Sources:

MS Bing Chat Copilot

A StackPane in JavaFX is a layout container that stacks its child nodes (elements) on top of each other. It is particularly useful when you want to overlay elements, ensuring that the last added child appears on top of the stack. Think of it as a "stacked deck" of UI components.

Key Features of StackPane:

1.Child Positioning:

By default, all child nodes are centered within the.

You can change the alignment of individual children using.

2. Stacking Behavior:

Nodes are drawn in the order they are added. The first added child will be at the bottom, and subsequent children will overlay on top.

3. Resizing:

StackPane resizes its children to fit within its bounds. It ensures all nodes adjust to the available space, depending on their preferences and constraints.

Example Code:

Here’s a simple example to demonstrate the usage of :

**package** application;

**import** javafx.application.Application;

**import** javafx.scene.Scene;

**import** javafx.scene.control.Label;

**import** javafx.scene.layout.StackPane;

**import** javafx.scene.paint.Color;

**import** javafx.scene.shape.Rectangle;

**import** javafx.stage.Stage;

**public** **class** **Main** **extends** Application {

**@Override**

**public** **void** **start**(**Stage** primaryStage) {

// Create a StackPane

**StackPane** **stackPane** = **new** **StackPane**();

// Add nodes to the StackPane

**Rectangle** **rectangle** = **new** **Rectangle**(**200**, **100**);

rectangle.**setFill**(**Color**.**LIGHTBLUE**);

**Label** **label** = **new** **Label**(**"Hello, StackPane!"**);

stackPane.**getChildren**().**addAll**(rectangle, label); // Rectangle at bottom, Label on top

// Create the Scene

**Scene** **scene** = **new** **Scene**(stackPane, **300**, **200**);

primaryStage.**setTitle**(**"StackPane Example"**);

primaryStage.**setScene**(scene);

primaryStage.**show**();

}

**public** **static** **void** **main**(**String**[] args) {

***launch***(args);

}

}

A screenshot of a computer screen

AI-generated content may be incorrect.

**\*Explanation:**

Nodes Added: A rectangle and a label are added to the . The rectangle is rendered first, forming the background, while the label is displayed on top.

Center Alignment: Both nodes are aligned to the center by default.

**\*When to Use StackPane:**

To overlay multiple UI elements (e.g., text on a background shape).

To create backgrounds behind other components.

In cases where child nodes need to share the same space.

Ex2:

**package** application;

**import** javafx.application.Application;

**import** javafx.scene.Scene;

**import** javafx.scene.control.Label;

**import** javafx.scene.layout.StackPane;

**import** javafx.scene.paint.Color;

**import** javafx.scene.shape.Rectangle;

**import** javafx.stage.Stage;

**public** **class** **Main** **extends** Application {

**@Override**

**public** **void** **start**(**Stage** primaryStage) {

// Create nodes

**Rectangle** **rectangle** = **new** **Rectangle**(**200**, **100**);

rectangle.**setFill**(**Color**.**LIGHTBLUE**);

**Label** **label** = **new** **Label**(**"Hello, StackPane!"**);

**Rectangle** **rectangle2** = **new** **Rectangle**(**200**, **100**);

rectangle2.**setFill**(**Color**.**GREEN**);

// Create a StackPane

**StackPane** **stackPane** = **new** **StackPane**(rectangle,label,rectangle2);

// rectangle at bottom, label on top of rectangle, rectangle2 on top of label

// Create the Scene

**Scene** **scene** = **new** **Scene**(stackPane, **300**, **200**);

primaryStage.**setTitle**(**"StackPane Example"**);

primaryStage.**setScene**(scene);

primaryStage.**show**();

}

**public** **static** **void** **main**(**String**[] args) {

***launch***(args);

}

}

A screenshot of a computer

AI-generated content may be incorrect.

