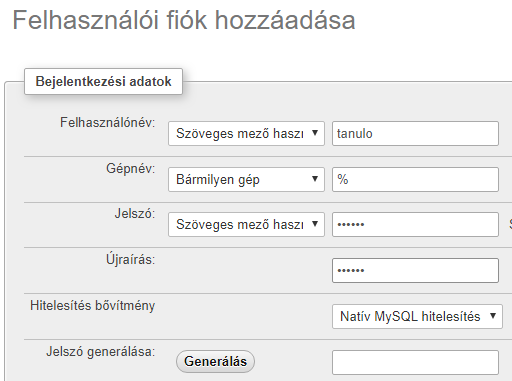
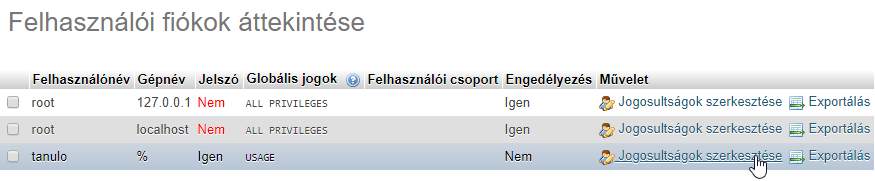
# Gépnapló

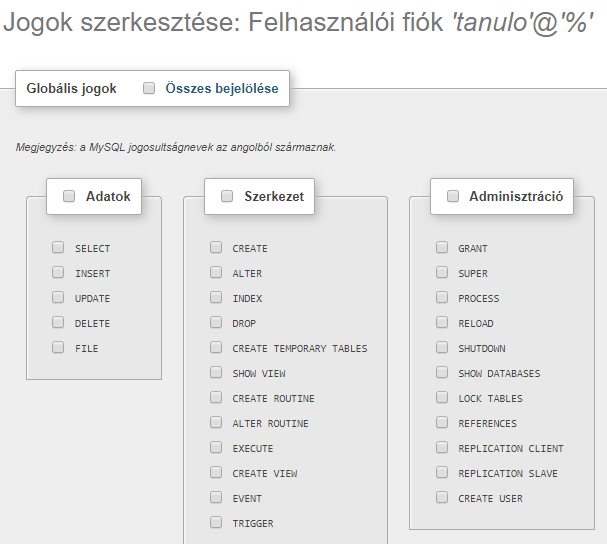
## Előkészületek

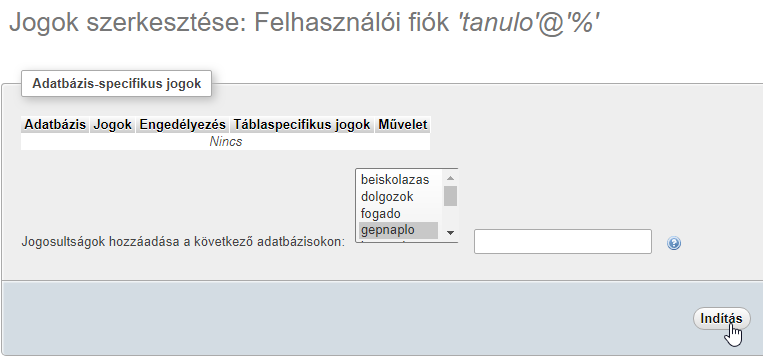
1. Indítsd el a Wamp servert!
2. Készíts egy új, *gepnaplo* nevű adatbázist!
3. Importáld az adatbázisba a *gepek.sql* fájlt!
4. Nézd meg a *gepek* tábla szerkezetét!
5. Hozz létre egy új, *tanulo* nevű felhasználót, aki bármely gépről bejelentkezhet a *tanulo* jelszóval!

