# Gépnapló

Ebben a leckében a tanárok számára készítünk egy programot, amely kilistázza a bejelentkezéseket a gepnaplo adatbázisban, és törli a 30 napnál régebbi rekordokat.

## Előkészületek

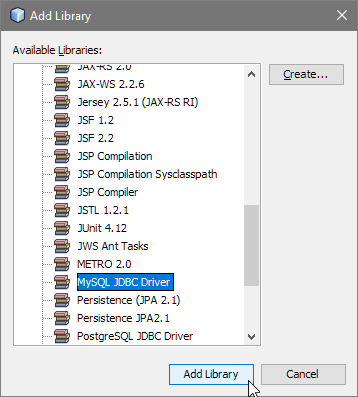
1. Indítsd el a Wamp servert, és nyisd meg a PHPMyAdmin programot! Jelentkezz be root-ként! (nincs jelszó)
2. Hozz létre egy *gepnaplo* nevű adatbázist!
3. Importáld az adatbázis tartalmát a gepnaplo.sql fájlból!
4. Nézd meg a *gepek* tábla szerkezetét!
5. Hozz létre egy új, *tanar* nevű felhasználót, aki bármely gépről bejelentkezhet a *tanar122* jelszóval!
6. Állítsd be a *tanar* felhasználó jogait úgy, hogy csak a *gepnaplo* adatbázisban tudjon lekérdezni és törölni (semmi mást)!
7. Írj be néhány bejelentkezést! Legyen 30 napnál régebbi, 10 nappal ezelőtti, 2 nappal ezelőtti, mai és 45 percem belüli is!

## Új projekt

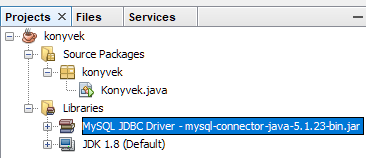
Kezdj egy új Java FXML projektet a NetBeans-ben *Gepnaplo2* néven!

Kattints 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!



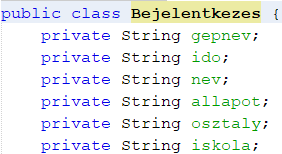
Ezzel hozzáadtad a drivert a projekthez:



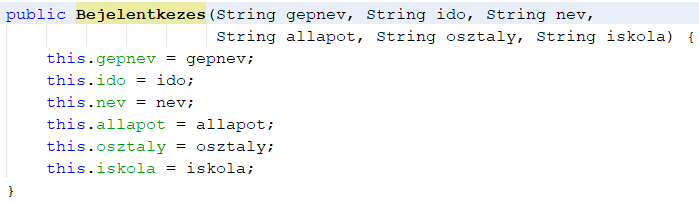
## Bejelentkezés osztály

Egy bejelentkezés adatait a Bejelentkezés osztály egy példányában fogjuk tárolni. Ezek adják a táblázat sorait.

Készíts egy új osztályt Bejelentkezes béven, és add meg az alábbi mezőket:



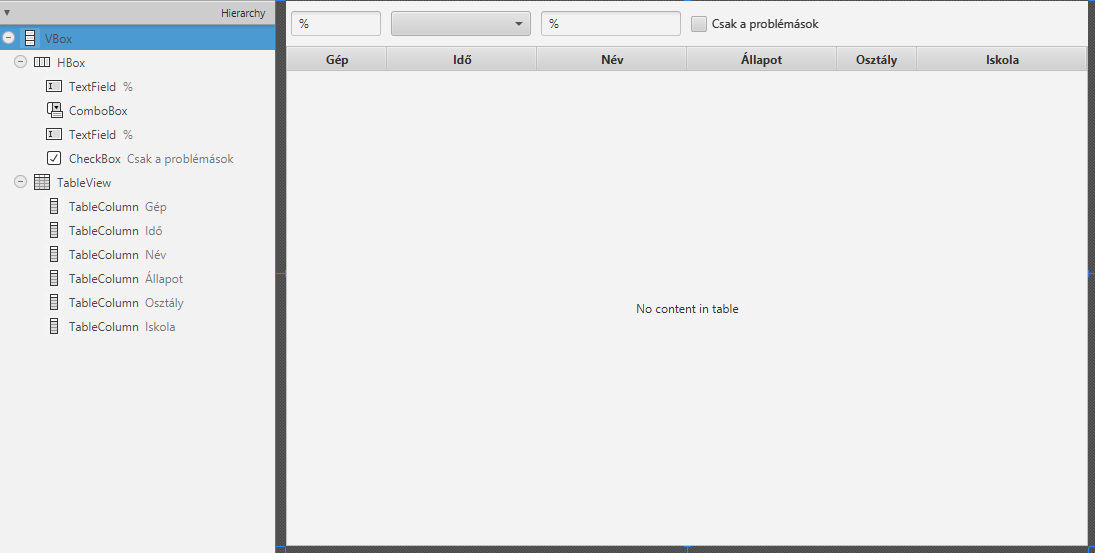
Készíts konstruktort automatikusan (Alt+Insert):



