# Bevezetés

## Adatbázis létrehozása

Indítsd el a PHPMyAdmin programot, és készíts egy új adatbázist *konyvek* néven!

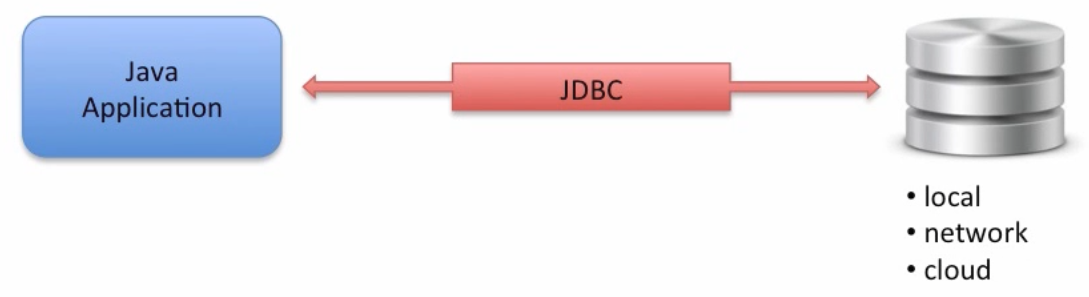
Importáld az adatok táblát az *adatok\_mysql.sql* fájlból!

Nézd meg a tábla szerkezetét és tartalmát!

Figyeld meg, hogy az id mezőn kívül (amely az elsődleges kulcs) bármelyik mező lehet üres (null)!

## JDBC

A JDBC lehetővé teszi a Java programoknak, hogy egy tetszőleges SQL adatbázishoz kapcsolódjanak (pl. MySQL, SQLite, MS SQL Server, Oracle, …).

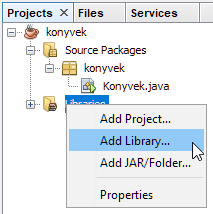


JDBC driver: konnektor egy adott típusú adatbázishoz. Általában az adatbázis fejlesztői biztosítják. A MySQL driver benne van a NetBeans-ben.

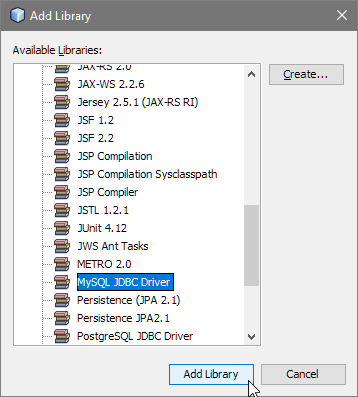
## Új projekt

Kezd egy új Java projektet a NetBeans-ben konyvek néven! Készíts main metódust a Konyvek osztályban!

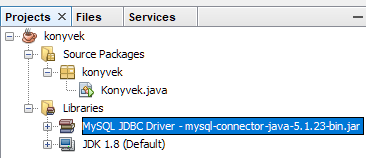
Kattins a jobb gombbal a Projects ablakban a Libraries elemre, majd válaszd az Add Library parancsot!



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



Ezzel hozzáadtad a drivert a projekthez:



## Lépések a programban

Az adatbázisokat kezelő programokban a következő fő lépéseket kell végrehajtani:

1. Kapcsolódás az adatbázishoz (Connection).  
   Ehhez kell egy JDBC URL:  
   MySQL: jdbc:mysql://localhost:3306/konyvek  
   SQLite: jdbc:sqlite:*fájl\_elérési\_útja\_és\_neve*  
   (*konyvek* az adatbázis neve)
2. Parancs objektum (Statement) létrehozása.
3. SQL lekérdezés végrehajtása.
4. Eredmények (Result Set) feldolgozása.

## Kapcsolódás az adatbázishoz

Az adatbázissal kapcsolatos műveleteket egy külön osztályban fogjuk elvégezni. Hozz létre egy új osztályt DB néven a konyvek csomagban!

Az adatbázishoz való kapcsolódáshoz szükséged lesz három stringre: a címre (lásd fent), a felhasználónévre és a jelszóra.

