# Leltár 2.

Ebben a leckében a leltar programot fejlesztjük tovább.

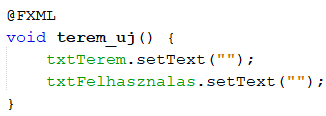
## Előkészületek

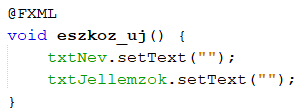
1. Indítsd el a Wamp servert, és nyisd meg a PHPMyAdmin programot! Jelentkezz be root-ként! (nincs jelszó)
2. Készíts egy *nyilvantartas* nevű adatbázist!
3. Importáld a nyilvantartas.sql fájlt!
4. Nézd meg a táblák tartalmát és szerkezetét!
5. Készíts egy *raktaros* nevű felhasználót, akinek a jelszava is *raktaros*, és a *nyilvantartas* adatbázisban az adatokkal kapcsolatos műveletekhez van joga!
6. Nyisd meg a NetBeans-ben az előző alkalommal készített leltar programot! Nézd át a forráskódot!
7. Próbáld ki a programot!

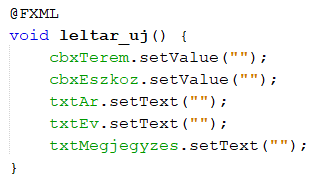
## Új gombok

Az új gombok feladata minden lapon az, hogy alaphelyzetbe állítsa az adatokat a táblázatokon kívüli részekben.

Készítsd el ezeket a controller osztályban, majd próbáld ki őket!



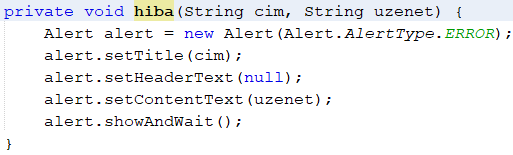




## Hibaüzenet

A továbbiakban szükségünk lesz egy metódusra, amely egy hibaüzenetet jelenít meg a egy párbeszédpanelen.

Írd be ezt a controller osztály végére:



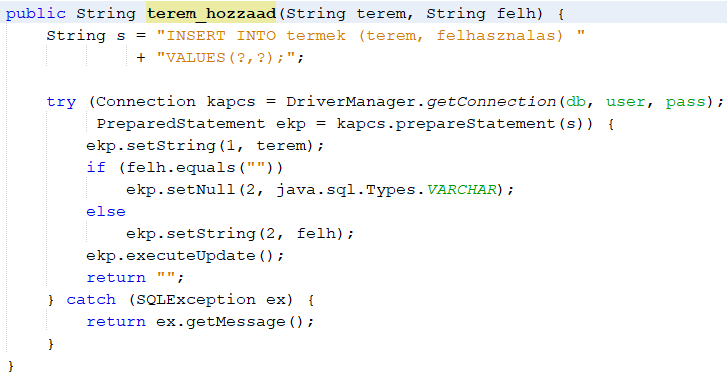
## Hozzáadás

A Hozzáadás gombok segítségével adhatunk hozzá egy rekordot az adattáblákhoz.

Minden hozzáadás gombhoz két metódust készítünk:

* Az egyiket a DB osztályba. Ez az adatbázissal közvetlenül kapcsolatos feladatokat végzi SQl parancs segítségével.
* A másikat a controller osztályba. Ez a a grafikus felülettel kapcsolatos feladatokat végzi, és meghívja a másik metódust.

Nézzük először a terem hozzáadását! Írd ezt a DB osztályba:



Figyeld meg, hogy a nem kötelező felhasznalas mezőnél hogyan kezeljük a lehetséges null értéket!

Mit ad vissza a metódus?

Ezután a controller osztályban készítsd el az alábbi metódust:

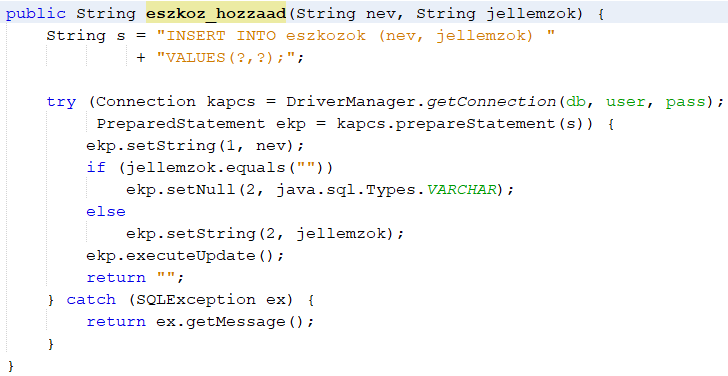


A metódus első fele beolvassa és ellenőrzi a bevitt értékeket. A hibákat párbeszédpanelen jeleníti meg az előbb elkészített hiba metódus segítségével.