Készíts automatikus getter és setter metódusokat is mindegyik mezőhöz!

## Grafikus felület elkészítése

Készítsd el a grafikus felületet az alábbi minta szerint:



A VBox tárolóban legyen 10 pontos térköz és felül 10 pontos padding! Méretei legyenek számítottak!

A Hbox-ban is 10 pontos térköz legyen, bal oldalon 5 pontos margó, és számított méretek!

A bal felső szövegdoboz preferált szélessége legyen 90 képpont, azonosítója txtGepszuro!

A ComboBox (cbxIdo) legyen 140 képpont széles!

A következő szövegdoboz (txtNevszuro) szélessége 140 pont legyen!

A jelölőnégyzet azonosítója chkProb legyen!

A TableView (tblGepek) szélessége számított, magassága 500 képpont, szélessége számított legyen!

A táblázat oszlopainál kapcsold ki a rendezhetőséget!

Az egyes oszlopok adatai:

* Gép: oGep, 100 pont,
* Idő: oIdo, 150 pont,
* Név: oNev, 150 pont,
* Állapot: oAllapot, 150 pont,
* Osztály: oOsztaly, 80 pont,
* Iskola: oIskola, 170 pont.

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!

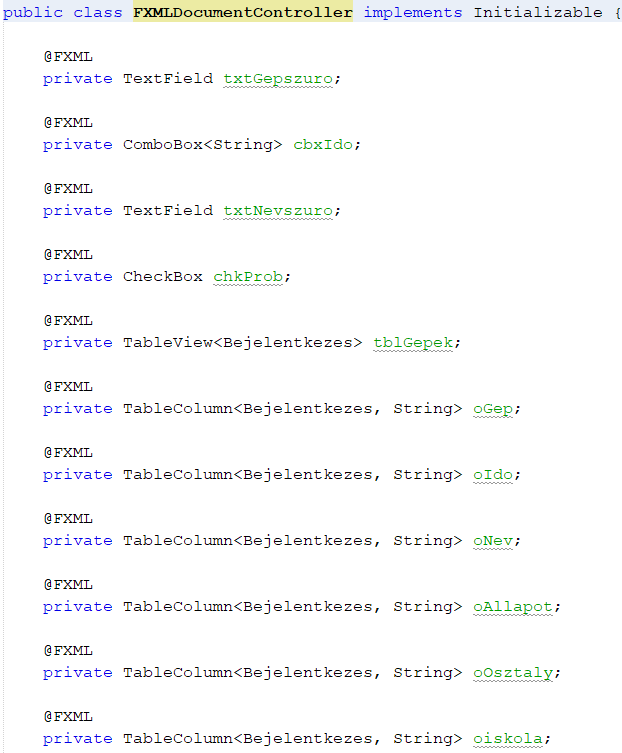
Mentsd a jelenetet, és válts vissza a NetBeans-re!

Nézd át a generált FXML dokumentumot!

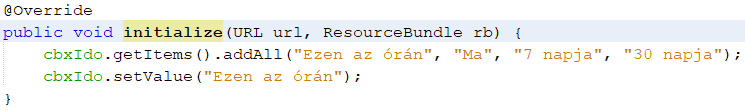
## Controller osztály

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

Add meg a típust a szükséges helyeken:

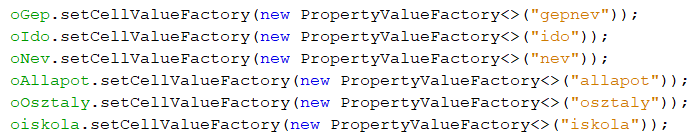


Add meg a ComboBox elemeit:



Próbáld ki!

