# Közös költség 2.

Ebben a leckében a közös költséget nyilvántartó 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 *tarsashaz* nevű adatbázist!
3. Importáld a *tarsashaz.sql* fájlt!
4. Nézd meg a táblák tartalmát és szerkezetét!
5. Készíts egy *kepviselo* nevű felhasználót, akinek a jelszava is *kepviselo*, és a *tarsashaz* adatbázisban az adatokkal kapcsolatos műveletekhez van joga!
6. Nyisd meg a NetBeans-ben az előző alkalommal készített kozoskoltseg programot! Nézd át a forráskódot!
7. Próbáld ki a programot!

## Hibaüzenetek

A továbbiakban szükségünk lesz egy metódusra, amely egy hibaüzenetet jelenít meg a egy párbeszédpanelen, és egy másikra, amely rákérdez egy műveletre.

Ezeket egy külön projektben elkészítettük. Nyisd meg a Panel projektet is!

Nézd meg a Panel osztály kódját, majd készíts JAR fájlt! Másold át a JAR fájlt a *tarsashaz* projekt mappájába!

A hiba metódust így hívhatod meg:

Panel.hiba(cím, üzenet);

Az igennem metódus hívása:

boolean v = Panel.igennem(cím, üzenet);

v akkor igaz, ha a felhasználó az Igen gombot választja.

Ha importálod a Panel osztály statikus metódusait, akkor nem kell megadnod a metódus neve előtt:



Ezután válts a *Tarsashaz* projektre, és add hozzá az előbb elkészített JAR fájlt a könyvtárakhoz! (Add JAR/Folder)

Ezután módosítsd a DB osztály beolvasó metódusai catch ágában lévő utasításokat erre (és importáld a Panel osztályt):

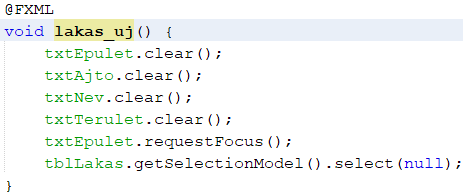
hiba("Hiba!",ex.getMessage());

## Új gomb

Az új gomb feladata a Lakások lapon az, hogy alaphelyzetbe állítsa az adatokat a táblázaton kívüli részben, rávigye a kurzort az első mezőre, és megszüntesse a kijelölést a táblázatban.

Ez utóbbira azért van szükség, hogy hogy a felhasználó véletlenül ne módosíthassa a kijelölt sort.

Készítsd el ezt a controller osztályban, majd próbáld ki!



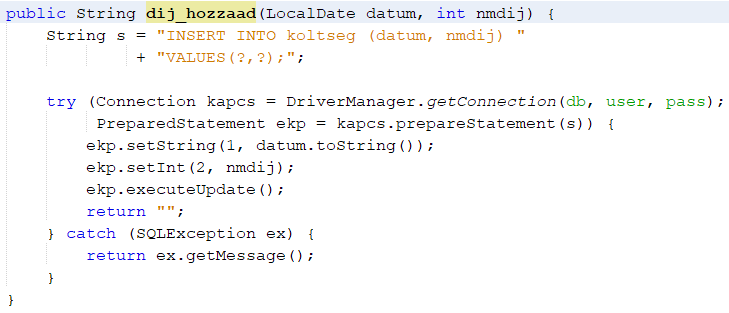
## Hozzáadás

A Hozzáadás gombok segítségével adhatunk hozzá egy rekordot az adattáblákhoz.

Minden hozzáadás gombhoz két metódust készítünk:

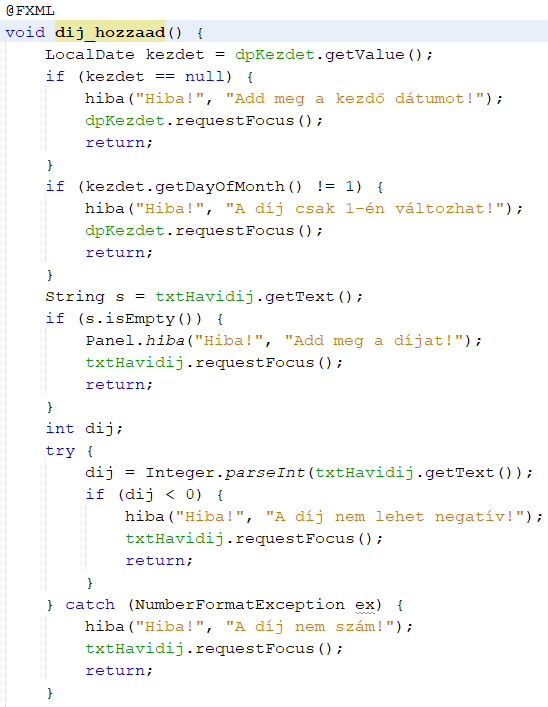
* Az egyiket a DB osztályba. Ez az adatbázissal közvetlenül kapcsolatos feladatokat végzi SQL parancs segítségével.
* A másikat a controller osztályba. Ez a a grafikus felülettel kapcsolatos feladatokat végzi, és meghívja a másik metódust.