A második felében először meghívjuk a DB terem\_hozzaad metódusát. Ha sikeres volt a művelet, és üres stringet kaptunk vissza, akkor újra beolvassuk a rekordokat a táblából, majd megkeressük és kijelöljük az új elemet. A keresésnél kihasználjuk, hogy a teremazonosítók egyediek.

Teszteld a terem hozzáadását! Próbáld ki a lehetséges hibákat is!

Ezután készítsd el az eszkoz\_hozzaad metódusokat a DB osztályban és a controllerben is (érdemes másolni):





Teszteld az eszközök hozzáadását! Próbáld ki a lehetséges hibákat is!

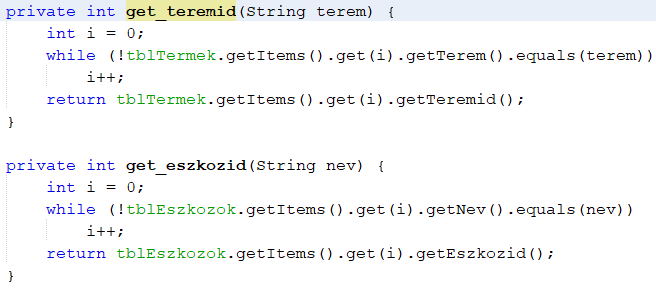
Készísd el a leltar\_hozzaad metódust a DB osztályban:



Figyeld meg, hogyan kezeljük a lehetséges null értékeket!

A controller osztályban szükségünk lesz két metódusra, amelyek a terem azonosítója, illetve az eszköz neve alapján megkeresik a hozzájuk tartozó teremid-t, illetve eszkozid-t.

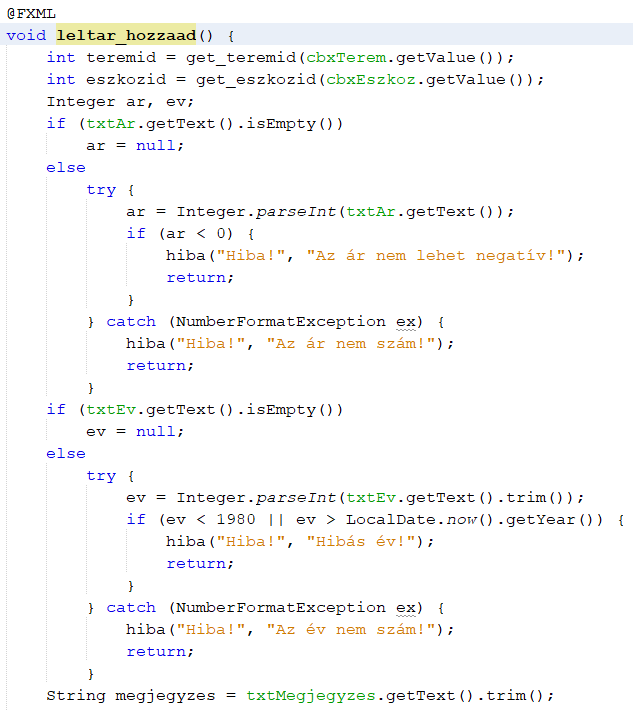
Készítsd el először ezeket a controller osztályban:



Gondold végig a működésüket! Vigyázat! Ezek a metódusok csak akkor működnek helyesen, ha a táblázatban létező teremszámot, illetve eszköznevet adunk meg!!!

Itt egy kicsit bonyolultabb a helyzet, mert az árat és az évet át kell váltani egész számmá, és ezek ráadásul nem is kötelezőek.

