## ObservableList

Olyan lista, amelynek elemeit meg lehet jeleníteni egy vezérlőben (ComboBox, ListView, …).

A vezérlő getItems metódusával kérdezhető le.

A JCF bővítése. Az ArryList-től abban különbözik, hogy figyeli az elemek változásait.

Az ArrayList metódusai használhatók az ObservableList esetén is: add, AddAll, get, set, remove, clear, contains, isEmpty, size.

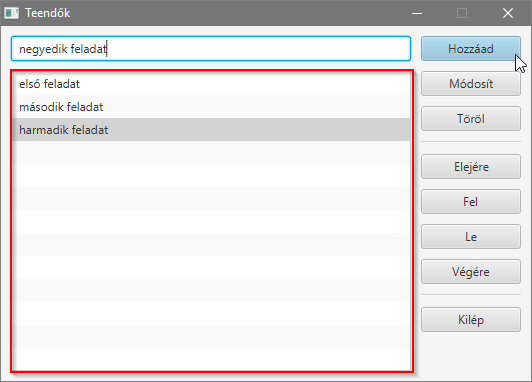
## ListView

Objektumokat listáz. Ha az objektum nem String típusú, akkor a toString() metódusával átalakítja.

Alapértelmezésben egyszerre egy elem jelölhető ki, de ez módosítható.

Egy elem kijelölése: kattintás, nyilak, Home, End, PgUp, PgDn.

Szükség esetén automatikusan megjelennek a görgetősávok.



Konstruktorok:

|  |  |
| --- | --- |
| ListView<T>() | üres lista létrehozása |
| ListView<T>(ObservableList<T> items) | új lista a megadott elemekkel |

Metódusok:

|  |  |
| --- | --- |
| ObservableList<T> **getItems**() | elemek listájának lekérdezése |
| void **setItems**(ObservableList<T> items) | elemek listájának beállítása |

Elemek kezeléséhez le kell kérni az elemek listáját:

ListView lista = new ListView();  
lista.**getItems().addAll**("sonka", "szalámi", "sajt"); // több elem hozzáadása  
lista.**getItems().add**("gomba"); // egy elem hozzáadása  
lista.**getItems().get**(0); // 0-dik elem tartalma  
lista.**getItems().set**(0,"hús"); // 0-dik elem módosítása  
lista.**getItems().remove**(0); // 0-dik elem törlése

A kijelöléssel kapcsolatos műveletekhez le kell kérni a kijelölésmodellt:

lista.**getSelectionModel().select(0)**; // 0-dik lem kijelölése  
// kijelölt elem indexe (ha nincs kijelölve: -1):  
int i = lista.**getSelectionModel().getSelectedIndex()**;   
// kijelölt elem tartalma (ha nincs kijelölve: null):  
String s = lista.**getSelectionModel().getSelectedItem()**;

Eseménykezelő, amely a kijelölés változásakor fut le:

lista.**getSelectionModel().selectedItemProperty().addListener**(  
 (v, regi, uj) -> {  
 // *utasítások*

});