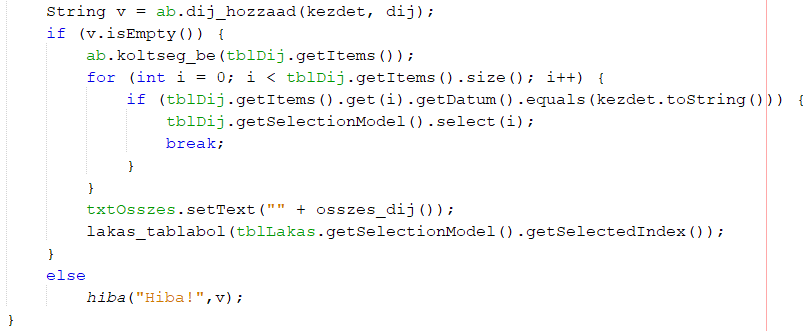
Nézzük először a díj hozzáadását! Írd ezt a DB osztályba:



Mit ad vissza a metódus?

Ezután a controller osztályban készítsd el az alábbi metódust:



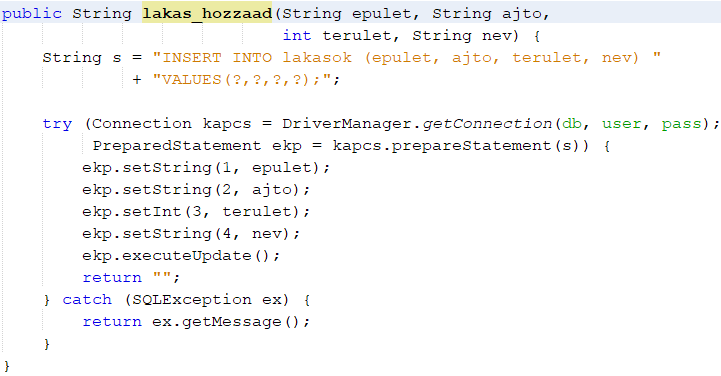


A metódus első fele beolvassa és ellenőrzi a bevitt értékeket. A hibákat párbeszédpanelen jeleníti meg a hiba metódus segítségével.

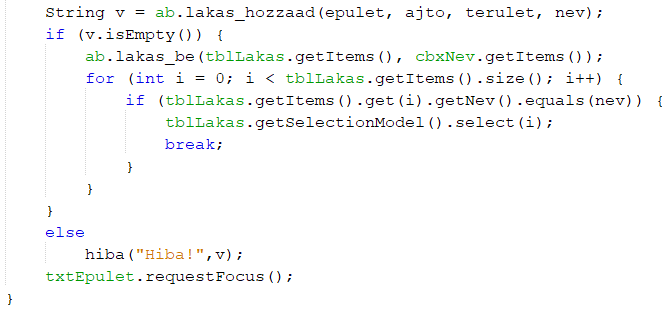
A második felében először meghívjuk a DB koltseg\_hozzaad metódusát. Ha sikeres volt a művelet, és üres stringet kaptunk vissza, akkor újra beolvassuk a rekordokat a táblából, majd kijelöljük a bevitt rekordot. Utána újraszámítjuk az eddigi nm-enkénti díjat és a kijelölt lakás egyenlegét a Lakások lapon.