Készítsd el ezeket az osztály elejére:



A dbURL végén az adatbázis neve szerepel.

A kapcsolat objektumot a DriverManager osztály getConnection metódusával lehet megnyitni. A kapcsolat megnyitása közben SQLException kivétel jöhet létre, amelyet kezelni kell.

Például:

Connection kapcs = DriverManager.getConnection(dbUrl, user, pass);

## Parancs létrehozása

Egy parancs objektumot, amellyel majd SQL parancsokat futtathatunk, a kapcsolat createStatement metódusával hozhatunk létre.

Például:

Statement parancs = kapcs.createStatement();

## Lekérdezés futtatása

Lekérdezést egy Statement típusú objektum executeQuery metódusával futtathatunk.

Az eredmény egy ResultSet típusú objektumba kerül.

Például:   
ResultSet eredmeny = parancs.executeQuery("select \* from adatok");

Az eredmény több rekordból állhat, amelyeken egy while ciklussal tudunk végigmenni.

A kurzor induláskor az első rekord előtt áll. Az eredmény next metódusa megadja, hogy van-e még rekord, és ha igen, akkor a következőre lép.

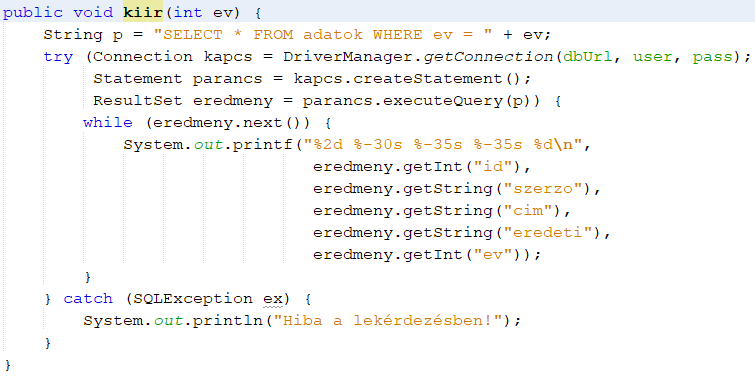
Egy rekord egy mezőjét az objektum getString, vagy getInt, vagy getXXX metódusával olvashatjuk ki, ahol XXX az elem típusa. Paraméterként meg kell adni a mező nevét (vagy sorszámát 1-től kezdve).

Ha például az eredmeny objektum minden rekordjából szeretnénk kiíratni a könyv címét, akkor ezt kell írnunk:

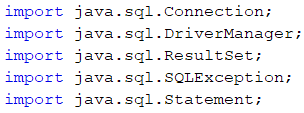
while (eredmeny.next())  
 System.out.println(eredmeny.getString("cim"));

## Tábla rekordjainak kiírása

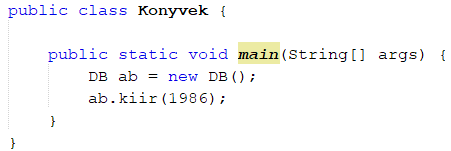
Készíts egy kiir nevű metódust a DB osztályban:



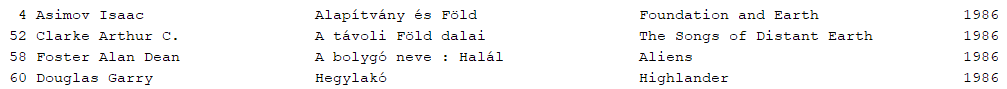
Végezd el a szükséges importálásokat, de ügyelj arra, hogy a java.sql csomagból importálj (ne a java.mysql-ből)!



Utána válts a Konyvek osztályra, és a main metódusban készíts egy új példányt a DB osztályból, majd hívd meg a kiir metódust 1986-os évszámmal!



Futtasd le a programot!



Próbáld ki többféle évszámmal!

## Előkészített parancs

Lekérdezések esetén van a Statement használatánál egy jobb megoldás is: a PreparedStatement (előkészített parancs).

