# Java y GUI

- Java cuenta con un set de *packages* que permiten desarrollar aplicaciones que faciliten la interacción con el usuario mediante GUI o “ventanas”.

```
Enter the Operation to be Performed
1.New Account
2.Deposit Money
3.Withdraw Money
4.Search
5.Exit
1
Enter Name:
Sashank
Enter Type:
1
Enter Starting Deposit:
3000
Account No is: 01
Enter the Operation to be Performed
1.New Account
2.Deposit Money
3.Withdraw Money
4.Search
5.Exit
4
Enter Account No
02
Invalid Details
Enter the Operation to be Performed
1.New Account
2.Deposit Money
3.Withdraw Money
4.Search
5.Exit
2
Enter Account No
1
Enter Amount
500
Deposit Successful
Enter the Operation to be Performed
1.New Account
2.Deposit Money
3.Withdraw Money
4.Search
5.Exit
4
Enter Account No
1
Name: Sashank
Current Balance: 3500.0
Enter the Operation to be Performed
1.New Account
2.Deposit Money
3.Withdraw Money
4.Search
5.Exit
```

v/s

# Java y GUI

---

- Para esto se cuenta con el package javax.swing.\*
- Corresponde a la Java Foundation Classes (JFC)
  - originalmente parte de un proyecto alternativo para crear bibliotecas Java un tanto más simples, flexibles y prácticas.
- Desde el JDK versión 1.2 se incluyó de manera oficial en el lenguaje.
- Posee un conjunto de objetos gráficos para crear GUI o “ventanas”

Más detalles de swing

<http://www.javatpoint.com/java-swing>

# Java y GUI

---

- Principales características Java-swing
  - hecha 100% en Java y para Java.
  - “no pide nada prestado” al SO
    - se dibuja el objeto sobre la ventana como se requiera.
  - se mejoró bastante la independencia del SO
    - lo único que se pide al SO es dibujar una ventana, sobre la cual se dibuja
    - agrega más portabilidad a nuestras aplicaciones

# Java y GUI

---

- Entre los elementos disponibles para trabajar con Java y swing, se tiene:
  - JButton, JTextField, JLabel, Choice, JCheckBox, JPanel, Window, JDialog, JFileDialog, JFrame, JList, JScrollbar, JTextArea, JComboBox, etc.
- Ok...
  - pero ¿Cómo se crea una GUI?

# Java y GUI

---

- Para crear las ventanas, se debe considerar:
  - paquetes a importar
  - manejo de eventos e interfaces
  - objetos gráficos que contendrá la ventana

Interfaces para  
gestión de eventos

```
import java.awt.*;
```

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

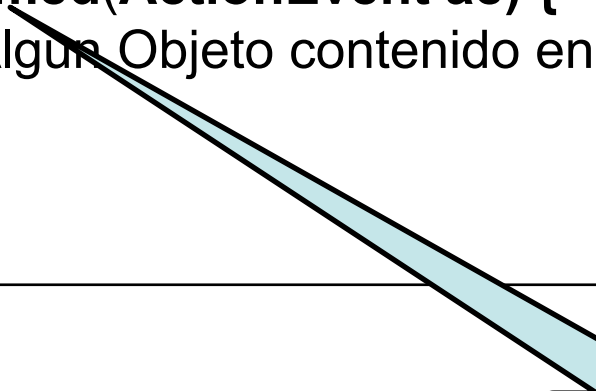
```
import javax.swing.*;
```

```
class MiVentana extends JFrame implements XXXXListener {  
    ....OBJETOS gráficos contenidos en la GUI ....  
}
```



# Java y GUI

```
class MiVentana extends JFrame implements ActionListener {  
    .....  
    public void actionPerformed(ActionEvent ae) {  
        if (ae.getSource() == Algun Objeto contenido en MiVentana) {  
            hacer algo....  
        } .....  
    }  
}
```



Procesa el  
Evento capturado

- En el constructor de MiVentana, se debe decir a los objetos que se preparen para responder a los eventos:  
**algunObjeto.addXXXXXListener(this);**