**According to this example, if you want to see rectangle, label, or rectangle2 tool, try to bring it to the front or your screen with syntax like this:**

rectangle.toFront();

Ex:

**package** application;

**import** javafx.application.Application;

**import** javafx.scene.Scene;

**import** javafx.scene.control.Label;

**import** javafx.scene.layout.StackPane;

**import** javafx.scene.paint.Color;

**import** javafx.scene.shape.Rectangle;

**import** javafx.stage.Stage;

**public** **class** **Main** **extends** Application {

**@Override**

**public** **void** **start**(**Stage** primaryStage) {

// Create nodes

**Rectangle** **rectangle** = **new** **Rectangle**(**200**, **100**);

rectangle.**setFill**(**Color**.**LIGHTBLUE**);

**Label** **label** = **new** **Label**(**"Hello, StackPane!"**);

**Rectangle** **rectangle2** = **new** **Rectangle**(**200**, **100**);

rectangle2.**setFill**(**Color**.**GREEN**);

// Create a StackPane

**StackPane** **stackPane** = **new** **StackPane**(rectangle,label,rectangle2);

// rectangle at bottom, label on top of rectangle, rectangle2 on top of label

rectangle.**toFront**();

// Create the Scene

**Scene** **scene** = **new** **Scene**(stackPane, **300**, **200**);

primaryStage.**setTitle**(**"StackPane Example"**);

primaryStage.**setScene**(scene);

primaryStage.**show**();

}

**public** **static** **void** **main**(**String**[] args) {

***launch***(args);

}

}

A screenshot of a computer

AI-generated content may be incorrect.

To bring a node to the back, write something like

nodeName.toBack();

ex:

**package** application;

**import** javafx.application.Application;

**import** javafx.scene.Scene;

**import** javafx.scene.control.Label;

**import** javafx.scene.layout.StackPane;

**import** javafx.scene.paint.Color;

**import** javafx.scene.shape.Rectangle;

**import** javafx.stage.Stage;

**public** **class** **Main** **extends** Application {

**@Override**

**public** **void** **start**(**Stage** primaryStage) {

// Create nodes

**Rectangle** **rectangle** = **new** **Rectangle**(**200**, **100**);

rectangle.**setFill**(**Color**.**LIGHTBLUE**);

**Label** **label** = **new** **Label**(**"Hello, StackPane!"**);

**Rectangle** **rectangle2** = **new** **Rectangle**(**200**, **100**);

rectangle2.**setFill**(**Color**.**GREEN**);

// Create a StackPane

**StackPane** **stackPane** = **new** **StackPane**(rectangle,label,rectangle2);

// rectangle at bottom, label on top of rectangle, rectangle2 on top of label

rectangle2.**toBack**();

label.**toFront**();

// Create the Scene

**Scene** **scene** = **new** **Scene**(stackPane, **300**, **200**);

primaryStage.**setTitle**(**"StackPane Example"**);

primaryStage.**setScene**(scene);

primaryStage.**show**();

}

**public** **static** **void** **main**(**String**[] args) {

***launch***(args);

}

}

A screenshot of a computer

AI-generated content may be incorrect.

If I remove the line rectangle2.toBack() on the example above =>

**package** application;

**import** javafx.application.Application;

**import** javafx.scene.Scene;

**import** javafx.scene.control.Label;

**import** javafx.scene.layout.StackPane;

**import** javafx.scene.paint.Color;

**import** javafx.scene.shape.Rectangle;

**import** javafx.stage.Stage;

**public** **class** **Main** **extends** Application {

**@Override**

**public** **void** **start**(**Stage** primaryStage) {

// Create nodes

**Rectangle** **rectangle** = **new** **Rectangle**(**200**, **100**);

rectangle.**setFill**(**Color**.**LIGHTBLUE**);

**Label** **label** = **new** **Label**(**"Hello, StackPane!"**);

**Rectangle** **rectangle2** = **new** **Rectangle**(**200**, **100**);

rectangle2.**setFill**(**Color**.**GREEN**);

// Create a StackPane

**StackPane** **stackPane** = **new** **StackPane**(rectangle,label,rectangle2);

// rectangle at bottom, label on top of rectangle, rectangle2 on top of label

label.**toFront**();

// Create the Scene

**Scene** **scene** = **new** **Scene**(stackPane, **300**, **200**);

primaryStage.**setTitle**(**"StackPane Example"**);

primaryStage.**setScene**(scene);

primaryStage.**show**();

}

**public** **static** **void** **main**(**String**[] args) {

***launch***(args);

}

}

A screenshot of a computer

AI-generated content may be incorrect.