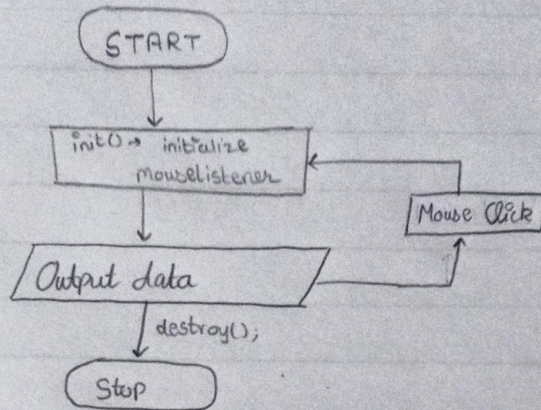


Practical No. 17

Aim: Create, debug and run Java programs based on mouse events and keyboard events.



Aim: Create, debug and run Java programs based on mouse events and keyboard events.

Theory:

What are Events in Java? (Event classes)?

- • All the events in Java have corresponding event classes associated with them.
- Each of these classes is derived from the one single super class, i.e. `EventObject`, it is contained in the `Java.util.` package.
- The event object contains the following two important methods for handling events:
 - 1) `getSource()`;
 - 2) `toString()`;

Java MouseListener Interface:

The java `MouseListener` is notified whenever you change the state of mouse. It is notified against `MouseEvent`. The `MouseListener` is found in `java.awt.event` package. It has five methods:

```
public abstract void mouseClicked(MouseEvent e);
public abstract void mouseEntered(MouseEvent e);
public abstract void mouseExited(MouseEvent e);
public abstract void mousePressed(MouseEvent e);
public abstract void mouseReleased(MouseEvent e);
```


in this class, we have two private methods: `mouseClicked` and `keyPressed`.

They:

- `mouseClicked` is called when the mouse is clicked on the window.
 - `keyPressed` is called when a key is pressed.
 - The code for these methods is as follows:
- ```

public void mouseClicked() {
 // Code for mouseClicked
}

public void keyPressed() {
 // Code for keyPressed
}

```

The above code is a simple Java program that demonstrates the use of `MouseListener` and `KeyListener` interfaces. It creates a window titled "MouseEventDemo" and adds a `JPanel` to it. The `JPanel` implements the `MouseListener` and `KeyListener` interfaces. The `mouseClicked` method prints "MouseClicked" to the console, and the `keyPressed` method prints "KeyPressed" to the console.

Conclusion: The above code demonstrates the use of `MouseListener` and `KeyListener` interfaces.

Hence, I created, debugged and executed java programs based on the concepts of `MouseEvent` and `KeyboardEvent`.