

Összefoglalás II.

Adattábla listázása és módosítása

1. Adattábla létrehozás és feltöltése

Hozd létre saját adatbázisodban a JAVAKONYVEK nevű táblát, amely az alábbi szerkezettel rendelkezik:

#	Oszlop	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Extra
1	<u>azon</u>	varchar(8)	utf8_general_ci		Nem		
2	szerzo	varchar(40)	utf8_general_ci		Igen	NULL	
3	cim	varchar(70)	utf8_general_ci		Igen	NULL	
4	kiadasev	int(11)			Igen	NULL	
5	mennyiseg	int(11)			Igen	NULL	
6	ar	int(11)			Igen	NULL	
7	kolcsonozheto	varchar(4)	utf8_general_ci		Igen	NULL	

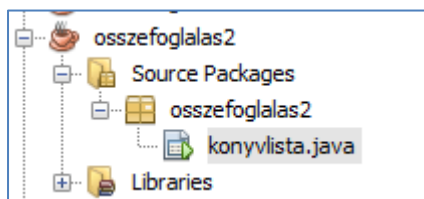
Az adattábla létrehozásához és feltöltéséhez használd a `KONYVEK.SQL` állományt!

2. Hozd létre a kapcsolatot a NetBEans-ben, amely a fenti táblát képes kezelni!

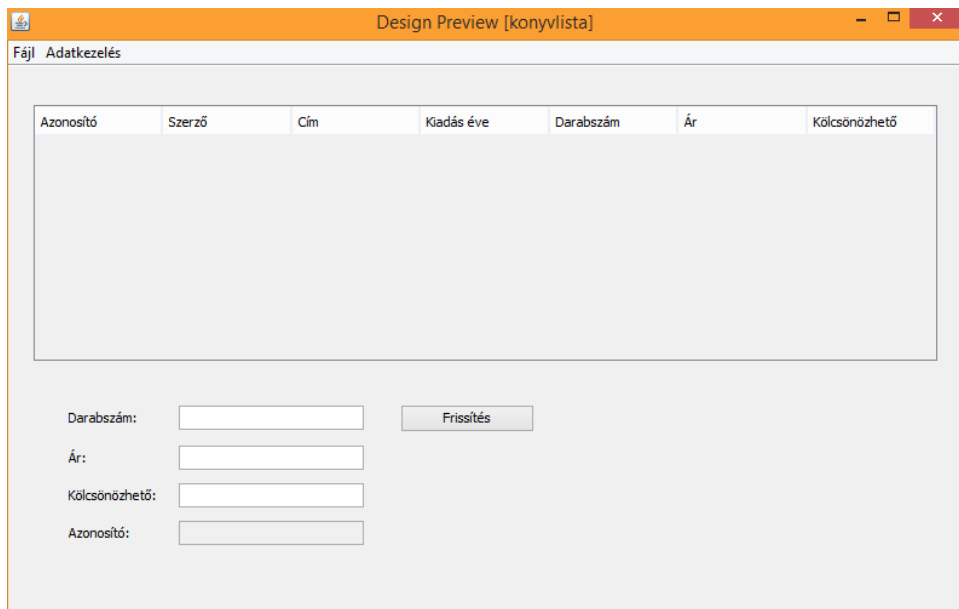
3. Készíts alkalmazást, amely az adatokat egy jTable segítségével listázza, valamint a táblázatban kiválasztott adatsor bizonyos elemeit módosítja.

A tábla betöltéséhez használj menüt!

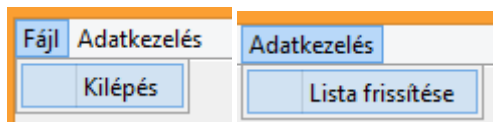
4. Hozz létre egy új projektet `osszefoglalas2` néven! A projekten belül hozz létre egy `JFrameForm`-ot `konyvlista` néven az alábbiak szerint:



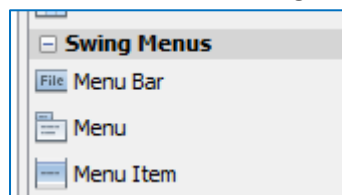
5. Készítsd el az alábbi formot, amely tartalmaz menüt és jTable vezérlőt! **(Részletek a következő oldalon!)**



