# Bevezetés

## GUI (Graphical User Interface)

A korszerű programokat többnyire nem parancssorból, hanem grafikus felületen használjuk.

A Java-ban az idők során többféle megoldás is volt grafikus felület készítésére:

* Az első megoldás az AWT (Abstract Window Toolkit) volt. Ez már elavult.
* Erre épült a Swing, amely még használható, de már nem fejlesztik tovább.
* A legkorszerűbb megoldás a JavaFX, amelyet a Swing leváltására fejlesztettek ki. Ezt fogjuk tanulni.   
  A JavaFX a 8-as változattól kezdve a 10-esig a Java része, az újabb változatokhoz külön kell letölteni a <https://openjfx.io/> címről.

## JavaFX alkalmazások

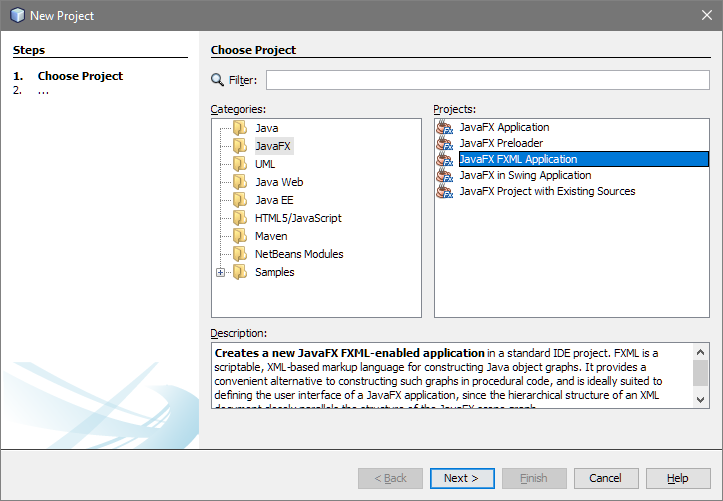
JavaFX alkalmazást kétféleképpen lehet készíteni:

* Az első megoldás a tiszta Java kód használata. Ekkor a felületet és a hozzá tartozó logikát is egy osztályban adjuk meg.
* A második esetben a grafikus felület leírása külön, egy FXML formátumú fájlba kerül. Így a program feladatai jobban elkülönülnek egymástól.

Mi a második lehetőséget választjuk, és FXML alkalmazásokat fogunk készíteni.

## Minta alkalmazás

Indítsd el a NetBeans-t, és kezdj egy új JavaFX FXML projektet *proba* néven!



Futtasd le és próbáld ki a kapott programot! Mit csinál a program?

## Az ablakok felépítése

Az alkalmazás ablakait a JavaFX-ben **Stage**-nek (színpadnak) nevezik.

Az ablakban egyszerre megjelenő elemek egy **Scene**-be (jelenetbe) kerülnek.

A jelenetbe kerülő elemek a **Node**-ok (csomópontok). Ezek lehetnek vezérlők (control, például egy gomb) vagy tárolók (container, például egy AnchorPane), amelyek újabb node-okat tartalmazhatnak.

Egy root node-nak kell lennie, és azon lehetnek újabb node-ok. Példánkban egy AnchorPane tároló a root node, és azon van egy Button és egy Label vezérlő.

A root node kitölti a rendelkezésre álló helyet, ezért általában tároló szokott lenni.

Stage (ablak)

Scene (jelenet)

AnchorPane (root node)

Button (node)

Label (node)

## Az FXML alkalmazás elemei

Nézd meg a kapott három (!) fájlt!

Az ***FXMLdocument.fxml*** fájl a jelenet leírását tartalmazza XML formátumban. Hasonlít a weblapok HTML kódjához, de az XML-ben minden elemet le kell zárni.

A NetBeans-ben a jobb kattintás, Edit paranccsal lehet megnyitni.

Az XML változat megadása és a szükséges importálások után a 7-12. sorban van a felület definiálása.

Az XML előnye, hogy az elemek szerkezete könnyen áttekinthető. Itt a root elem az AnchorPane, amelynek két „gyermeke” van: egy Button (gomb) és egy Label (címke).

Egyszerűbb módosításokat itt, az XML fájlban is el lehet végezni:

* Módosítsd az AnchorPane szélességét (prefWidth) 400-ra, és futtasd a programot!
* Változtasd a Button és a Label x koordinátáját (layoutX) 166-ra, és próbáld ki ezt is!

