

## 1. A környezet beállítása

- Töltsük le a SceneBuilder telepítőt az alábbi linkről:  
<http://gluonhq.com/products/scene-builder/>  
Sajnos az Oracle verziójának telepítője nem érhető el futtatható formában, ezért használjuk ezt.
- Telepítsük a SceneBuildert. (Jogosultsági problémák esetén a .jar verzió is használható, ez viszont biztosan nem kapcsolható a NetBeans környezetbe). Ne felejtjük el, hova telepítettük a SceneBuildert.
- Indítsuk el a NetBeanst de ne készítsünk új projektet. Navigáljunk a Tools->Options->Java-JavaFx menüpont alá. Állítsuk be a Scene Builder Home értéket az előbb megjegyzett telepítési helyre. (Ok).

## 2. Indítsunk egy új JavaFX projektet

- File->New Project...->JavaFX->**JavaFX Application**
- Legyen a projekt neve StudentsRegister. A Main Application class nevét nem szükséges megváltoztatni.
- Tekintsük át a projekt csomagszerkezetét és a generált fájlokat.

## 3. Szervezzük újra a projektünket

- Készítsünk egy alcsomagot a studentsregister csomag alatt model néven.
- Készítsünk egy másik alcsomagot view néven.
- Ajunk egy új üres FXML fájlt a view csomaghoz. (Jobbklikk a csomag nevére, majd -> New...->Other->JavaFX->Empty FXML). Nevezzük el FXMLStudentsScene-nek. Jelöljük ki a "Use Java Controller" checkbox-ot és használjuk az alapértelmezett controller osztály nevet. Ne használjunk formátumfájlt.
- Írjuk át a StudentsRegister#start metódust az alábbiak szerint:
 

```
FXMLLoader loader = new
      FXMLLoader(StudentsRegister.class.getResource("view/FXMLStudentsScene.fxml"));
      Scene scene = new Scene(loader.load());
      primaryStage.setTitle("Students Register");
      primaryStage.setScene(scene);
      primaryStage.show();
```
- Dupla klikk (vagy jobbklikk->Open) a FXMLStudentsScene.fxml fájlra (ez megnyitja a SceneBuildert. A Scene Builder-ben húzzunk egy gombot az ablak közepére. Futtassuk az alkalmazást...

## 4. Adjunk funkciót a gombnak

- Nyissuk meg az FXMLStudentsSceneController.java fájlt és adjunk egy új handleButtonPushed() nevű metódust hozzá. Ez a metódus írja a képernyőre, hogy "Button pushed".
- Használjuk az @FXML annotációt a metóduson (így tudjuk a gombhoz kötni a Scene Builder-ben)
- Térjünk vissza a Scene Builder-be. Válasszuk ki a gombot és a jobboldali panelen válasszuk a "Code : Button" lehetőséget. Állítsuk be az "On action" mezőt "handleButtonPushed"-ra. Mentjük el a projektet és futtassuk..

## 5. Kössük a gomb működését egy címkéhez

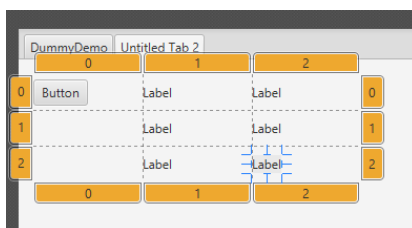
- A SceneBuilder-ben adjunk egy új címkét a Scene-hez.
- Nyissuk meg az FXMLStudentsSceneController.java fájlt és adjunk egy „Label” típusú mezőt a kódhoz "helloLabel" néven. Állítsuk privátra és használjuk rajta az @FXML annotációt.
- Adjuk az alábbi sort a handleButtonPushed metódushoz:
 

```
helloLabel.setText("Button pushed");
```

- d. Menjünk át a Scene Builder-be és válasszuk ki a címkét. Állítsuk be az fx:id tulajdonságot (Code : Label a jobboldalon) helloLabel-re.