Menü elkészítése (az alábbi 2-2 menüpontot hozd létre)



Menüt a NetBeans Swing Menus csoportjából kiválasztva tudunk létrehozni:



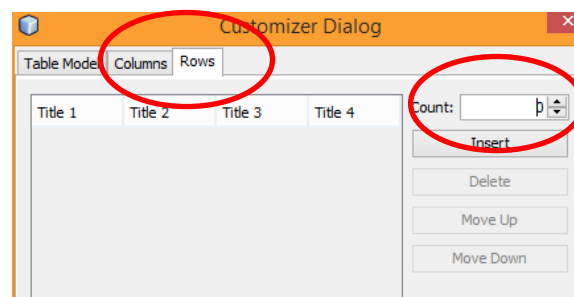
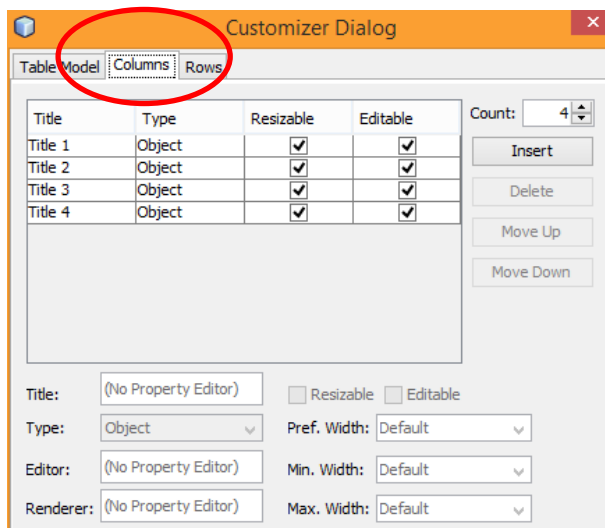
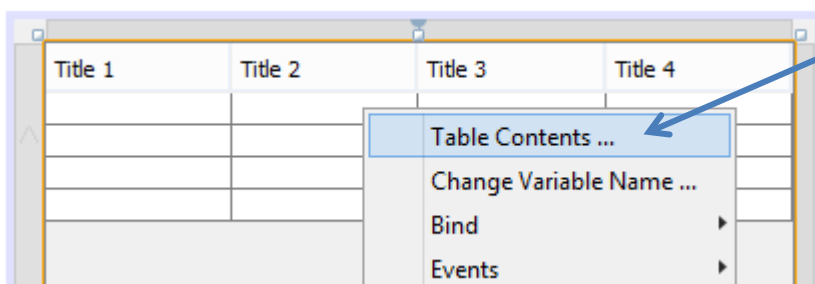
- Menu Bar hozza létre a menüt
- Menu hozzáadásával fő és almenüket készíthetünk

A menü elemek ugyanúgy használhatók, mint bármelyik vezérlő (két klikk → metódus létrehozás)