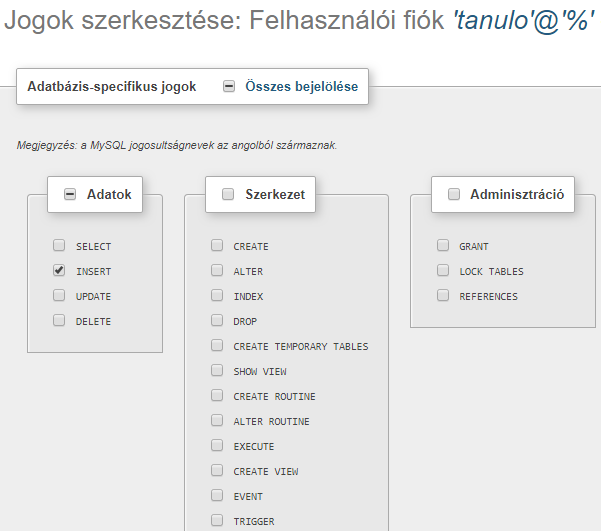


Állítsd be a tanulo felhasználó jogait úgy, hogy csak a *gepnaplo* adatbázisba tudjon adatot bevinni (semmi mást)!







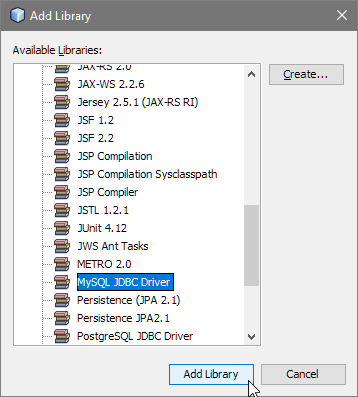


## Új projekt

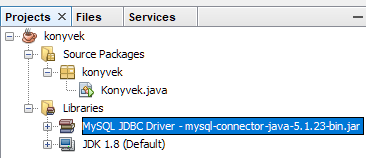
Kezdj egy új Java FXML projektet a NetBeans-ben *gepnaplo* néven!

Kattins a jobb gombbal a Projects ablakban a Libraries elemre, majd válaszd az Add Library parancsot!

Válaszd ki a MySQL JDBC Driver-t, majd kattints az Add Library gombra!



Ezzel hozzáadtad a drivert a projekthez:



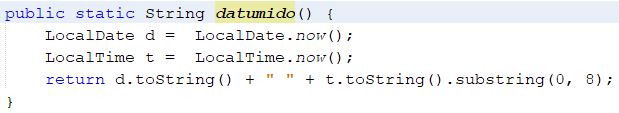
## Dátum és idő lekérése

Készíts egy új, Seged nevű osztályt a gepnaplo csomagba!

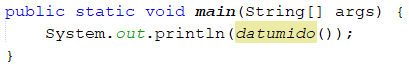
A bejelentkezés dátumát és idejét a programnak kell meghatároznia.

Ehhez a LocalDate és a LocalTime osztályok now metódusait fogjuk használni.

Készíts a Seged osztályban egy statikus datumido metódust:



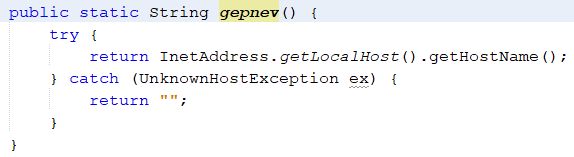
Készíts egy main metódust (psvm, Tab), és hívd meg belőle a datumido metódust:



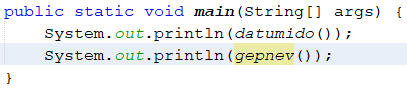
Próbáld ki (Shift+F6)!

## A számítógép nevének lekérdezése

A számítógépnevet is a programnak kel meghatároznia. Erre is készíts egy metódust:



Próbaképpen hívd meg a main metódusból:



Próbáld ki!

## Stringek méretre vágása

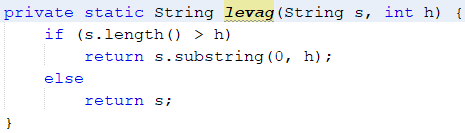
Amikor szöveges adatot viszünk be az adattáblába, nem adhatunk meg hosszabb szöveget, mint ami a mezőben elfér.

A szöveges mezőknek viszont nem lehet megadni beviteli korlátot. Ezért a beolvasás után fogjuk méretre vágni a szövegeket.

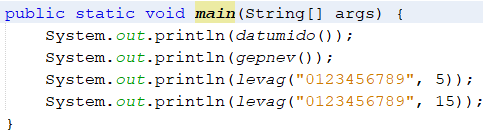
Kézenfekvő megoldás a substring metódus használata, de van vele egy kis probléma: Ha olyan határt adunk meg, amely nagyobb, mint a string tényleges hossza, kivétel keletkezik.