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

Ezután készítsd el a lakas\_hozzaad metódusokat a DB osztályban és a controllerben is:

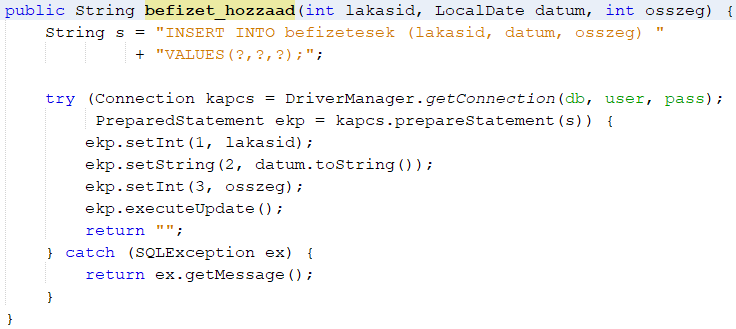






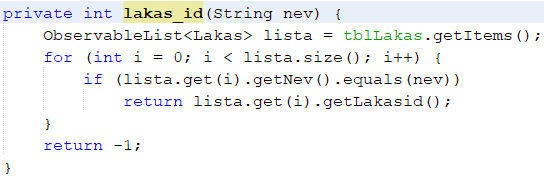
Teszteld a lakások hozzáadását! Próbáld ki a lehetséges hibákat is!

Készísd el a befizet\_hozzaad metódust a DB osztályban:



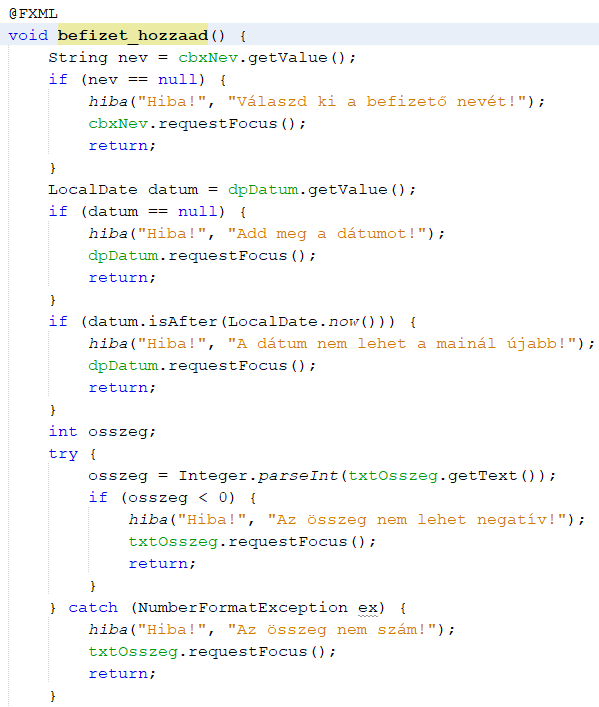
A controller osztályban szükségünk lesz egy metódusra, amely a tulajdonos neve alapján megkeresi a hozzá tartozó lakasid-t. (Feltételezzük, hogy minden lakásnál más név van megadva.)

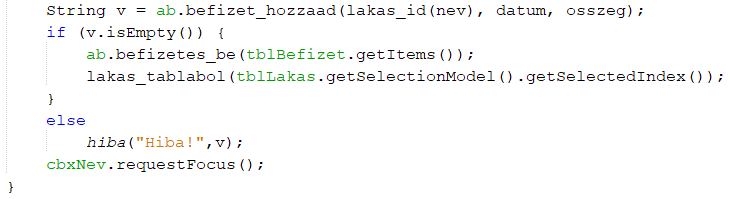
Készítsd el először ezt a controller osztályban:



Gondold végig a működését! Milyen értéket ad vissza, ha a táblázatban nem szereplő nevet adunk meg?

Ezután készítsd el a befizet\_hozzaad metódust a controller osztályban:



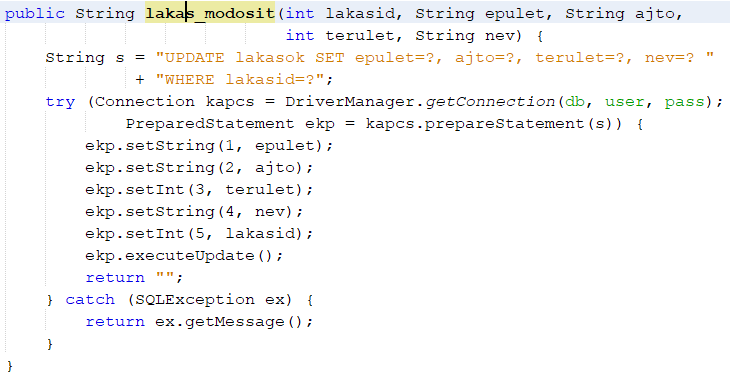


Teszteld a hozzáadást! Milyen dátumot fogad el a metódus?