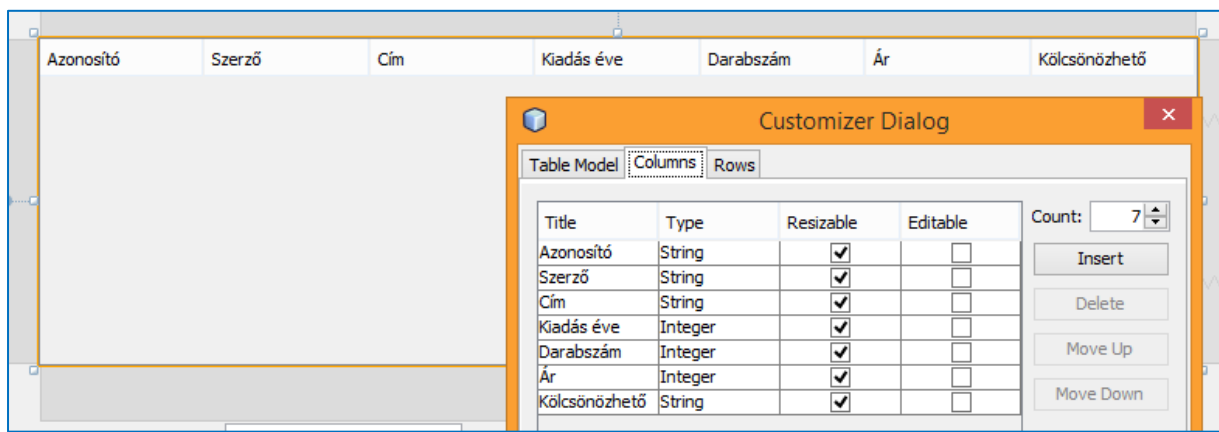
Táblázat készítése

Swing Controls → Table vezérlő beszáúrása

Ezzel egy 4 oszlopos táblázat készül el, ami tetszés szerint alakítható: **Jobb gomb → Table Contents**



Hozzunk létre egy 7 oszlopos táblázatot, és töröljük a sorokat a fenti párbeszédablakban! Ez lesz az adattábla megjelenítésére használt felület ☺



Készítsd el a táblázat alatt az alábbi jTextField-eket tartalmazó területet is!

Darabszám:	<input type="text"/>	Frissítés
Ár:	<input type="text"/>	
Kölcsönözhető:	<input type="text"/>	
Azonosító:	<input type="text"/>	

6. PROGRAMKÓD ELKÉSZÍTÉSE

A) Csatlakozási adatok rögzítése osztályszinten:

```
public class könyvlista extends javax.swing.JFrame {

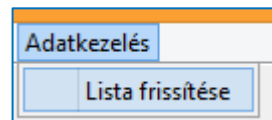
    //Csatlakozási adatok osztályszintű változóknban rögzítve
    String host = "jdbc:mysql://localhost:3306/konyvtar";
    String user = "root";
    String pass = "qwe123";

    public könyvlista() {
        initComponents();
    }
}
```

B) Az adattábla beolvasása és a táblázatba írása

A metódust a Lista frissítése menüponthoz rendeljük!

Ezzel a tábla feltöltődik az adatbázisból kiolvasott adatokkal, de előtte törli a korábban esetleg benne lévő adatsorokat.



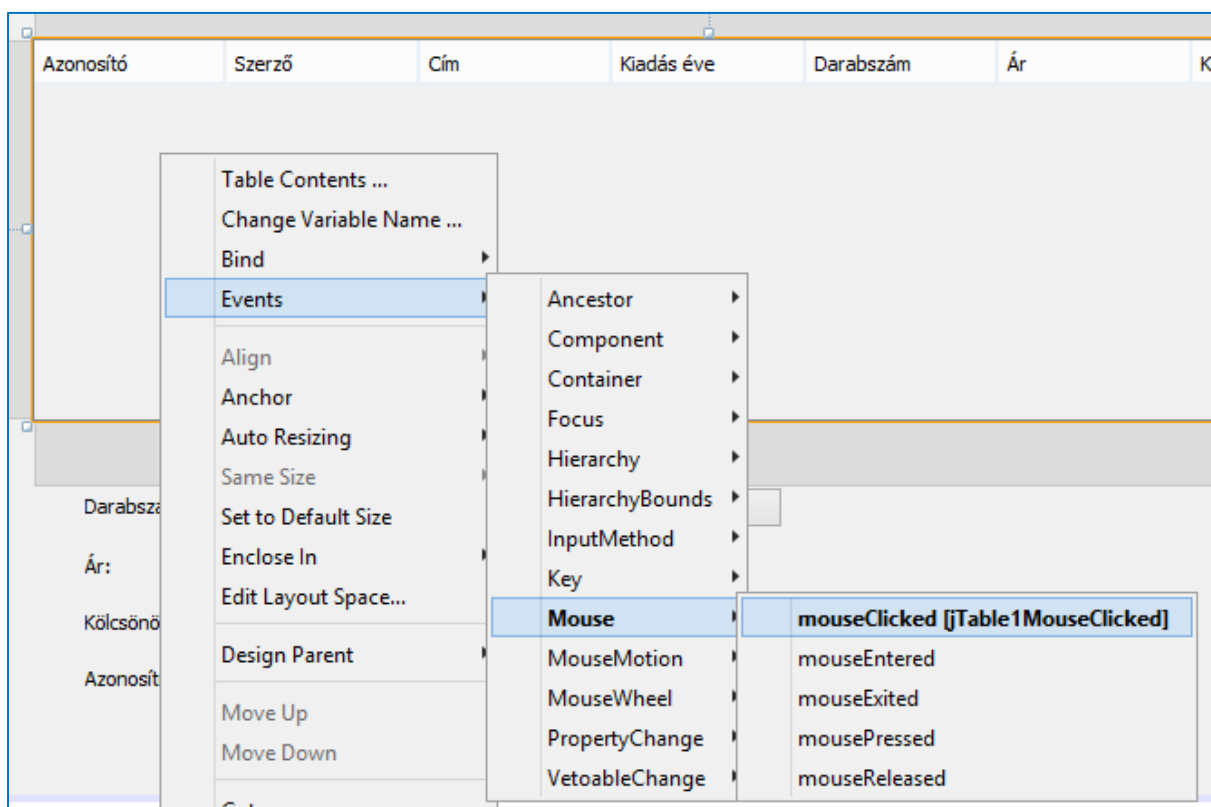
```
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
//csatlakozás, adatlekérés és a táblázat törlése és feltöltése
DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
int sorokszama = model.getRowCount();
//tábla esetleges sorainak törlése (a későbbi frissítés miatt)
for (int i = sorokszama - 1; i >= 0; i--) {
    model.removeRow(i);
}
try
{
    Connection con = DriverManager.getConnection(host,user,pass);
    String sql = "SELECT * FROM konyvek";
    Statement sta = con.createStatement();
    ResultSet rs = sta.executeQuery(sql);

    //kiolvassuk az adatokat rekordonként és beírjuk a táblába
    while(rs.next())
    {
        //aktuális rekord hozzáadása a táblához
        model.addRow(new Object[]{rs.getString("azon"),rs.getString("szerzo"),rs.getString("cim"),
            rs.getInt("kiadasev"),rs.getInt("mennyisege"),rs.getInt("ar"),rs.getString("kolcsonozheto"),});
    }
    sta.close();
    con.close();
}
catch (SQLException e)
{
    JOptionPane.showMessageDialog (null, "Hiba a lekérdezés során!");
}
}
```