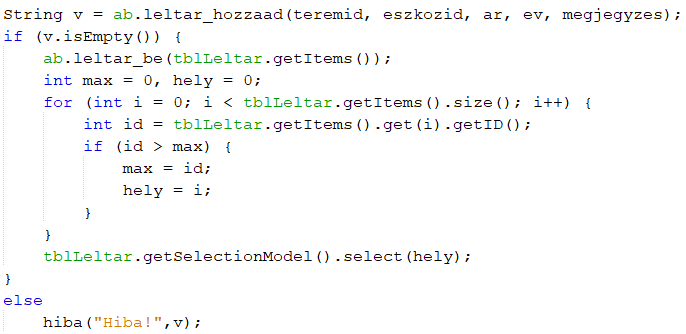
Nézzük először a beolvasást és az ellenőrzést:



Figyeld meg, hogyan váltjuk át a szöveget egész számmá, és hogyan ellenőrizzük!

Milyen évszámokat fogadunk el?

Ezután folytasd a metódust a művelet végrehajtásával és az új tétel kijelölésével:



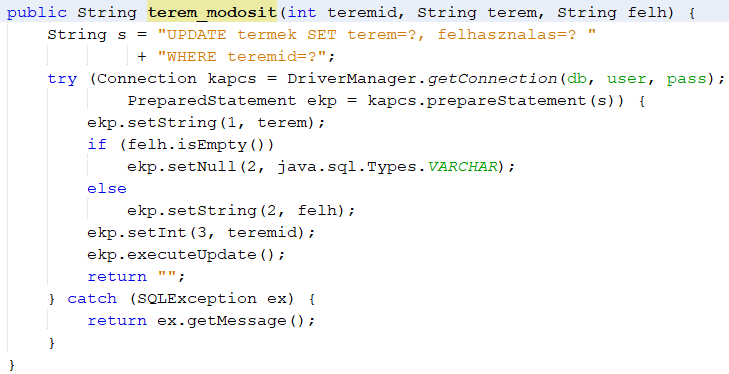
Az új tétel keresésénél azt használjuk ki, hogy az új tétel azonosítója (id) a legnagyobb.

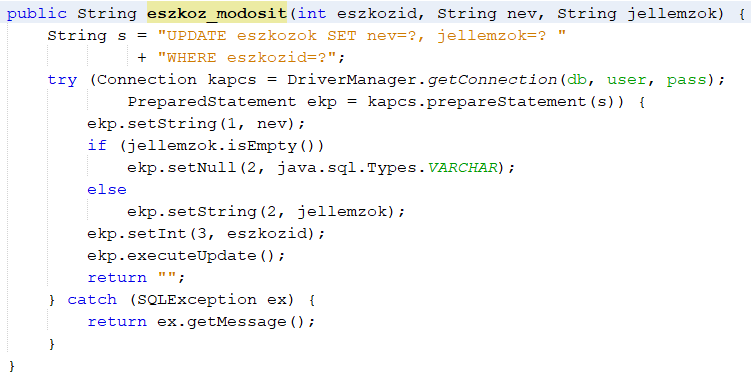
Teszteld a tételek hozzáaádását! Próbáld ki a lehetséges hibákat is!

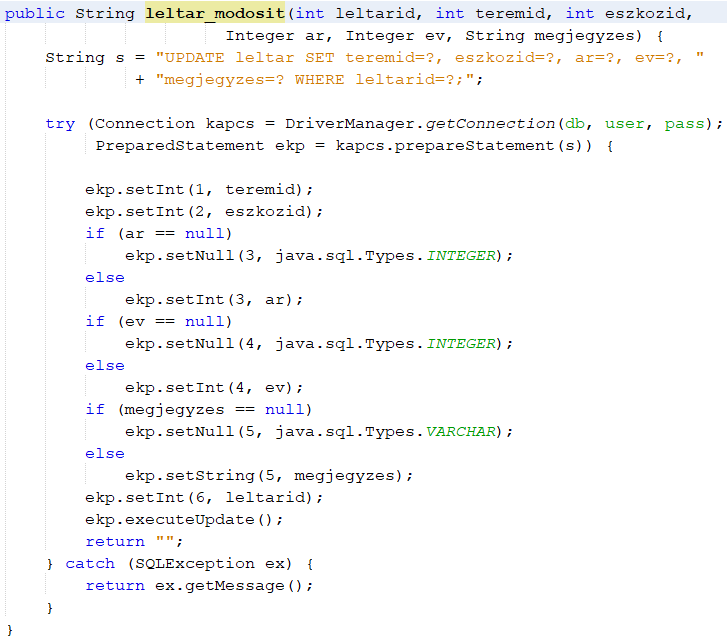
## Módosítás

A módosítás hasonlít a hozzáadásra. Ennél is két-két metódust kell megadnunk egy-egy művelethez.