## Módosítás

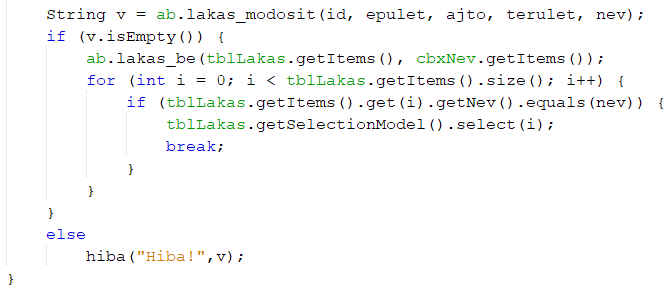
A módosítás hasonlít a hozzáadásra. Ennél is két metódust kell megadnunk egy művelethez.

A DB osztályban készítsd el az alábbi metódust:



Utána írd ezt a controller osztályba:



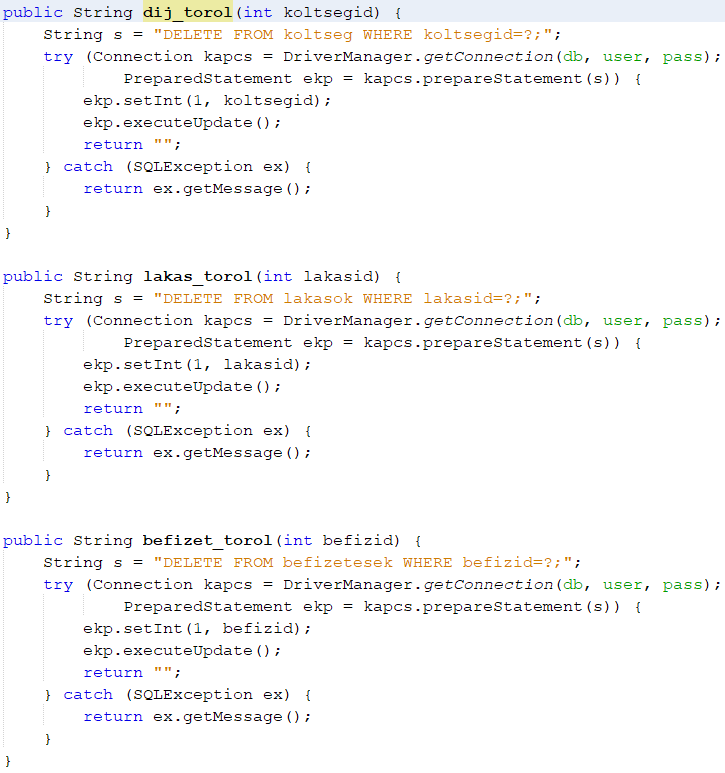


Teszteld a lakások módosítását a lehetséges hibákkal együtt!

## Törlés

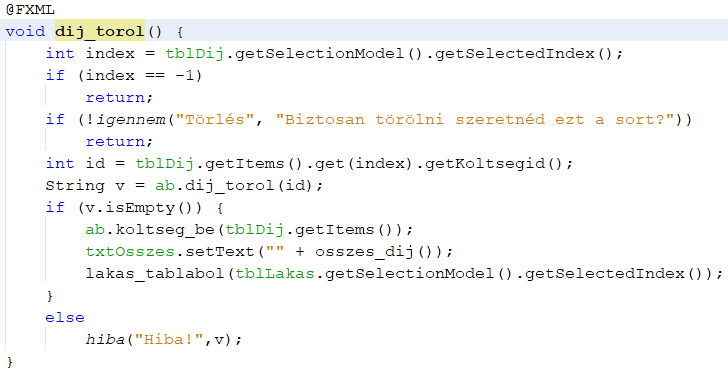
A törlésekhez is két-két metódus kell.

