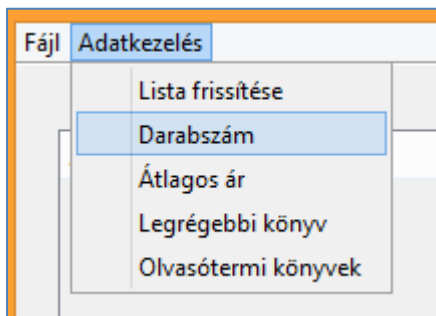


Összefoglalás III.

Egyszerű lekérdezések használata

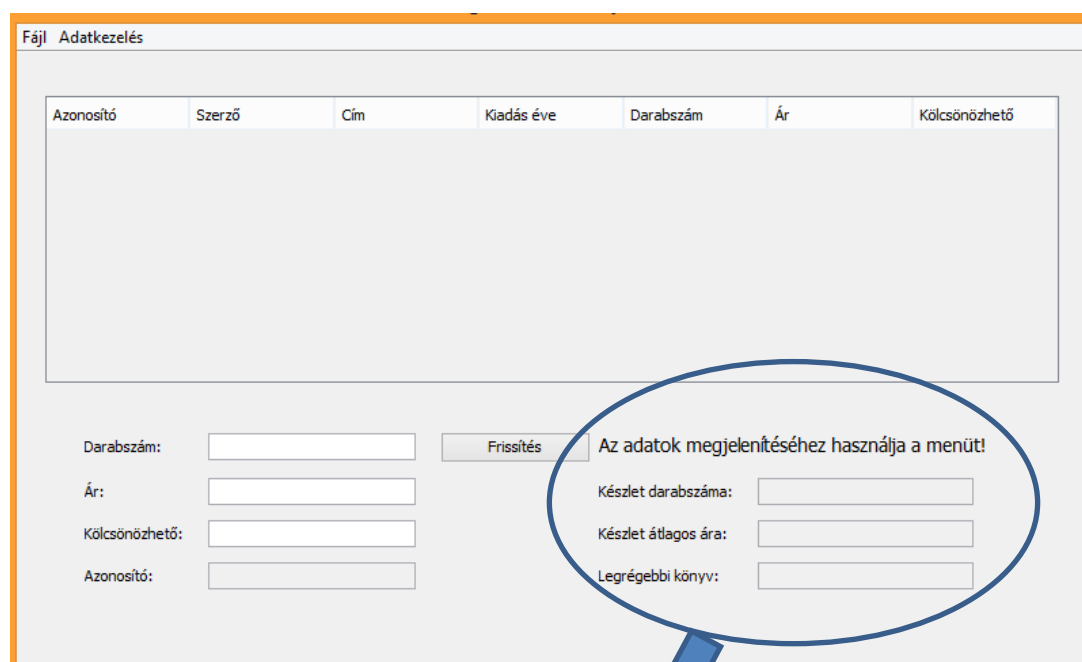
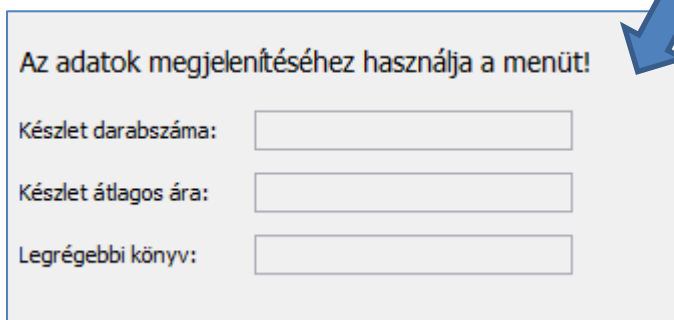
Bővítsük az osszefoglalas2 projektünket az alábbiakkal!

1. Menü kiegészítése – Hozzunk létre újabb négy menüpontot az ábra szerint:



2. Form kiegészítése

Hozzunk létre a könyvlista formon textField-eket az adatok megjelenítéséhez, a szerkeszthetőségüket tiltsuk le, hogy a felhasználó ne tudjon beleírni. (Mert minek is...)