Folytasd az initialize metódust annak megadásával, hogy a Bejelentkezes osztály melyik mezője a táblázat melyik oszlopába kerüljön!



Válts a gepnaplo2 osztályra! Állítsd be az ablak címét, és kapcsold ki az átméretezhetőségét!

Futtasd a programot!

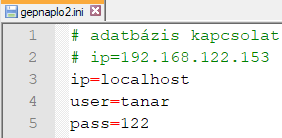
## Beállítások tárolása fájlban

Hozz létre egy új, DB nevű osztályt! Ebben lesznek az adatbázissal kapcsolatos műveletek.

A programot több tanteremben is szeretnénk alkalmazni. Ezekben különböző IP címeken vannak a szerverek. Ráadásul jó lenne, ha a felhasználónevet és a jelszót is különbözőre lehetne állítani az egyes termekben.

Ha ezeket az adatokat beleírjuk a program kódjába, akkor minden teremnél új programváltozatot kell készítenünk. Ezért inkább egy szöveges fájlban tároljuk őket, amelyet a program induláskor beolvas.

A fájl neve tetszőleges lehet, most *gepnaplo2.ini* lesz. Készítsd el ezt a fájlt a projekt mappájába:



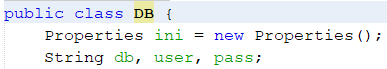
A #-tel kezdődő sorok megjegyzések, amelyeket nem vesz figyelembe a program.

Egyelőre a helyi gépről veszi az adatokat a program (localhost). Amikor áttérünk a teremben lévő szerverre, csak át kell tenni a #-et a 2. sorból a 3. sor elejére.

A jelszónak csak a végét adjuk meg a fájlban, a program elé fogja tenni a tanar szót. Így nem olvashatja ki mindenki a teljes jelszót a fájlból, és a rendszergazda módosíthatja a jelszó végét.

A beolvasott tulajdonságokat egy Properties típusú változóban tároljuk. Ennek getProperty metódusával lehet lekérdezni az egyes tulajdonságokat.

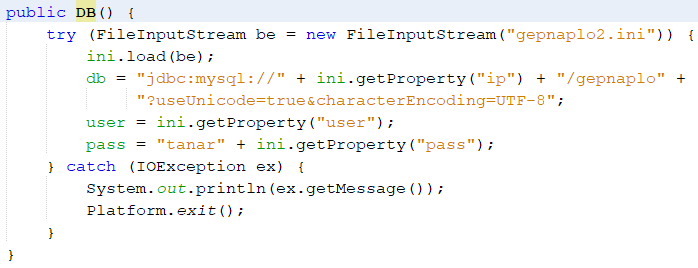
Hozd létre a DB osztályban a szükséges változókat:



A beolvasást a konstruktorban fogjuk elvégezni.

A fájlt FileInputStream-ként kell megnyitni. A megnyitott fájlból az ini objektum load metódusával lehet beolvasni az adatokat.

Írd ezt a konstruktorba:



A konstruktor a példányosításkor fog lefutni. Válts a controller osztályra, és az initialize elé vagy után írd be ezt a sort:



Futtasd a programot! Ellenőrizd a beolvasást egy töréspont segítségével!

## Lekérdezés összeállítása

A programnak az ablak tetején megadott feltétek szerint kell lekérdezést készítenie.

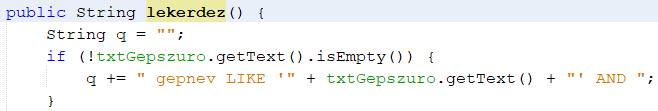
A lekérdezés feltételét – bonyolultsága miatt – a lekerdez metódusban állítjuk össze a controller osztályban.

A lekérdezés feltételét a q változóba tesszük, amely induláskor üres.

