# Közös költség

Ebben a leckében egy olyan programot készítünk, amellyel egy társasház közös kötségének befizetéseit lehet nyilvántartani.

A programnak a következő adatokat kell nyilvántartania:

* a közös költség négyzetméterenkénti díjai és ezek bevezetésének időpontjai (pl. 2019. január 1-től 160 Ft/nm) – csak 1-jétől változhat,
* lakások adatai: épület, ajtó, tulajdonos neve, lakás területe nm-ben (pl. A épület, 1. ajtó, Balázs Róbert, 90 nm)
* Befizetések adatai: tulajdonos neve, befizetés dátuma és összege (pl. Balázs Róbert, 2019. március 8-án befizetett 14 500 Ft-ot)

A program írja ki, hogy eddig összesen mennyi közös költséget kellett fizetni nm-enként!

Ha kijelölünk egy lakást, akkor írja ki ennek a lakásnak az összes befizetését és egyenlegét! Az összes lakás egyenlegét lehessen exportálni egy CSV fájlba!

A befizetéseknél a nevet ComboBox-ból lehessen kiválasztani, és ajánlja fel az aktuális dátumot!

## Adatbázis tervezése

Tervezd meg a szükséges adattáblákat és mezőket! Ügyelj a redundancia elkerülésére!

Add meg az elsődleges és az idegen kulcsokat!

### Megoldás:

koltseg  
koltsegid: int PK AI,  
datum: date,  
nmdij: int

lakasok  
lakasid: int PK AI,  
epulet: char(1),  
ajto: varchar(2),  
terulet: int,  
nev: varchar(50)

befizetesek  
befizid: int PK AI,  
lakasid: int FK -> lakasok.lakasid (+index),  
datum: date,  
osszeg: int



A lakasok táblában az epulet és az ajtó együttesen egyedi kell, hogy legyen! (Nem lehet két olyan lakás, amelynél az épület és az ajtó is megegyezik.)

## Adatbázis elkészítése

Készíts egy új adatbázist *tarsashaz* néven!

Importáld a kapott *tarsashaz.sql* fájlt!

Állítsd be a lakas táblában az épület és az ajtó egyediségét egy index-szel!

Készíts egy *kepviselo* nevű felhasználót, akinek a jelszava is *kepviselo*, és a *tarsashaz* adatbázisban az adatokkal kapcsolatos műveletekhez van joga!

## Grafikus felület

Készíts a NetBeans-ben egy új FXML projektet *kozoskoltseg* néven!

Add hozzá a projekthez a MySQL JDBC Drivert!

A grafikus felületet három füllel szeretnénk elkészíteni úgy, hogy minden fülön:

* táblázatban jelenjenek meg az adatok,
* a Lakások lapon a kijelölt sort külön lehessen szerkeszteni,
* a Lakások lapon legyen Új, Hozzáad, Módosít és Töröl gomb, a többin pedig Hozzáad és Töröl gomb!

Másold be a projektbe a kapott FXMLDocument.xml fájlt, majd nyisd meg a Scene Builderben!

Nézd végig az elemeket! Milyen beállításokat alkalmaztunk, milyen azonosítókat adtunk meg, és mikor milyen eseménykezelőket indítunk el?

Ellenőrizd a controller megadását, és másold át a szükséges kódot a controller osztályba!

Állítsd be az ablak címét, és kapcsold ki az átméretezhetőségét a Kozoskoltseg osztályban!

## Osztályok a táblázatokhoz

A táblázatok soraihoz osztályokat kell készítenünk:

* egy Dij nevű osztályt a tblDij táblázathoz,
* egy Lakas nevű osztályt a tblLakas táblázathoz,
* és egy Befizetes nevű osztályt a tblBefizetes táblázathoz.

Ezeket is elkészítettük, másold őket a forrásfájlok közé, majd nézd meg a tartalmukat!

Figyeld meg, hogy a Befizetes osztályba belevettük a lakástulajdonos nevét is, mert arra is szükség lesz a táblázatban!

Ezután írd be a megfelelő típusokat a controller osztályba a kérdőjelek helyére!

Töröld a felesleges ActionEvent paramétereket a metódusokból!

Add meg az initialize metódusban, hogy a táblázatok melyik oszlopába melyik mező kerül a megfelelő osztályból! Ezt bemásolhatod a factory.txt fájlból.

Futtasd a programot!

## Adatok beolvasása

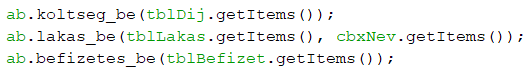
Másold a kapott DB osztályt a forrásfájlok közé, és nyisd meg!

Nézd végig az adatok beolvasásához szükséges metódusokat!

Hozz létre egy új példányt a DB osztályból ab néven a controller osztály elején!



Hívd meg a koltseg\_be, a lakas\_be és a befizetes\_be metódusokat az initialize metódus végéről!



Próbáld ki! Ellenőrizd, hogy jól betöltődtek-e az adatok!