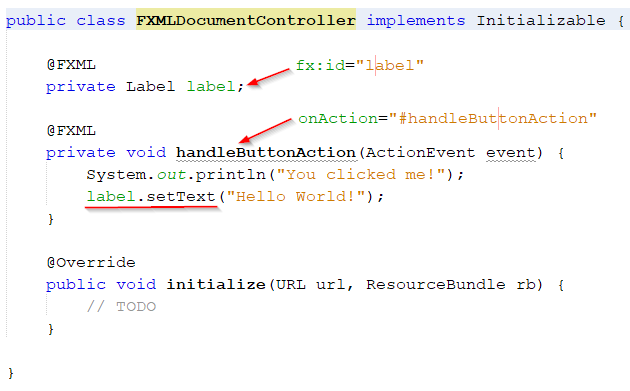
Figyeld meg, hogy minden vezérlőnek van egy fx:id azonosítója!

A grafikus felületen a programok futását az események (events) irányítják. Ezek közül leggyakoribb, amikor a felhasználó rákattint valamire, például egy gombra. Ez egy ActionEvent.

Figyeld meg, hogy a Button elemnek van egy onAction argumentuma, amelyben meg van adva egy metódus neve (handleButtonAction)! Amikor megnyomják a gombot, akkor ez a metódus fog lefutni. Ezt a metódust nevezik a gomb eseménykezelőjének.

Az eseménykezelők kódjait az ***FXMLDocumentController*** osztály tartalmazza.

Az @FXML annotáció jelzi azokat a részeket, amelyekben az FXML fájlban megadott azonosítót vagy metódust definiáljuk.



Figyeld meg, hogyan lehet egy címke szövegét beállítani az eseménykezelőben (aláhúzva)!

Az initialize metódusba írt kódot a program indulásakor hajtaná végre, de erre most nincs szükség.

Feladat: Írd át az üzeneteket az eseménykezelőben, és próbáld ki!

Nyisd meg végül a **Proba.java** fájlt, amely elindítja a programot!

Figyeld meg, hogy mindent a javafx csomagból importáltunk!

A program osztálya mindig az Application osztály leszármazottja lesz.

Az Application osztály main metódusát a program csak a parancssori paraméterek átadására használja. Mi sem fogunk írni bele semmi mást.

A main metódusból elindított launch metódus lefuttatja a start metódust. Ezt felül kell írnunk az ablak megjelenítéséhez szükséges utasításokkal.

A start metódus először betölti az FXML fájlt tartalmát egy Parent típusú root elembe. Ezután készít egy új jelenetet belőle (Scene), a jelenetet hozzárendeli egy ablakhoz (Stage), majd megjeleníti az ablakot a show metódus segítségével.

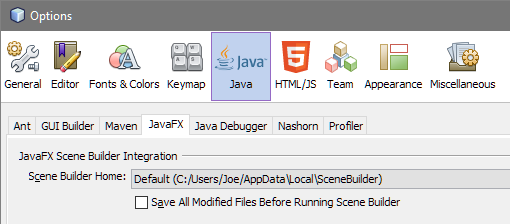
## Scene Builder

Az FXML fájl kézzel történő módosítása elég nehézkes. Szerencsére azonban a Scene Builder alkalmazás segítségével grafikusan szerkeszthető a felület, és automatikusan elkészíthető az FXML fájl.

A Scene Buildert az Oracle nyílt forráskódúvá tette, és a Gluon támogatásával fejlesztik tovább.

A Scene Builder nem része sem a JDK-nak sem a NetBeans-nek. Külön kell letölteni a <https://gluonhq.com/products/scene-builder/> címről. Ügyelj arra, hogy a JDK változatának megfelelő verziót telepítsd (8-as vagy 11-es)!

Telepítéskor a NetBeans-ben beállítja a program helyét. Szükség esetén a Tools, Options, Java, JavaFX lapon lehet ellenőrizni vagy beállítani:



Telepítés után az FXML fájlokra duplán kattintva lehet elindítani a Scene Buildert. Próbáld ki!

Megjegyzés: Az FXML fájlok megnyitása a Scene Builderben csak akkor működik, ha az elérési útban nincs ő vagy ű karakter!

## Kérdések

1. Milyen elemekből áll egy JavaFX alkalmazás ablaka?
2. Milyen fájlokat hozunk létre egy egyszerű FXML alkalmazáshoz?
3. Mit nevezünk eseménykezelőnek?