Ezért készítünk egy metódust, amely csak akkor vágja le a string végét, ha az hosszabb, mint a megadott határ. Ezt majd az adatbevitelnél fogjuk használni.

Készítsd el a levag metódust a Seged osztályba:



Egészítsd ki a main metódust a levag metódus tesztelésével:



Futtasd a fájlt!

## Adatok beírása az adatbázisba

Készíts egy DB nevű osztályt!

A tanulók bejelentkezésének adatait a beir metódussal fogjuk bevinni az adatbázisba. A metódus paraméterként megkapja a szükséges adatokat.

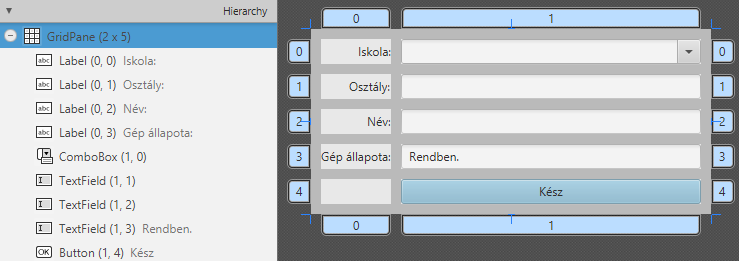
A metódusban kapcsolódunk az adatbázishoz, létrehozunk egy előkészített parancsot, majd lefuttatjuk.

Írd be a következőket:



## GUI létrehozása

Nyisd meg az FXML fájlt a Scene Builderben, majd készítsd el a grafikus felületet az alábbi minta alapján:



A GridPane, valamint sorainak és oszlopainak szélessége és magassága számított legyen, körben 10 pontos paddinggel, és mindkét irányban 10 pontos térközökkel!

A címkék preferált szélessége 70 pont, és jobbra vannak igazítva (CENTER\_RIGHT).

A ComboBox legyen szerkeszthető, 300 pont széles, azonosítója pedig cbxIskola!

A szövegmezők is legyenek 300 képpont szélesek, azonosítóik pedig: txtOsztaly, txtNev, txtAllapot!

A gomb szintén 300 pont széles, default gomb, és megnyomásakor a kesz metódus fusson le!

Állíts be stílus segítségével narancssárga (orange) hátteret a GridPane elemre!

Add meg a controller osztály nevét a bal alsó sarokban (Document, Controller)!

Ezután nyisd meg a View, Show Sample Controller Skeleton paranccsal a controllerbe másolandó kódot, és másold a vágólapra a szükséges deklarációkat!

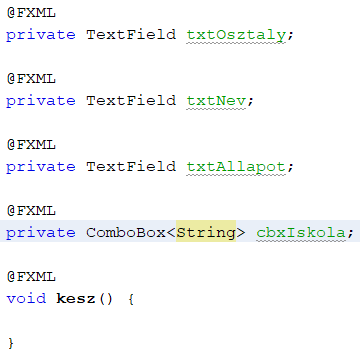
Mentsd a jelenetet, és válts vissza a NetBeans-re!

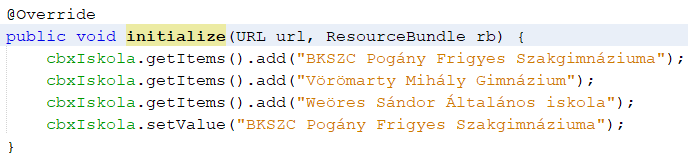
Nézd át a generált FXML dokumentumot!

## Controller osztály

Másold be a kódot a vágólapról a controller osztályba! Végezd el a szükséges importálásokat!

Add meg a típust, és töröld az eseményt a szükséges helyeken:

  
Állítsd be a ComboBox értékeit:



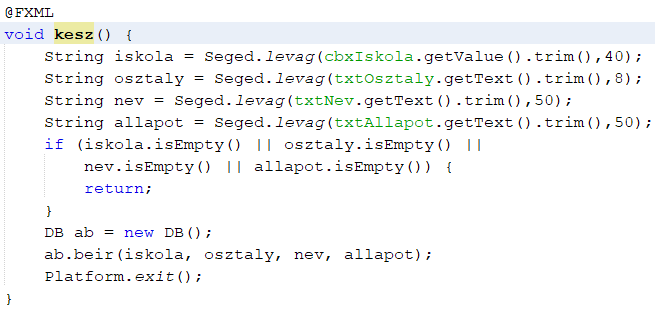
Futtasd a programot! Figyeld meg, hogy a ComboBox-ba be lehet írni a listában nem szereplő szöveget is!

