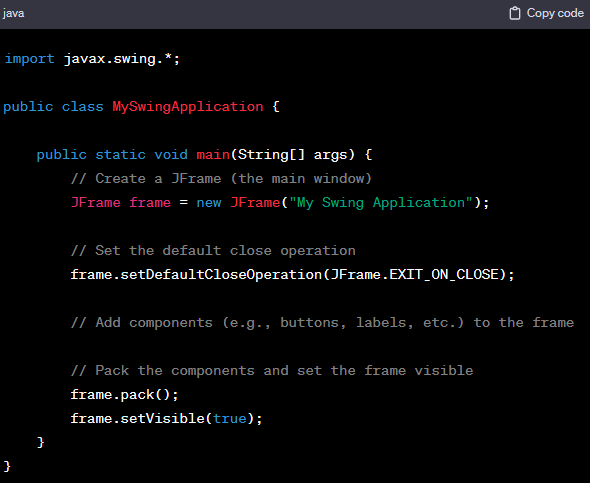
**JAVA SWING GUI**

Java Swing is a GUI (Graphical User Interface) toolkit for Java that allows developers to create rich desktop applications. It provides a set of components and containers to build interactive and user-friendly interfaces. Below, I'll provide you with a basic overview of a Swing module, along with some syntax and examples for common Swing components.

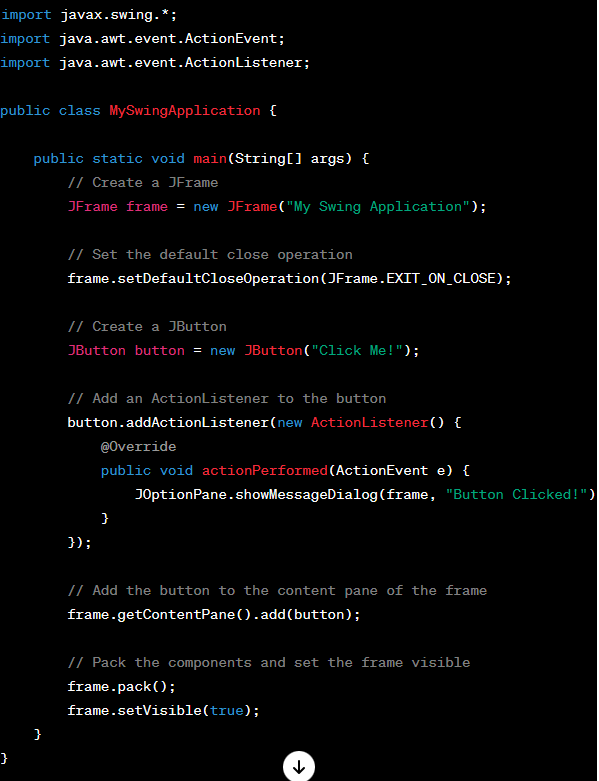
**Swing Module Syntax:**

In a Swing application, you typically create a **JFrame** (the main window) and add various components to it. Here's a basic structure:



**Example: Adding a JButton to the JFrame**

*Let's extend the example by adding a JButton to the JFrame:*

**

*In this example:*

We've added a **JButton** with the label **"Click Me!"** to the frame.

An ***‘ActionListener’*** is attached to the button, which displays a message dialog when the button is clicked.

**KEY SWING COMPONENTS:**

**Key Swing Components:**

1. **JFrame:** The main window of the application.



1. **JButton:** A clickable button.



1. **JLabel:** A non-editable text element.



1. **JTextField:** A single-line text input field.



1. **JTextArea:** A multi-line text input field.



1. **JPanel:** A container for organizing components.



These are just a few examples, and Swing provides many more components for building diverse GUIs. Remember to perform Swing-related operations on the Event Dispatch Thread (EDT) to ensure proper thread safety in GUI applications.

**Using JOptionPane in Java Swing**

Introduction:

**JOptionPane** is a class in the Java Swing library that provides a convenient way to display standard dialog boxes for interaction with the user. These dialog boxes can be used to show messages, gather input, or ask the user to make a decision. **JOptionPane** simplifies the process of creating dialog boxes in a Swing application.

Syntax:

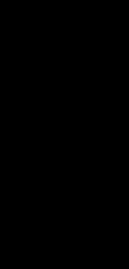
The basic syntax for displaying a message dialog using **JOptionPane** is as follows:

JOptionPane.showMessageDialog(parentComponent, message, title, messageType);

* ***parentComponent***: The component that determines the position of the dialog. It can be a **JFrame**, **JDialog**, or **null** for the default frame.
* ***message***: The message to be displayed in the dialog.
* ***title***: The title of the dialog.
* ***messageType***: The type of message, such as **JOptionPane.PLAIN\_MESSAGE**, **JOptionPane.INFORMATION\_MESSAGE**, **JOptionPane.QUESTION\_MESSAGE**, **JOptionPane.WARNING\_MESSAGE**, or **JOptionPane.ERROR\_MESSAGE**.

**Example: Message Dialog**

Let's create a simple Swing application that displays a message dialog when a button is clicked:





In this example:

* We've added a **JButton** labeled "Show Message Dialog" to the frame.
* An **ActionListener** is attached to the button, which shows a message dialog when the button is clicked.

Key **JOptionPane** Methods:

1. **showMessageDialog:** Displays a message dialog.

JOptionPane.showMessageDialog(parentComponent, message, title, messageType);

2. **showInputDialog:** Displays an input dialog to prompt the user for input

JOptionPane.showMessageDialog(parentComponent, message, title, messageType);

3. **showConfirmDialog:** Displays a confirmation dialog to ask the user for a yes/no/cancel decision.

JOptionPane.showMessageDialog(parentComponent, message, title, messageType);

Homework Assignment:

1. Create a new Swing application.
2. Use **JOptionPane** to ask the user for their name using **showInputDialog**.
3. Display a message dialog (with their name) using **showMessageDialog**.
4. Implement a confirmation dialog using **showConfirmDialog** to ask if they are sure about proceeding.
5. Display an appropriate message based on their response.

Feel free to experiment with different message types and dialog options provided by **JOptionPane**.