A screenshot of a software application's data management form. The form has a title bar with 'Fájl' and 'Adatkezelés'. Below the title bar is a table with columns: 'Azonosító', 'Szerző', 'Cím', 'Kiadás éve', 'Darabszám', 'Ár', and 'Kölcsönözhető'. Below the table are input fields for 'Darabszám:', 'Ár:', 'Kölcsönözhető:', and 'Azonosító:'. A 'Frissítés' button is located next to the 'Darabszám' input field. A blue oval highlights a section with the text 'Az adatok megjelenítéséhez használja a menüt!' and three disabled text fields: 'Készlet darabszáma:', 'Készlet átlagos ára:', and 'Legrégebbi könyv:'.A close-up screenshot of the highlighted section from the previous image. It shows the text 'Az adatok megjelenítéséhez használja a menüt!' and three disabled text fields: 'Készlet darabszáma:', 'Készlet átlagos ára:', and 'Legrégebbi könyv:'.

3. A darabszám megállapítása (a menüponthoz rendelt metódus)

```
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {  
    try  
    {  
        Connection con = DriverManager.getConnection(host,user,pass);  
        //a mennyiség mező értékeinek összege  
        String sql = "SELECT sum(mennyiseg) as osszeg FROM konyvek";  
        Statement sta = con.createStatement();  
        ResultSet rs = sta.executeQuery(sql);  
        rs.next();  
        jTextField5.setText(rs.getString("osszeg"));  
        sta.close();  
        con.close();  
    }  
    catch (SQLException e)  
    {  
        JOptionPane.showMessageDialog (null, "Hiba a lekérdezés során!");  
    }  
}
```

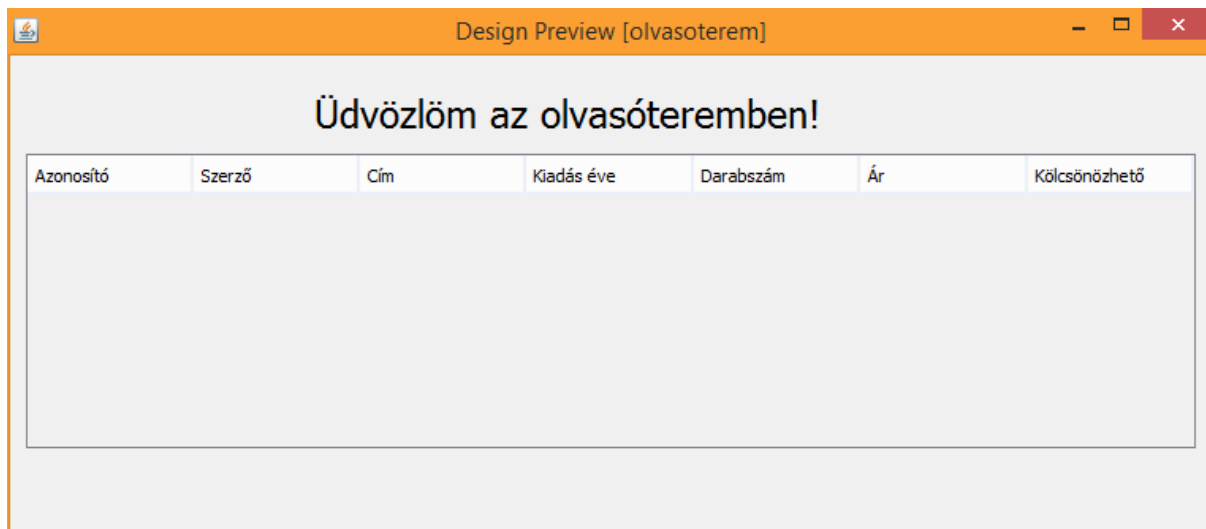
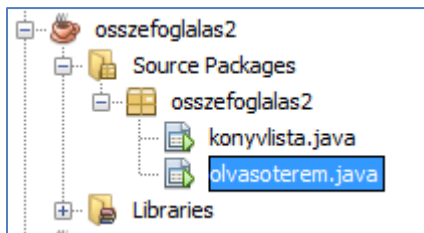
4. Az átlagos ár meghatározása (a menüponthoz rendelt metódus)

```
private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) {  
    try  
    {  
        Connection con = DriverManager.getConnection(host,user,pass);  
        //a mennyiség mező értékeinek összege  
        String sql = "SELECT avg(ar) as atlag FROM konyvek";  
        Statement sta = con.createStatement();  
        ResultSet rs = sta.executeQuery(sql);  
        rs.next();  
        Double atlag = rs.getDouble("atlag");  
        jTextField6.setText(Double.toString(atlag));  
        sta.close();  
        con.close();  
    }  
    catch (SQLException e)  
    {  
        JOptionPane.showMessageDialog (null, "Hiba a lekérdezés során!");  
    }  
}
```