Ennek az a lényege, hogy előre megadjuk a SELECT parancsot úgy, hogy a változó részek helyére ?-et írunk. A kérdőjeles helyeket 1-től kezdve számozzuk.

Például:

PreparedStatement ekp = kapcs.prepareStatement("SELECT \* FROM adatok WHERE ev = ?");

Ebben a példában az egyetlen kérdőjel az 1-es helyen van.

A futtatás előtt a a kérdőjeles helyek értékét meg kell adni a setString vagy a setInt, vagy a setDouble, … parancsokkal.

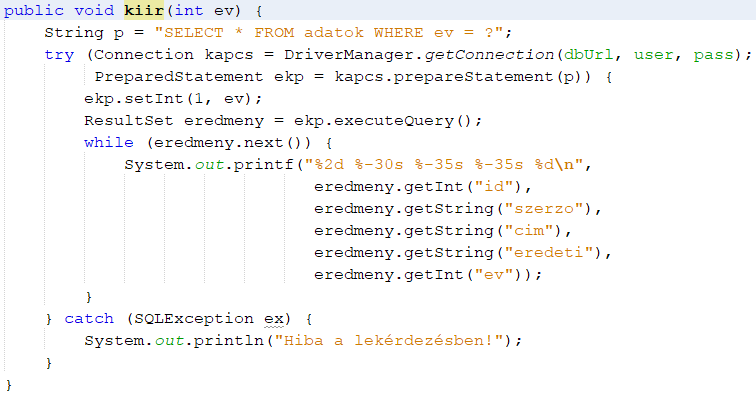
Például:

ekp.setInt(1, 1986);

Az előkészített parancs azért jobb, mert:

* Ellenőrzi az adatok típusát, és így véd a támadások ellen.
* Gyorsabban fut.

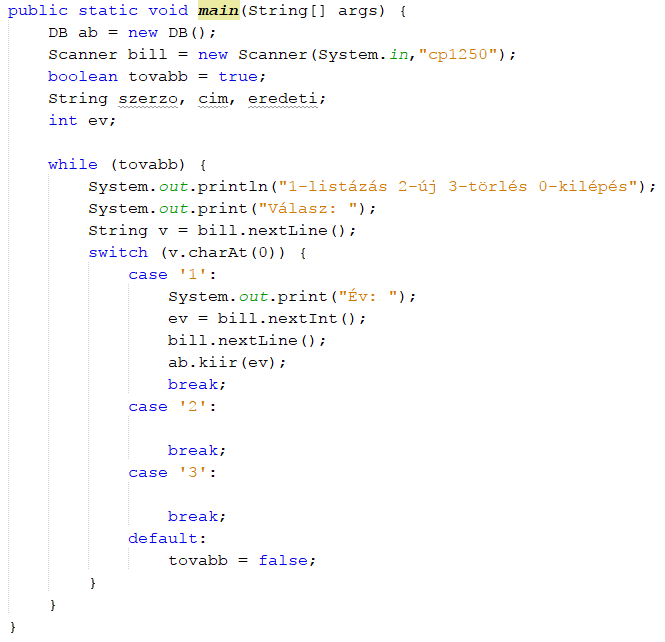
Alakítsd át a kiir metódus így:



Futtasd le! Uyganazt az eredményt kell kapnod, mint az előző változatnál.

## Főprogram átalakítása

A Konyvek osztály főprogramját szeretnénk több lehetőséggel bővíteni, ezért írd át így:



Figyeld meg a tovabb változó használatát! Miért nem a break utasítással szakítottuk meg a ciklust?

