

Week 13 - LAQ's

Instructions

Explain key board events and its listener.

In Java, keyboard events are generated when a user interacts with the keyboard, such as pressing or releasing a key. These events are handled using the `KeyListener` interface, which is part of the `java.awt.event` package.

A `KeyListener` has three main methods to override:

- `keyPressed(KeyEvent e)`: This method is invoked when a key is pressed down. It allows you to respond immediately to the key press.
- `keyReleased(KeyEvent e)`: This method is called when a key is released. It can be used to perform actions after the key has been released.
- `keyTyped(KeyEvent e)`: This method is fired when a key press and release result in a character being typed. It is primarily used for character input.

Key Events Types

- **Key Typed Events**: These events are generated when a character is typed. They are useful for capturing input that corresponds to a character, such as letters and numbers.
- **Key Pressed Events**: These events occur when any key is pressed, including non-character keys like function keys (F1, F2, etc.) and control keys (Shift, Ctrl).

- Key Released Events: These events are triggered when a key is released, allowing you to detect when the user has stopped pressing a key.

Implementing KeyListener

To implement a KeyListener, you need to:

1. Implement the KeyListener Interface: Your class should implement the KeyListener interface.
2. Override the Methods: Provide implementations for the three methods mentioned above.
3. Register the Listener: Use the addKeyListener() method on the component (like a text field) that should listen for keyboard events.

Example Code

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

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

```
public class KeyEventDemo extends JFrame implements KeyListener {
```

```
    JTextField typingArea;
```

```
    public KeyEventDemo() {
```

```
        typingArea = new JTextField(20);
```

```
        typingArea.addKeyListener(this);
```

```
        add(typingArea);
```

```
        setSize(300, 200);
```

```
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }

    public void keyPressed(KeyEvent e) {
        System.out.println("Key Pressed: " + e.getKeyChar());
    }

    public void keyReleased(KeyEvent e) {
        System.out.println("Key Released: " + e.getKeyChar());
    }

    public void keyTyped(KeyEvent e) {
        System.out.println("Key Typed: " + e.getKeyChar());
    }

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

Focus and Key Events

- A component must have keyboard focus to receive key events. You can manage focus using methods like `setFocusTraversalKeysEnabled(false)` to control how focus is handled.