## Kesz metódus

A kesz metódus a Kész gomb megnyomásakor fut le. Először végez egy ellenőrzést, és ha valamelyik mező üres, akkor kilép a metódusból.

Ha ki vannak töltve az adatok, akkor meghívja az adatbázis beir metódusát, és átadja neki az adatokat. Végül kilép a programból.

Írd be a kesz metódust:



Figyeld meg, hogy a beolvasott szövegek elejéről és végéről töröljük a szóközöket a trim metódussal, majd méretre vágjuk őket!

Próbáld ki adatok bevitele nélkül, és adatokkal is!

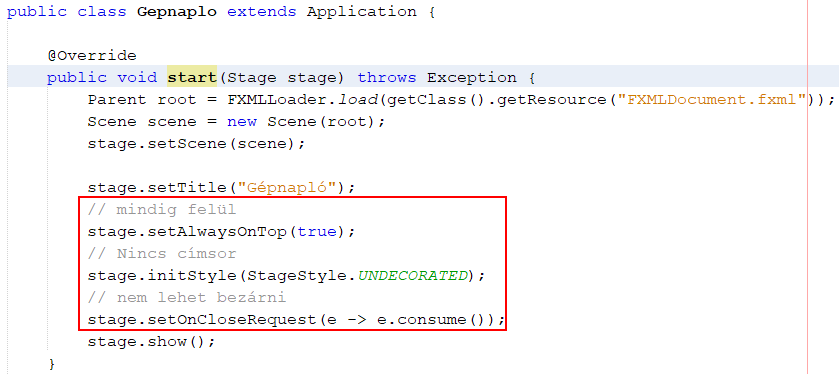
Ellenőrizd az adatbevitelt az adatbázisban is!

## Ablak kezelése

Azt szeretnénk, hogy:

* az ablak mindig felül legyen,
* ne legyen címsora,
* ne lehessen bezárni az adatok bevitele nélkül.

Ezeket a következő módon lehet beállítani:

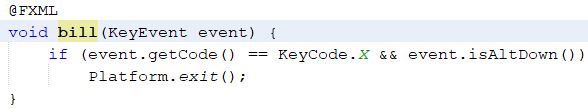


Próbáld ki!

Szeretnénk magunknak egy kiskaput hagyni, hogy adatbevitel nélkül is ki tudjunk lépni az Alt+x megnyomásával, de ezt nem fogjuk elárulni a felhasználóknak.

Ehhez először rendeld hozzá a Scene Builderben a GridPane elemnél a billentyűlenyomáshoz a bill metódust.

Utána készítsd el a bill metódust a controller osztályban:



Figyeld meg, hogyan adjuk meg, hogy az Alt-nak lenyomva kell lennie! Próbáld ki!

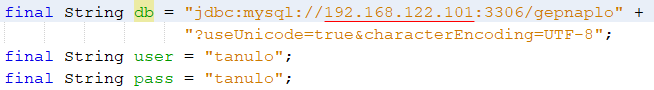
## Távoli adatbázis

Az eddigiekben a helyi adatbázist használtuk. A valóságban azonban egy szerveren fut, amelynek IP címe 192.168.122.101

Most ehhez fogunk kapcsolódni tanulo néven. A jelszó is tanulo lesz.

Először állítsd le a helyi szervert!

Utána írd át a db, a user és a pass változók értékeit a DB osztályban:



Próbáld ki!

## Megjegyzés

Ahhoz, hogy egy MySQL adatbázis más gépekről is elérhető legyen, a MySQL-t futtatató gép tűzfalán engedélyezni kell a 3306-os portot mindkét irányban.

## Jar fájl készítése

Készíts JAR fájlt a programból!

Figyeld meg, hogy a dist mappában van egy lib mappa, és abban egy mysql-connector-java-5.1.23-bin.jar nevű fájl!

A program csak akkor működik, ha ez ott van!