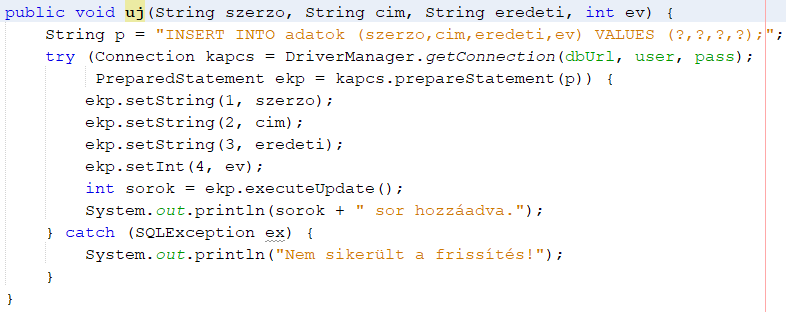
Miért van szükség az év beolvasása után a bill.nextLine(); sorra?

Futtasd a programot, és próbáld ki az 1-es menüpontot!

## Új könyv hozzáadása

Most az új könyv hozzáadásának lehetőségével bővítjük a programot.

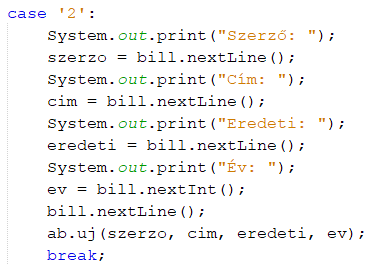
Ehhez először készítünk egy uj nevű metódust a DB osztályba:



Az SQL parancsot egy p stringben állítjuk össze.

Új rekord hozzáadását, rekord módosítását vagy törlését az executeUpdate metódussal lehet elvégezni. Ez a művelet által módosított sorok számát adja vissza. Végrehajtása közben SQLException kivétel keletkezhet.

Egészítsd ki a Konyvek osztály main metódusában az elágazás 2-es ágát így:



Ez a programrész beolvassa az új könyv adatait, majd meghívja az uj metódust ezekkel.

(Megjegyzés: A bemenő adatokat célszerű lenne ellenőrizni!)

Próbáld ki!

Nézd meg az adatbázisban is az új rekordot!

Mi történik, ha ő vagy ű betűt írsz be?

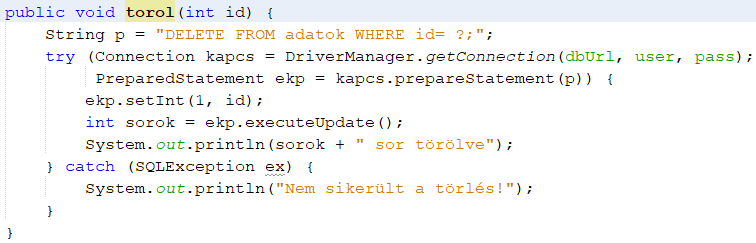
A hiba javításához módosítsd a dbURL változót így:



Próbáld ki!

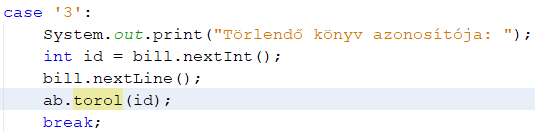
## Könyv törlése

Egy adott id-jú könyv törléséhez készíts egy torol metódust a DB osztályba:



Ez a metódus hasonlít az uj metódusra. Most is az executeUpdate metódust hívjuk meg, amely a törölt sorok számát adja vissza.

Egészítsd ki a Konyvek osztály main metódusában az elágazás 3-as ágát így:



Próbáld ki! Ellenőrizd az adatbázisban is a törlést! Mi történik, ha nem létező azonosítót adsz meg?

## SQLite adatbázis

Most átalakítjuk a programot úgy, hogy SQLite adatbázist használjon. (A WAMP szervert leállíthatod.)

Nyisd meg a kapott konyvek.db fájlt, és nézd meg a tartalmát! Másold a T:\-be!

Nyisd meg a <https://bitbucket.org/xerial/sqlite-jdbc/downloads> oldalt, és töltsd le a legújabb drivert!

Add hozzá a letöltött JAR fájlt a projekt könyvtáraihoz!  
(A MySQL drivert törölheted.)

Írd át az adatbázis URL címét így:



A user és a pass változók tetszőleges szöveget tartalmazhatnak, akár üreset is. A getConnection metódusban sem kell megadni őket.

Próbáld ki a programot!