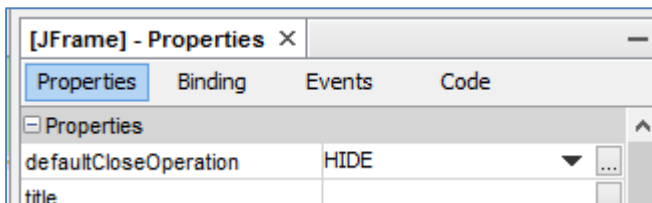
5. A fentiek alapján készítsünk metódust a legrégibbi könyv megállapítására!



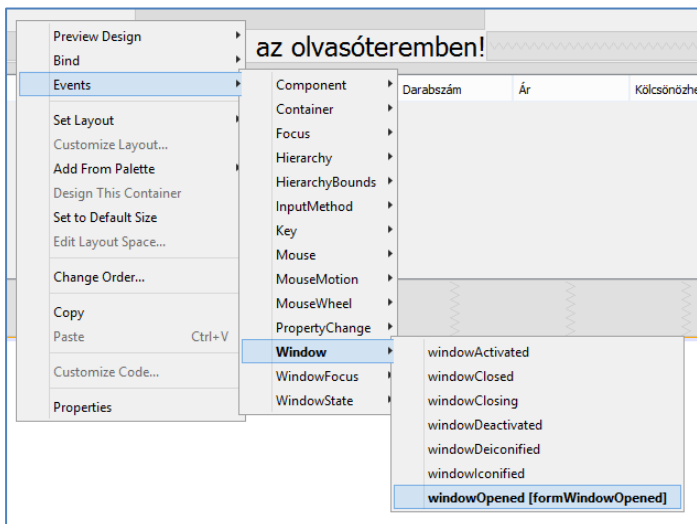
6. Az utolsó menüpont eredménye új formon jelenjen meg, ehhez hozzuk létre az **olvasoterem.java**-t, ami szintén JFrameForm:



- a táblázat UGYANAZ, mint az előző formon, átmásolható ☺
- ha nem szeretnénk, hogy a form bezárásával az egész program bezáruljon, akkor a form Properties-ben:



- a metódust az ablak betöltődésekor hajtsuk végre!



- ez kell hozzá, ugye...

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

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

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

- és a metódus végre (majdnem ugyanaz, mint a másik formon):

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {
    //Olvasótermi könyvek megjelenítése

    //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 WHERE kolcsonozheto LIKE 'Nem\|r'";
        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("mennyiseg"),rs.getInt("ar"),rs.getString("kolcsonozheto"),});
        }
        sta.close();
        con.close();
    }
    catch (SQLException e)
    {
        JOptionPane.showMessageDialog (null, "Hiba a lekérdezés során!");
    }
}
```

Természetesen a lekérdezés WHERE feltétele a megoldás kulcsa!

- a könyvlista form megfelelő menüjének kódja:

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    olvasoterem ot = new olvasoterem();
    ot.setVisible(true);
}
```