## 6. Töltsünk be adatokat a modellből

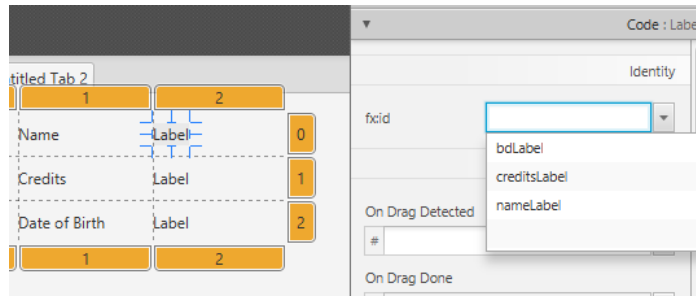
- a. A Scene Builder-ben szervezzük át a Scene-ünket. Adjunk egy TabPane-t a gyökér AnchorPane-hez és mozgassuk a korábban elkészült Button és Label elemünket az első Tab-ra. Legyen ennek a Tab-nak a neve DummyDemo
- b. Nevezzük át a másik Tabot LoadDemo-ra.
- c. Jobbklikk a TabPane-en a jobboldali menüben és válasszuk a Fit to Parent lehetőséget, hogy a TabPane kitöltse az ablakot.
- d. Adjunk egy GridPane-t a LoadDemoTab-hoz és töltsük ki az alábbiak szerint (új oszlop hozzáadásához rá kell kattintani a táblázat fejlécére jobbklikkel):



- e. A gomb szövege legyen "Load" Az első oszlopban lévő címkék szövegei pedig rendre fentről lefelé "Name", "Credits", és "Date of birth".
- f. Adj további három Label típusú mezőt az FXMLElementsSceneController-hez "nameLabel", "creditsLabel" és "bdLabel" nevekkel. (Használd rajtuk az @FXML annotációt)
- g. A model csomagban készíts egy osztályt, ami egy hallgatót reprezentál Student néven. Három tulajdonságot adj meg: name, credits, és date\_of\_birth. Generálj getter és setter metódusokat, valamint egy konstruktort az osztálynak..
- h. Adj egy Model nevű osztályt a model csomaghoz. Legyen ennek egy Student típusú mezője. Generálj hozzá gettert és setttert. Készíts alapértelmezett konstruktort, ami a new Student értékét az alábbiak szerint állítja be: ("Robert Smith", 55, "1991-12-03")
- i. Adj egy mezőt a StudentsSceneController-hez model néven (Model típusú)és adj hozzá egy setttert.
- j. A StudentsRegister#start metódushoz add hozzá az alábbi sort:  

```
((FXMLStudentsSceneController)loader.getController()).setModel(new Model());
```

a Scene inicializációja után.
- k. Most, hogy a Model és a View össze van kötve, adj egy handleLoad metódust a StudentsSceneController osztályhoz, ami beállítja a megfelelő változókat a Student-ből olvasott értékekkel.
- l. Ne felejtjük beállítani az fx:id tulajdonságokat a GridPane harmadik oszlopában szereplő címkéken.



- m. A Load gomb "On Action" tulajdonságát kössük hozzá a "handleLoad" metódushoz.
- n. Mentsünk mindent és futtassuk az alkalmazást...

## 7. Mi van, ha változik az adat?

- a. Adjunk a Tabhoz egy új Gombot "Change Name" szöveggel.
- b. Adjunk egy kezelő metódust a StudentsSceneController osztályhoz handleChangeName néven. Ez a gomb állítsa át a hallgató nevét "Tom Smith"-re.
- c. Kössük össze a gombot és a metódust. (Ha jól csináltad, a név nem változik 😊.)
- d. Használjunk a Student osztályban StringProperty mezőt a név esetén az egyszerű String helyett. (Tipp: ALT+INSERT segítségével JavaFX Property-t tudunk az osztályhoz adni). Javítsuk a hibákat...
- e. A StudentsSceneController# handleLoad metódusban írjuk át a név beállító sort az alábbiak szerint:
 

```
nameLabel.textProperty().bind(model.getStudent().nameProperty());
```
- f. Futtassuk az alkalmazást. (Ha most megnyomjuk a "Change name" gombot, a név automatikusan frissül.)

## 8. Dialógusok

- a. Adjuk az alábbi sorokat a handleChangeName metódushoz:
 

```
Alert alert = new Alert(AlertType.INFORMATION);
alert.setTitle("Button pushed");
alert.setHeaderText("You have pushed the Change Name button");
alert.setContentText("This shows that you have pushed a button...");
alert.showAndWait();
```
- b. A dialógusokról bővebben: <http://code.makery.ch/blog/javafx-dialogs-official/>

Egy sokkal részletesebb JavaFX tutorial: <http://code.makery.ch/library/javafx-8-tutorial/>

Serializáció: [https://www.tutorialspoint.com/java/java\\_serialization.htm](https://www.tutorialspoint.com/java/java_serialization.htm)