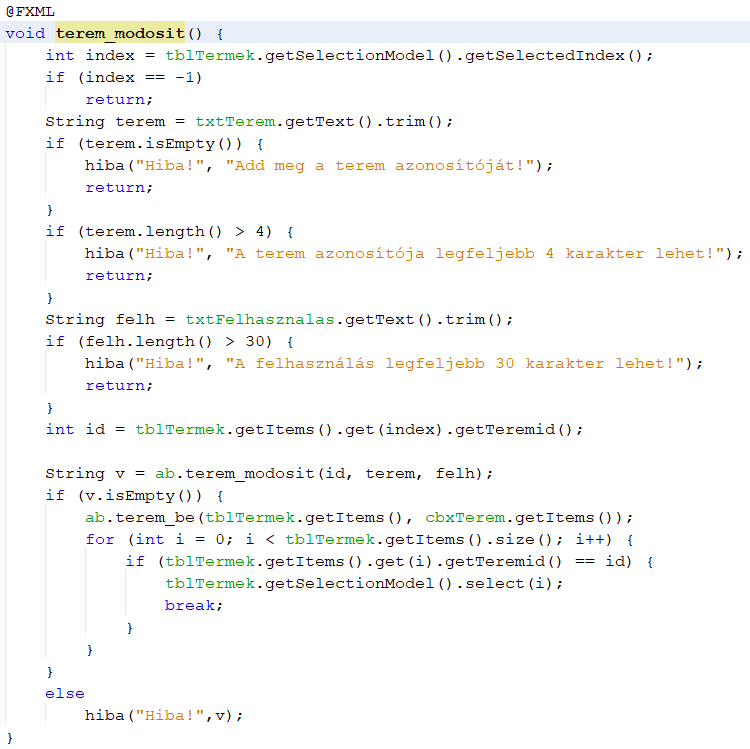
A DB osztályban készítsd el az alábbi metódusokat:







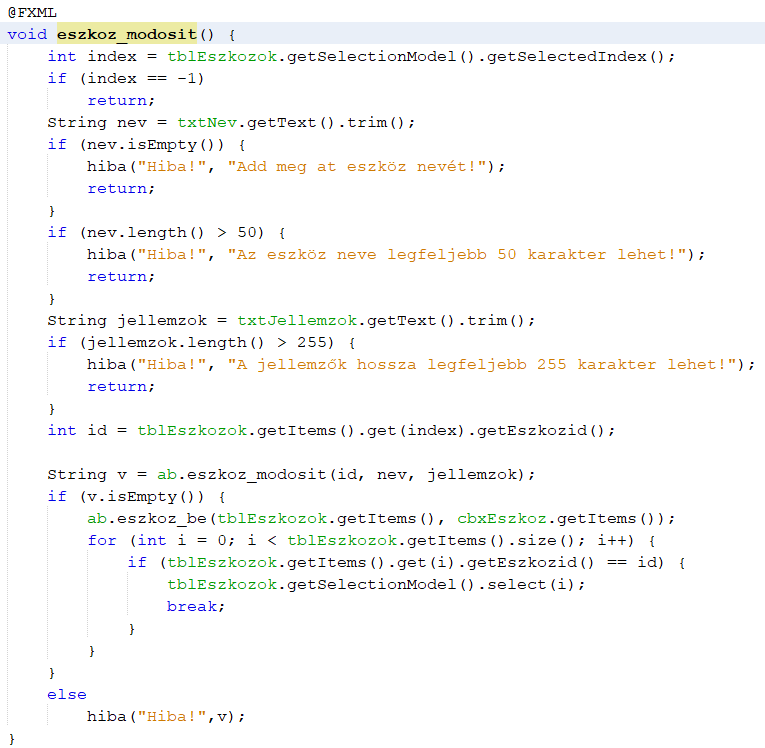
A controller osztályban először hozd létre a az alábbi metódust:



Az ellenőrzéseket átmásolhatod a terem\_hozzaad metódusból.

