Universidad Politécnica de Chiapas

INVESTIGACIÓN NO.1

Programación Visual

INGENIERIA DE DESARROLLO DE SOFTWARE

Eduardo Marcelino Díaz Díaz MATRICULA:183442

30 DE ABRIL DE 2019

Clases contenedoras (hijas de Container)

1. Panel: Permite hacer una presentación más avanzada que Container mediante la combinación con subpaneles o subclases para crear contenedores personalizados. La clase Applet que sirve para crear applets Java, hereda de esta clase Panel.
2. ScrollPane: Una barra de desplazamiento, horizontal o vertical.
3. Window: Una ventana sin borde.
4. Frame: Una ventana que no tiene borde. Puede tener asociado un objeto Menubar (una barra de herramientas o barra de menú personalizada).
5. Dialog: Una ventana usada para crear diálogos. Tiene la capacidad de ser modal con lo que sólo este contenedor recibiría entradas del usuario.
6. Filedialog: Un diálogo que usa el selector de archivos nativo del sistema operativo.
7. jframe : sirve para crear ventanas, como la ventana principal

Clases componentes (hijas directas de Component)

1. Button: Un botón gráfico para el que se puede definir una acción que sucederá cuando se presione el botón.
2. Canvas: Permite pintar o capturar eventos del usuario. Se puede usar para crear gráficos o como clase base para crear una jerarquía de componentes personalizados.
3. Checkbox: Soporta dos estados: on y off. Se pueden asociar acciones que se ejecuten (triggers) cuando el estado cambie.
4. Choice: Menú desplegable de opciones.
5. Label: Cadena de etiqueta en una localización dada.
6. List: Una lista desplegable de cadenas.
7. Scrollbar: Desplegable de objetos Canvas.
8. TextComponent: Cualquier componente que permita editar cadenas de texto.Tiene dos clases hijas:
9. TextField: Componente de texto consistente en una línea que puede ser usada para construir formularios.
10. TextArea: Componente para edición de texto de tamaño variable.

Eventos físicos

- InputEvent: Se ha producido una entrada del usuario. Tiene como eventos hijos KeyEvent (pulsación de una tecla) y MouseEvent (acción sobre el ratón).
- FocusEvent: Avisa al programa de que el componente ha ganado o perdido la atención (enfoque) del usuario. Esto se deduce de la actividad del usuario (ratón y teclado).
- WindowEvent: Avisa al programa de que el usuario ha utilizado uno de los controles de ventana a nivel del sistema operativo, como los controles de minimizar o cerrar.
- ContainerEvent: Se envía cuando se añaden o eliminan componentes a un contenedor.
- PaintEvent: Evento especial que señala que el sistema operativo quiere dibujar de nuevo una parte de la interfaz. Un componente debe sobreescribir el método paint() o el método update() para gestionar este evento.

Eventos semánticos

Son todos hijos del evento AWTEvent, que es el evento base de la jerarquía de eventos:

- ActionEvent: Avisa al programa de acciones específicas de componentes como las pulsaciones de botones.
- AdjustmenteEvent: Comunica que una barra de desplazamiento ha sido ajustada.
- ItemEvent: Avisa al programa cuando el usuario interacciona con una elección, una lista o una casilla de verificación.
- TextEvent: Avisa cuando un usuario cambia texto en un componente TextComponent, TextArea o TextField.
- InputMethodEvent: Avisa que un texto que está siendo creado utilizando un método de entrada está cambiando (se ha escrito algo más...).
- InvocationEvent: Este evento ejecuta el método run() en una clase Runnable cuando es tratado por el thread del despachador (dispatcher) de AWT.

Listenner

NOMBRE LISTENER	DESCRIPCIÓN	MÉTODOS	EVENTOS
ACTIONLISTENER	Se produce al hacer click en un componente, también si se pulsa Enter teniendo el foco en el componente.	public void actionPerformed(ActionEvent e)	▪ **JButton**: click o pulsar Enter con el foco activado en él. ▪ **JList**: doble click en un elemento de la lista. ▪ **JMenuItem**: selecciona una opción del menú. ▪ **JTextField**: al pulsar Enter con el foco activado.
KEYLISTENER	Se produce al pulsar una tecla. según el método cambiara la forma de pulsar la tecla.	public void keyTyped(KeyEvent e) public void keyPressed(KeyEvent e) public void keyReleased(KeyEvent e)	Cuando pulsamos una tecla, segun el Listener: ▪ **keyTyped**:al pulsar y soltar la tecla. ▪ **keyPressed**: al pulsar la tecla. ▪ **keyReleased**: al soltar la tecla.
FOCUSLISTENER	Se produce cuando un componente gana o pierde el foco, es decir, que esta seleccionado.	public void focusGained(FocusEvent e) public void focusLost(FocusEvent e)	Recibir o perder el foco.
MOUSELISTENER	Se produce cuando realizamos una acción con el ratón.	public void mouseClicked(MouseEvent e) public void mouseEntered(MouseEvent e) public void mouseExited(MouseEvent e) public void mousePressed(MouseEvent e) public void mouseReleased(MouseEvent e)	Según el Listener: ▪ **mouseCliked**: pinchar y soltar. ▪ **mouseEntered**: entrar en un componente con el puntero. ▪ **mouseExited**: salir de un componente con el puntero ▪ **mousePressed**: presionar el botón. ▪ **mouseReleased**: soltar el botón.
MOUSEMOTIONLISTENER	Se produce con el movimiento del mouse.	public void mouseDragged(MouseEvent e) public void mouseMoved(MouseEvent e)	Según el Listener: ▪ **mouseDragged**: click y arrastrar un componente. ▪ **mouseMoved**: al mover el puntero sobre un elemento