A txtGepszuro mezőbe lehet beírni a listában megjelenő gépek nevét. A % jel tetszőleges számú karaktert helyettesít. Például %6 azokat a gépeket jeleníti meg, amelyeknek neve 6-ra végződik.

Ha a txtGepszuro mező nem üres, akkor a feltételbe beírjuk a megfelelő részt.

Így a metódus eleje ez lesz:



Figyeld meg, hogy AND-et írtunk a q végére, mert így további feltételek kapcsolhatók hozzá! A lekérdezés összeállításánál nagyon figyelj a szóközökre! (Inkább több szóköz legyen, mint kevesebb!)

A feltétel következő részében az időre szűrünk. Eközben használunk néhány dátummal és idővel kapcsolatos SQL függvényt.

Az első ilyen függvény a NOW(). Ennek értéke az aktuális dátum és idő.

Válts a PHPMyAdmin programra!

Futtasd le a következő parancsot:

SELECT NOW()

Az eredményből kivághatjuk a dátumot a **DATE()** függvény segítségével. Próbáld ki ezt is:

SELECT DATE(NOW())

Ugyanígy az idő is kivágható a **TIME()** függvénnyel.

A következő függvény a **TIMEDIFF**(), amely két időpont különbségét adja meg óra, perc, másodpercben.

Programunk alaphelyzetben az utolsó 45 perc bejelentkezéseit fogja listázni. Ehhez a következő feltételt fogjuk használni:  
TIMEDIFF(NOW(),ido)<'00:45'

Uána futtasd a következő parancsot:

SELECT \* FROM gepek WHERE TIMEDIFF(NOW(),ido)<'00:45'

A következő választható idő a mai nap lesz. Futtasd le a következő SQL parancsot, amely a mai bejelentkezéseket listázza:

SELECT \* FROM gepek WHERE DATE(ido)=DATE(NOW())

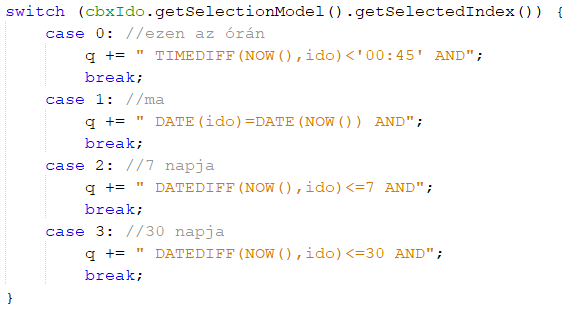
A **DATEDIFF**() függvény két időpont különbségét adja meg napokban.

Listáztasd ki azokat, akik az utóbbi 7 napban jelentkeztek be:

SELECT \* FROM gepek WHERE DATEDIFF(NOW(),ido)<=7

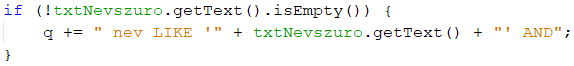
Listáztasd önállóan azokat, akik az utóbbi 30 napban jelentkeztek be!

Ezek alapján a lekerdez metódust így kell folytatnod:

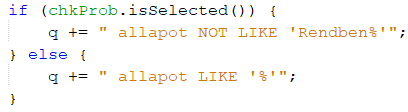


Figyeld meg, hogy a ComboBox-ban kiválasztott elem indexét kérdezzük le, mert a switch utasításban számot kell megadni!

A tanuló nevének szűrésére hasonló feltételt adunk meg, mint a gép nevére:



A „Csak a problémások” jelölőnégyzet bekapcsolt állapotában csak azok a rekordok jelennek majd meg, amelyek állapota nem a *Rendben* szóval kezdődik. Folytasd így a lekerdez metódust:



Figyeld meg, hogy itt már nem írtunk AND-et a q végére! Ezzel lezártuk a feltételt.