Teszteld a termek módosítását a lehetséges hibákkal együtt!

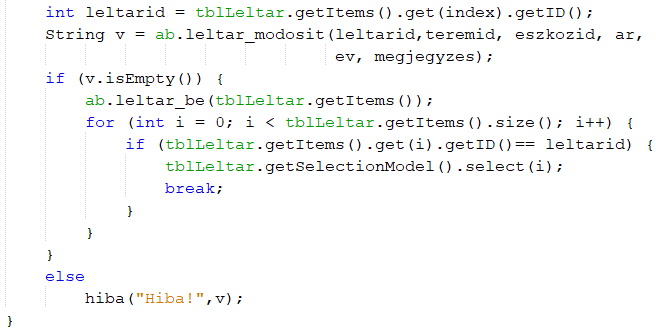
Az eszközök módosítása így történik:



Teszteld ezt is!

Végül nézzük a leltári tételek módosítását:

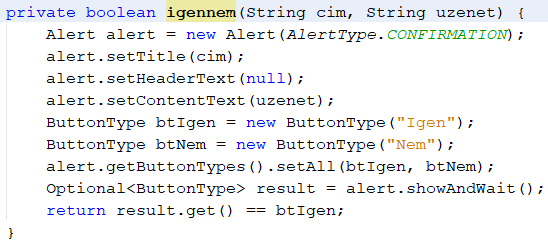




Teszteld ezt is!

## Megerősítés

A törlések előtt rá fogunk kérdezni, hogy tényleg törölni szeretne-e a felhasználó. Ehhez kell majd a következő metódus, amelyet a controller osztály végén helyezz el:



## Törlés

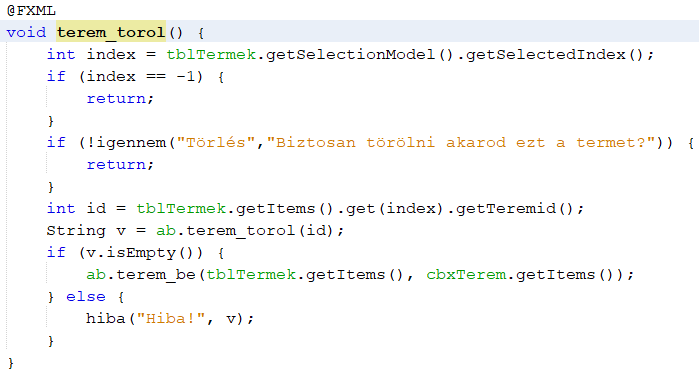
A törlésekhez is két-két metódus kell.

Írd ezeket a DB osztályba:



Mit adnak vissza a metódusok?

A controller osztályban először a terem\_torol metódust hozd létre:



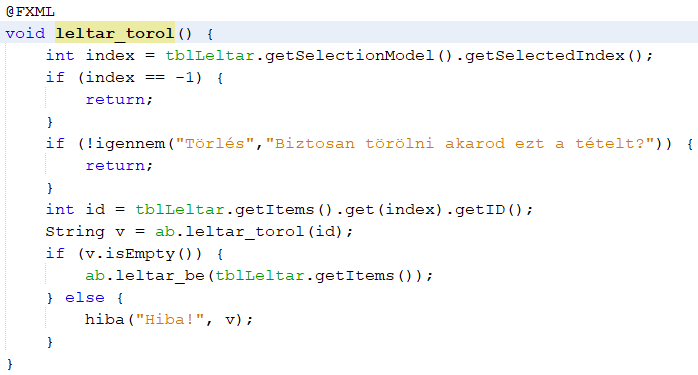
Próbáld ki!

Ehhez nagyon hasonlít az eszkoz\_torol metódus:



Próbáld ki ezt is!

Mai utolsó metódusunk a leltar\_torol metódus:



Teszteld!

Figyeld meg, hogy a törlés után nem jelölünk ki sort a táblázatokban. Így viszont a törölt adatok megmaradnak a tábla melletti részen, és újra hozzáadhatók a táblához, ha mégsem szeretnénk törölni.

## 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, ezért a dist mappa teljes tartalmát kell terjeszteni!

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

Próbáld ki a JAR fájl működését!

## Távoli adatbázis

Az eddigiekben a helyi adatbázist használtuk. Ha máshol fut a szerver, módosítsd az IP címet!