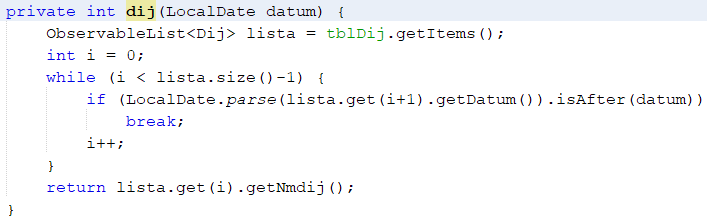
## Eddigi összes díj négyzetméterenként

Az első lapon ki kell íratni, hogy mennyi díjat kellett eddig összesen fizetni négyzetméterenként.

Ezt két lépésben fogjuk megoldani:

* Először készítünk egy metódust, amely tetszőleges dátumra megadja az akkor érvényes díjat a Dijak tábla alapján.
* Utána készítünk egy metódust, amely összegzi a díjakat a mai napig.

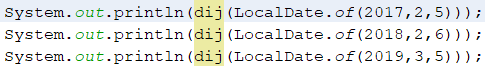
Mindkettőt a controller osztályba kell írni. Az első metódus:



Az isAfter metódus azt adja meg, hogy a . előtti dátum a zárójelben lévő dátum után van-e.

Gondold végig a metódus működését!

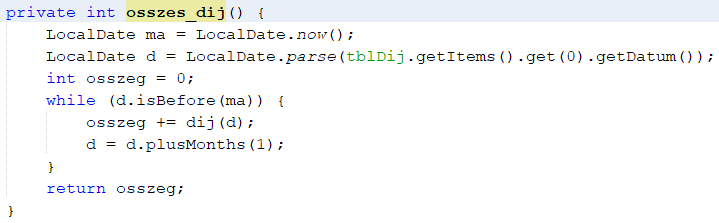
Teszteld a metódust úgy, hogy az initialize metódus végére ideiglenesen beírod a következőket, majd lefuttatod a programot:



A várt eredmények: 150(?), 150, 160.

Utána kommenteld ki!

A második metódus:



Az isBefor metódus azt adja meg, hogy a . előtti dátum a zárójelben lévő dátum előtt van-e.

A plusMonth(n) metódus n hónapot hozzáad a dátumhoz.

Feltételezzük. hogy a fizetés a Díjak tábla első sorában megadott dátumtól indul (és van legalább egy sor).

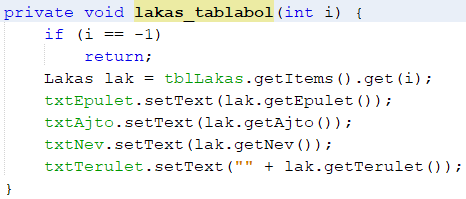
Hívd meg az osszes\_dij metódust az initialize metódus végéről így:



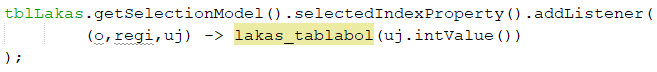
Próbáld ki! Számolj utána az eredménynek!

## Kijelölt táblázatsor megjelenítése

Ezután készítsd el a controller osztályban azt a metódust, amely a Lakások táblázat kijelölt sorát jeleníti meg:

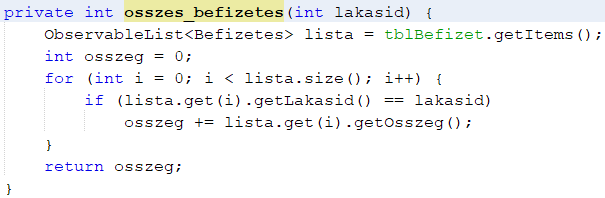


Ezt akkor kell futtatni, amikor a táblázatban megváltozik a kijelölés. Rendeld hozzá őket ehhez az eseményhez az initialize metódus végén:

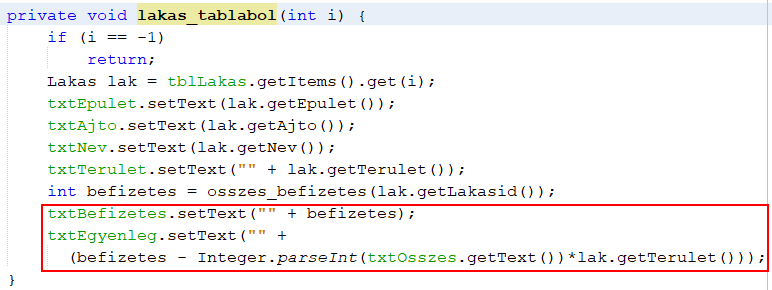


Próbáld ki!

A kijelölt lakás összes befizetését is ki kell számíttatnunk. Készíts ehhez egy metódust a controller osztályba:



Hívd meg a metódust a lakas\_tablabol metódus végén, majd írasd ki az egyenleget is:



Próbáld ki!