Végül összeállítjuk és visszaadjuk a SELECT parancsot:

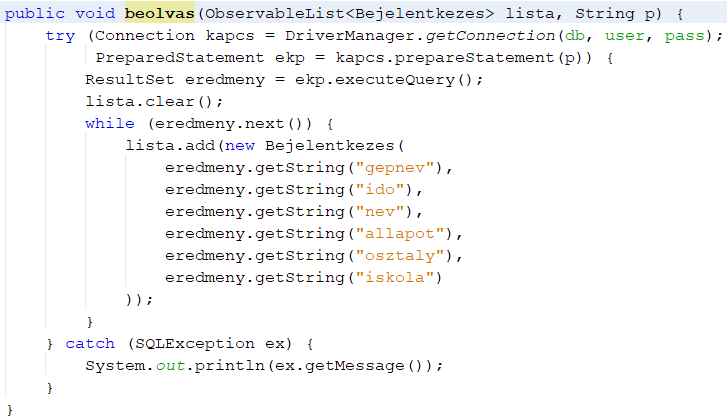


Figyeld meg, hogy idő szerint csökkenő sorrendet állítunk be!

## Rekordok beolvasása az adatbázisból

Az adatok beolvasását az adatbázisból a szokásos módon végezzük, kivéve, hogy a lekérdezéshez használt parancsot paraméterként adjuk át.

Írd ezt a DB osztályba:

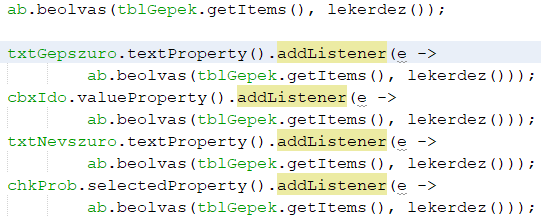


Figyeld meg, hogy a beolvasás előtt töröljük a listát!

Válts vissza a controller osztályra!

Hívd meg a beolvas metódust az initialize metódusból, majd rendeld hozzá a következő eseményekhez is a beolvas metódus végrehajtását:

* változás a txtGepszűrő mezőben,
* cbxIdo módosítása,
* változás a txtNevszűrő mezőben,
* chkProb állapotának változása!



Teszteld a programot! Szükség esetén írj be újabb rekordokat!

## Régi rekordok törlése

A 30 napnál régebbi adatokra nincs szükségünk, ezért a program induláskor törölni fogja őket.

Válts át a PHPMyAdmin programra!

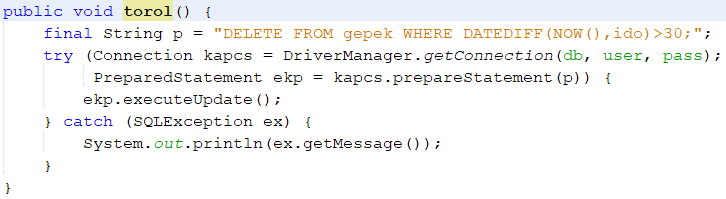
Utána próbáld ki a következő parancsot:

DELETE FROM gepek WHERE DATEDIFF(NOW(),ido)>30

Figyelj az összehasonlítás irányára! A PHPMyAdmin is rá fog kérdezni, hogy tényleg akarod-e a műveletet.

Ellenőrizd, hogy letörlődött-e a 30 napnál régebbi bejelentkezés (és a többi megmaradt-e)!

Készítsd el a torol metódust a DB osztályban:



Hívd meg a torol metódust a controller osztály initialize metódusában a beolvas metódus előtt!

A teszteléshez írj be ismét egy 30 napnál régebbi bejelentkezést, majd indítsd el a programot! Ellenőrizd, hogy törlődött-e a régi rekord!

## 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 ezek ott vannak!

Ezen kívül szükség van a gepnaplo2.ini fájlra is. Másold ezt a dist mappába!

Vigyázat! Új JAR fájl készítésekor minden törlődik a dist mappából! Utána újra oda kell másolnod a gepnaplo2.ini fájlt!

Próbáld ki a JAR fájl működését!

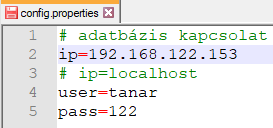
## Távoli adatbázis

Az eddigiekben a helyi adatbázist használtuk. A valóságban azonban egy másik gépen lévő szerveren fut, amelynek IP címe 192.168.122.153

Most ehhez fogunk kapcsolódni tanar néven. A jelszó tanar122 lesz.

Először állítsd le a helyi szervert!

Utána módosítsd a gepnaplo2.ini fájlt:



Próbáld ki!

## Megjegyzés

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