

Fájlkezelés – Swing Form – jTable használata

Hozzuk létre az alábbi formot, melyen új elem a jTable!

Név	Tantárgy	Jegy	
Bárdi Ágnes	történelem	4	
Bátai Attila	matematika	4	
Bellér Anita	magyar	3	
Berecz Andrea	angol	5	
Bognár Balázs	magyar	5	
Ceskel Katalin	magyar	5	
Csizmadia Brigitta	fizika	4	
Domonkos Tímea	matematika	4	

Kiválasztott tanuló adatai

Tantárgyi átlag megállapítása

történelem

3.0

A **jTable** a beszúrás után tartalmaz sorokat és oszlopokat, utóbbit alapértelmezett azonosító nevekkal.

JOBB GOMB – TABLE CONTENTS menüpont alatt megadható a sorok száma (állítsuk 0-ra) és az oszlopok neve és egyéb tulajdonságai.

A formon

- A nyomógomb hatására töltődjenek be a fakt.txt állományból az adatok Class és Lista használatával
- Szintén a nyomógomb eseményének részeként az adatok jelenjenek meg a jTable-ben!
- Ha a jTable-ben kiválasztunk egy sort, a tanuló adatai jelenjenek meg a táblázat alatti textfield-ekben!
- A conboBox-ba gyűjtsük ki a tantárgykat, de mindegyiket csak egyszer jelenítsük meg!
- A conboBox-ban kiválasztott tantárgyhoz tartozó jegyek átlagát adjuk meg az alatta lévő textfield-ben!

Az osztály:

```
12 public class FaktClass {
13     private String nev;
14     private String tantargy;
15     private byte jegy;
16
17     public FaktClass(String sor) {
18         String[] d = sor.split(";");
19         this.nev = d[0];
20         this.tantargy = d[1];
21         this.jegy = Byte.parseByte(d[2]);
22     }
23
24     public String getNev() {
25         return nev;
26     }
27
28     public String getTantargy() {
29         return tantargy;
30     }
31
32     public byte getJegy() {
33         return jegy;
34     }
35 }
```

Az osztály végzi a fájlból beolvasott adatsorok darabolását a konstruktorban.

A **form forráskódjában** osztályszinten hozzuk létre az adattároláshoz szükséges listát és a jTable kezeléséhez szükséges példányt:

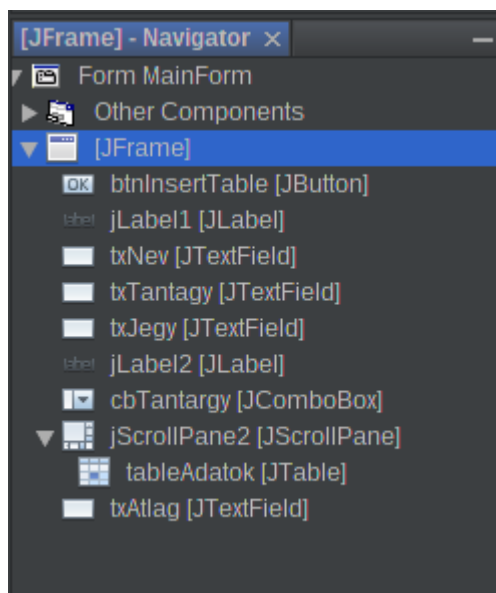
```
public class MainForm extends javax.swing.JFrame {

    //Az adatok tárolására szolgáló az osztálynak megfelelő típusú lista
    ArrayList<FaktClass> lista = new ArrayList<>();
    DefaultTableModel model;

    ....

    public MainForm() {
        initComponents();
        this.model = (DefaultTableModel) tableAdatok.getModel();
    }
}
```

A jTable-t **tableAdatok**-nak neveztem el...



A nyomógomb hatására olvassuk be az adatokat és töltjük fel a táblázatot!

```
private void btnInsertTableActionPerformed(java.awt.event.ActionEvent evt)
.....
    //adatok beolvasása
    try
    {
        FileReader fajl = new FileReader("fakt.txt");
        BufferedReader br = new BufferedReader(fajl);
        String sor= "";
        while((sor = br.readLine()) != null)
        {
            FaktClass f = new FaktClass(sor);
            lista.add(f);
        }
        System.out.println("Beolvasás OK!");
        System.out.println("A lista elemszáma: " + lista.size());
    }
    catch (IOException e)
    {
        JOptionPane.showMessageDialog(null, "Hiba a beolvasáskor!");
        System.out.println("HIBA: " + e);
    }
.....
    insertTable();
    btnInsertTable.setEnabled(false);
    insertComboBox();
} //GEN-LAST:event_btnInsertTableActionPerformed
```

A táblába való beszúrás kódja:

```
private void insertTable()
{
    //adattábla kiürítése
    int sorokSzama = model.getRowCount();
    for (int i = sorokSzama; i > 0; i--) {
        model.removeRow(i);
    }
.....
    //adattábla feltöltése a listából
    for (FaktClass f : lista) {
        model.addRow(new Object[]{f.getNev(), f.getTantargy(), f.getJegy()});
        System.out.println(f.getNev());
    }
}
.....
```

A comboBox feltöltése a tantárgyakkal

```
private void insertComboBox()
{
    //tantárgyak kigyűjtése
    ArrayList<String> tantargyak = new ArrayList<>();
    for (FaktClass f : lista)
    {
        if(!tantargyak.contains(f.getTantargy()))
        {
            tantargyak.add(f.getTantargy());
        }
    }.....
    //tantárgyak beírása a comboboxba.....
    for (String t : tantargyak) {
        cbTantargy.addItem(t);
    }
    .....
}
```

A combobox-ból kiolvasott tantárgy jegyeinek átlagolása

```
private void cbTantargyActionPerformed(java.awt.event.ActionEvent evt) { //GE
    String valTantargy = cbTantargy.getSelectedItem().toString();
    int osszeg = 0;
    int tantargyDB = 0;
    for (FaktClass f : lista) {
        if(f.getTantargy().equals(valTantargy))
        {
            osszeg += f.getJegy();
            tantargyDB++;
        }
    }
    double atlag = osszeg / tantargyDB;
    txAtlag.setText(Double.toString(atlag));
} //GEN-LAST:event_cbTantargyActionPerformed
```

A comboBox eseményét két kattintással érjük el...

A táblázatban kiválasztott sor adatait az alábbi kóddal nyerük ki:

```
private void tableAdatokMouseClicked(java.awt.event.MouseEvent evt) { //GEN-FIRST:event_tab
    // a kiválasztott sor adatainak felhasználása
    int sor = tableAdatok.getSelectedRow();
    .....
    String nev = tableAdatok.getModel().getValueAt(sor, 0).toString();
    String tantargy = tableAdatok.getModel().getValueAt(sor, 1).toString();
    int jegy = Integer.parseInt(tableAdatok.getModel().getValueAt(sor, 2).toString());
    txNev.setText(nev);
    txTantargy.setText(tantargy);
    txJegy.setText(Integer.toString(jegy));
} //GEN-LAST:event_tableAdatokMouseClicked
```