

Interfaces de usuario gráficas (JavaFX) en Kotlin

Introducción a JavaFX

JavaFX es un conjunto de bibliotecas y herramientas gráficas enriquecidas para crear aplicaciones de usuario interactivas en Java. En Kotlin, también podemos aprovechar JavaFX para diseñar interfaces de usuario modernas y atractivas.

Conceptos clave

Antes de sumergirnos en ejemplos prácticos, es importante comprender algunos conceptos clave relacionados con JavaFX en Kotlin.

Escena

En JavaFX, una escena es un contenedor de nivel superior que contiene todos los elementos de la interfaz de usuario. Las aplicaciones JavaFX pueden tener múltiples escenas que representan diferentes pantallas o vistas.

Controles

Los controles son elementos de la interfaz de usuario que permiten a los usuarios interactuar con la aplicación. JavaFX proporciona una variedad de controles, como botones, etiquetas, campos de texto y más.

Eventos

Los eventos son interacciones del usuario con la GUI, como hacer clic en un botón o ingresar datos en un campo de texto. JavaFX maneja eventos y permite que los controles respondan a

ellos mediante controladores de eventos.

Ejemplos

A continuación, presentaremos ejemplos de cómo crear controles y diseñar una GUI básica en Kotlin utilizando JavaFX.

Ejemplo 1: Crear una ventana

[crayon-6a9d580576199347591551/]

Ejemplo 2: Agregar un botón

[crayon-6a9d5805761a0183347112/]

Estos ejemplos demuestran cómo crear una ventana y agregar un botón a la interfaz de usuario utilizando JavaFX en Kotlin. JavaFX ofrece una amplia gama de controles y funcionalidades para diseñar interfaces de usuario modernas y dinámicas.

Componentes

Scene: Representa el contenedor de nivel superior que contiene todos los elementos de la interfaz de usuario

[crayon-6a9d5805761a2757691599/]

Stage: Es la ventana principal de la aplicación JavaFX. Puede contener una o varias escenas

[crayon-6a9d5805761a5264572043/]

Button: Un botón que los usuarios pueden hacer clic

[crayon-6a9d5805761a6554578287/]

Label: Etiqueta de texto utilizada para mostrar información

[crayon-6a9d5805761a8155330486/]

TextField: Un campo de texto que permite a los usuarios ingresar texto

TextArea: Un área de texto de varias líneas para ingresar texto extenso

CheckBox: Una casilla de verificación que permite a los usuarios seleccionar una o varias opciones

RadioButton: Un botón de radio que permite a los usuarios seleccionar una opción de un conjunto

ToggleButton: Un botón que puede alternar entre dos estados

ChoiceBox: Una lista desplegable que permite a los usuarios seleccionar una opción de una lista

ComboBox: Similar a ChoiceBox, pero permite a los usuarios escribir o seleccionar opciones de una lista desplegable

ListView: Muestra una lista de elementos que los usuarios pueden ver y seleccionar

TableView: Una tabla que muestra datos en filas y columnas

MenuBar: Un menú en la parte superior de la ventana que contiene elementos de menú

Menu: Un elemento de menú dentro de un MenuBar que puede contener submenús y opciones

ContextMenu: Un menú contextual que aparece cuando se hace clic derecho en un elemento

ToolBar: Una barra de herramientas que contiene botones u otros controles

Slider: Un control deslizante que permite a los usuarios seleccionar un valor en un rango

ProgressBar: Muestra el progreso de una tarea en forma de barra

ProgressIndicator: Muestra una animación de progreso circular

Accordion: Un contenedor que muestra una serie de paneles colapsables

TabPane: Un contenedor de pestañas que muestra múltiples vistas en pestañas

WebView: Muestra contenido web en una aplicación JavaFX

Canvas: Permite dibujar gráficos personalizados

ImageView: Muestra imágenes

ScrollPane: Proporciona barras de desplazamiento para contenido desbordante

Dialog: Muestra ventanas emergentes modales para interactuar con el usuario

FileChooser: Permite a los usuarios seleccionar archivos y directorios

ColorPicker: Permite a los usuarios seleccionar colores

DatePicker: Permite a los usuarios seleccionar fechas

TreeView: Muestra datos en forma de árbol