

Tokaj-Hegyalja
Egyetem

Komputer grafika és képszerkesztés

JavaFX bemutató

Mi is az a JavaFX?

- A JavaFX egy modern grafikus felhasználói felület készítésére szolgáló keretrendszer a Java platformon.
- Segítségével asztali és mobil alkalmazásokat készíthetünk
 - ezek grafikus elemeket – például gombokat, szövegmezőket, menüket vagy akár grafikonokat – tartalmaznak.
- **Régebben:** a Java **Swing** technológiát használták GUI alkalmazások készítésére
 - a JavaFX modernebb megközelítést kínál
- **Támogatja:**
 - a korszerű felhasználói felületeket,
 - animációkat, médiakezelést,
 - valamint CSS-alapú stílusozást is,
 - lehetővé teszi, hogy az alkalmazások megjelenése könnyen testre szabható legyen.

Mi is az a JavaFX?

- A JavaFX alkalmazás alapvető felépítése néhány fontos fogalom köré épül:
 - A legfelső szintű ablakot **Stage**-nek nevezzük, ez maga az alkalmazás ablaka.
 - Ebben helyezkedik el a **Scene**, amely a felhasználói felület teljes tartalmát tartalmazza.
 - A Scene-ben különböző grafikus elemeket – például gombokat, címkéket vagy beviteli mezőket – helyezhetünk el.
 - Ezeket az elemeket **node**-oknak nevezzük.
- A JavaFX eseményvezérelt módon működik:
 - a program reagál a felhasználó műveleteire,
 - például egy gomb megnyomására vagy egy mezőbe írt szövegre.
 - A fejlesztő feladata, hogy megadja, milyen művelet történjen egy adott esemény bekövetkezésekor.

Mi is az a JavaFX?

- A JavaFX egyik előnye, hogy a felhasználói felületet kétféleképpen is létrehozhatjuk:
 - 1) teljesen Java kódból,
 - 2) egy úgynevezett FXML fájl segítségével,
 - egy deklaratív leírása a felületnek.
 - Az FXML fájlok létrehozását és szerkesztését vizuális eszközök is segíthetik
 - **Pl. Scenebuilder**
 - <https://gluonhq.com/products/scene-builder/>

<https://openjfx.io>

Mi is az a JavaFX?

Példa hierarchia:

Stage

└ Scene

└ VBox

├ Label

├ TextField

└ Button

JavaFX Hello World...

Hello World

```
public void start(Stage primaryStage) {  
  
    primaryStage.setTitle("Hello World!");  
  
    Button btn = new Button();  
    btn.setText("Hello World!");  
    btn.setOnAction(new EventHandler<ActionEvent>() {  
  
        @Override  
        public void handle(ActionEvent event) {  
            System.out.println(" Hello World!");  
        }  
    });  
  
    StackPane root = new StackPane();  
    root.getChildren().add(btn);  
    primaryStage.setScene(new Scene(root, 300, 250));  
    primaryStage.show();  
}
```

Hello World - JavaFX



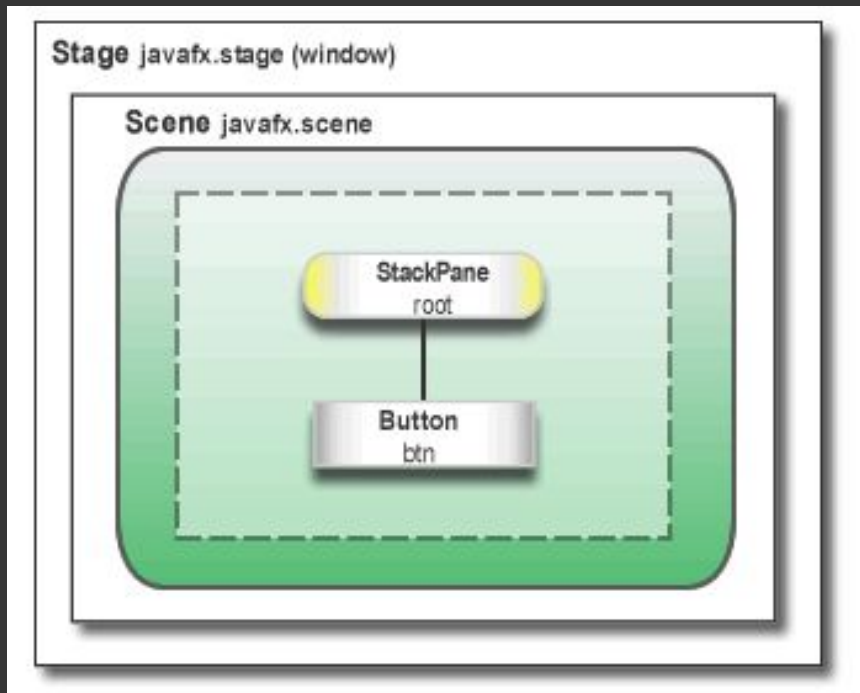
Hello World

- The main class for a JavaFX application extends the `javafx.application.Application` class.
 - The `start()` method is the main entry point for all JavaFX applications.
- A JavaFX application defines the user interface container by means of a `stage` and a `scene`.
- The JavaFX Stage class is the top-level JavaFX container. The JavaFX Scene class is the container for all content.
- In JavaFX, the content of the scene is represented as a hierarchical scene graph of nodes.
- In HelloWorld example, the root node is a StackPane object, which is a resizable layout node.
 - This means that the root node's size tracks the scene's size and changes when the stage is resized by a user.

Hello World

- The root node contains one child node:
 - a button control with text,
 - plus an event handler to print a message when the button is pressed
- The main() method is not required for JavaFX applications when the JAR file for the application is created with the JavaFX Packager tool,
 - this embeds the JavaFX Launcher in the JAR file.
- However, it is useful to include the main() method so we can run JAR files that were created without the JavaFX Launcher
 - when using an IDE in which the JavaFX tools are not fully integrated.
 - Also, Swing applications that embed JavaFX code require the main() method.

Hello World Scene Graph



Köszönöm a figyelmet!