# Java y GUI

| Event, Listener Interface y Métodos para vincular objeto que escucha el evento        | Algunos componentes que generan este tipo de eventos   |
|---|--|
| <b>ActionEvent</b><br><b>ActionListener</b><br><b>addActionListener()</b>             | <b>JButton, JList, JTextField, JMenuItem, JCheckBoxMenuItem, JMenu, JPopupMenu.</b>  |
| <b>AdjustmentEvent</b><br><b>AdjustmentListener</b><br><b>addAdjustmentListener()</b> | <b>JScrollbar y cualquier objeto que implemente la interface Adjustable.</b>   |
| <b>ComponentEvent</b><br><b>ComponentListener</b><br><b>addComponentListener()</b>    | <b>Component, JButton, JCanvas, JCheckBox, JComboBox, Container, JPanel, JApplet, JScrollPane, Window, JDialog, JFileDialog, JFrame, JLabel, JList, JScrollbar, JTextArea, JTextField.</b> |
| <b>ContainerEvent</b><br><b>ContainerListener</b><br><b>addContainerListener()</b>    | <b>Container, JPanel, JApplet, JScrollPane, Window, JDialog, JFileDialog, JFrame.</b>  |
| <b>FocusEvent</b><br><b>FocusListener</b><br><b>addFocusListener()</b>                | <b>Component, JButton, JCanvas, JCheckBox, JComboBox, Container, JPanel, JApplet, JScrollPane, Window, JDialog, JFileDialog, JFrame, JLabel, JList, JScrollbar, JTextArea, JTextField.</b> |

# Java y GUI

| Listener Interface y Adapter                            | Métodos   |
|---|---|
| <b>ActionListener</b>                                   | <b>actionPerformed(ActionEvent)</b>   |
| <b>AdjustmentListener</b>                               | <b>adjustmentValueChanged(AdjustmentEvent)</b>  |
| <b>ComponentListener</b><br><b>ComponentAdapter</b>     | <b>componentHidden(ComponentEvent)</b><br><b>componentShown(ComponentEvent)</b><br><b>componentMoved(ComponentEvent)</b><br><b>componentResized(ComponentEvent)</b>   |
| <b>ContainerListener</b><br><b>ContainerAdapter</b>     | <b>componentAdded(ContainerEvent)</b><br><b>componentRemoved(ContainerEvent)</b>  |
| <b>FocusListener</b><br><b>FocusAdapter</b>             | <b>focusGained(FocusEvent)</b><br><b>focusLost(FocusEvent)</b>  |
| <b>KeyListener</b><br><b>KeyAdapter</b>                 | <b>keyPressed(KeyEvent)</b><br><b>keyReleased(KeyEvent)</b><br><b>keyTyped(KeyEvent)</b>  |
| <b>MouseListener</b><br><b>MouseAdapter</b>             | <b>mouseClicked(MouseEvent)</b><br><b>mouseEntered(MouseEvent)</b><br><b>mouseExited(MouseEvent)</b><br><b>mousePressed(MouseEvent)</b><br><b>mouseReleased(MouseEvent)</b>   |
| <b>MouseMotionListener</b><br><b>MouseMotionAdapter</b> | <b>mouseDragged(MouseEvent)</b><br><b>mouseMoved(MouseEvent)</b>  |
| <b>WindowListener</b><br><b>WindowAdapter</b>           | <b>windowOpened(WindowEvent)</b><br><b>windowClosing(WindowEvent)</b><br><b>windowClosed(WindowEvent)</b><br><b>windowActivated(WindowEvent)</b><br><b>windowDeactivated(WindowEvent)</b><br><b>windowIconified(WindowEvent)</b><br><b>windowDeiconified(WindowEvent)</b> |
| <b>ItemListener</b>                                     | <b>itemStateChanged(ItemEvent)</b>  |

# Java y GUI

---

- Para poder “ejecutar” la ventana

```
public class TestVentana {  
    public static void main(String[ ] args) {  
        MiVentana miV = new MiVentana( );  
        miV.setVisible(true);  
        .....  
    }  
    ....  
}
```

Aplicar todo lo anterior y analice el archivo  
ButtonDemo.java  
¿Qué hace?  
¿Qué evento maneja?  
¿Qué interface implementa?

# Resumiendo

---

- Concepto de GUI con Java
- Java y el paquete Swing
- Modelo de Eventos
- Interfaces para gestión de Eventos
  - Implementando interfaces de tipo XXXListener



## Tópico 9: Intro GUI y Swing

Semestre: 01/16

No. Serie