C) Ha a táblában kijelölünk egy sort, akkor töltődjön be az alul lévő TextField-ekbe!

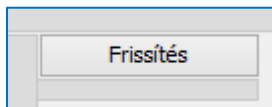
Ehhez a táblázat egy metódusát használjuk:



A forráskód:

```
private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {  
    //a táblázatban kiválasztott sor száma:  
    int sor = jTable1.getSelectedRow();  
  
    //az id, darabszám, ár és a kölcsönözhető értékének lekérdezése  
    String konyvID = (jTable1.getModel().getValueAt(sor, 0).toString());  
    int darabszam = Integer.parseInt(jTable1.getModel().getValueAt(sor, 4).toString());  
    int ar = Integer.parseInt(jTable1.getModel().getValueAt(sor, 5).toString());  
    String kolcsonozheto = (jTable1.getModel().getValueAt(sor, 6).toString());  
  
    //a lekérdezett adatok kiírása a textField-ekbe  
    jTextField1.setText(Integer.toString(darabszam));  
    jTextField2.setText(Integer.toString(ar));  
    jTextField3.setText(kolcsonozheto);  
    jTextField4.setText(konyvID);  
}
```

D) Az adatok frissítése a FRISSÍTÉS gomb megnyomásával



A TextField-ekben lévő adatok visszaírása az adatbázisba:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // Adatsor módosítása  
    try  
    {  
        Connection con = DriverManager.getConnection(host,user,pass);  
        String sql = "UPDATE konyvek SET mennyiseg = "+jTextField1.getText()+", "  
            + "ar = "+jTextField2.getText()+", "  
            + "kolcsonozheto = '"+jTextField3.getText()+"' "  
            + "WHERE azon = '"+jTextField4.getText()+"'";  
        Statement sta = con.createStatement();  
        sta.executeUpdate(sql);  
        sta.close();  
        con.close();  
        JOptionPane.showMessageDialog (null, "Frissítse a táblát!");  
    }  
    catch (SQLException e)  
    {  
        JOptionPane.showMessageDialog (null, "Hiba az írás során!");  
    }  
}
```

E) A Fájl – Kilépés metódusa

```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```