Írd ezeket a DB osztályba:



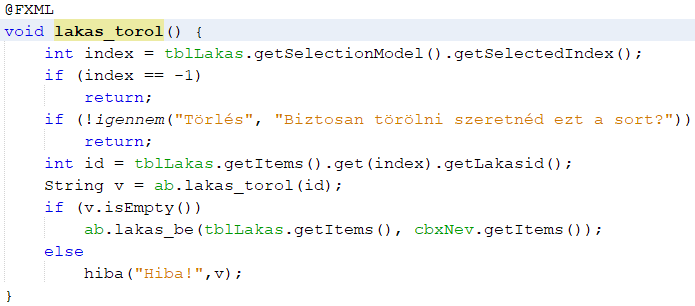
Mit adnak vissza a metódusok?

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



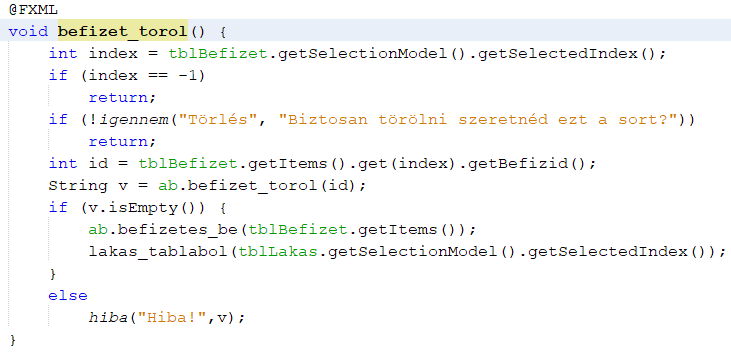
Próbáld ki! Figyeld meg, hogy törlés után újraszámolja a befizetéseket és az egyenleget a másik lapon kijelölt lakásra!

Ehhez hasonlít a lakas\_torol metódus:



Próbáld ki ezt is! Mi történik, ha egy olyan lakást próbálsz törölni, amelyhez tartozik befizetés?

Végül írd be, és próbáld ki a befizet\_torol metódust is:

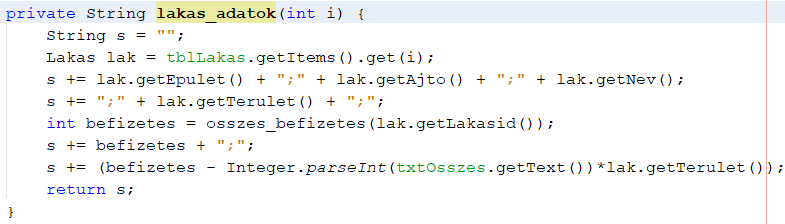


Teszteld! Figyeld meg, hogy törlés után újraszámolja a befizetéseket és az egyenleget a másik lapon kijelölt lakásra!

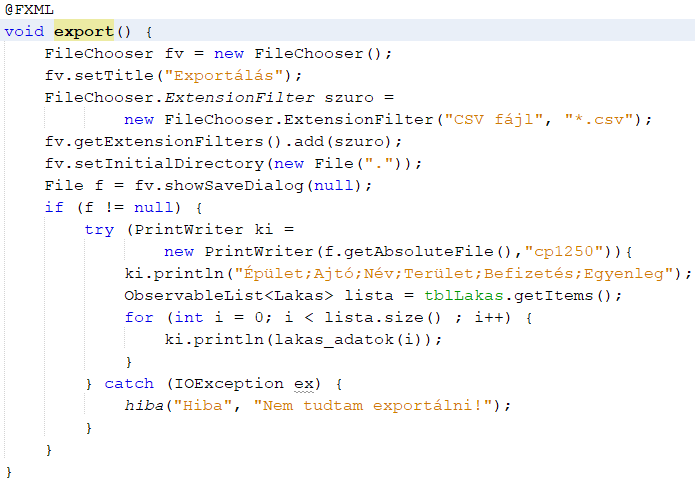
## Exportálás

Az export metódus a lakások adatait és egyenlegét egy CSV fájlba exportálja, amelyet Excel-lel is meg lehet nyitni.

Ehhez először egy olyan metódust írunk, amely a lakások adatait összeállítja egy sorba, pontosvesszőkkel elválasztva. Írd ezt az export metódus elé:



Ennek felhasználásával az export metódus:



Próbáld ki!

## 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 és egy Panel.jar nevű fájl!

A program csak akkor működik, ha ezek ott vannak!

Próbáld ki a JAR fájl működését!

## Távoli adatbázis

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