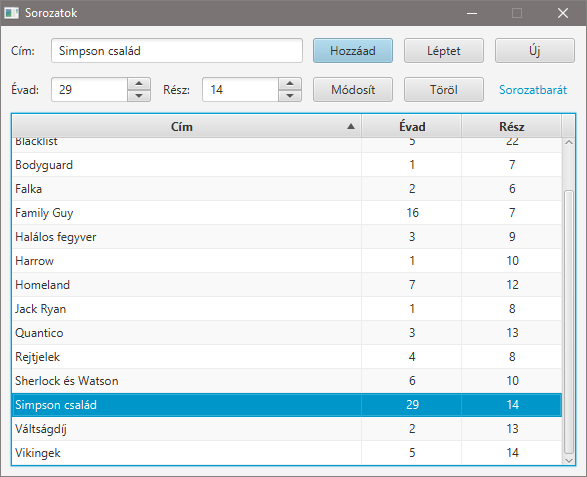
# Sorozatok

## Feladat

Ebben a leckében egy olyan programot készítünk, amellyel az általunk nézett sorozatokat tarthatjuk nyilván:



Új sorozatot a Hozzáad gombbal adhatunk a táblázat végéhez.

Ha a táblázatban kijelölünk egy sort, tartalma beíródik a felső részre. Ha megváltoztatjuk, majd megnyomjuk a Módosít gombot, a kijelölt sor helyett beíródik a módosított.

A Léptet gombbal egy részt hozzáadhatunk a kijelölt sorozathoz.

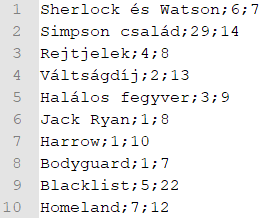
Az Új gombbal tudjuk alaphelyzetbe hozni a felső részt.

Ha már nem nézünk egy sorozatot, kijelöljük, majd a Töröl gombbal töröljük a táblázatból.

A sorokat rendezhetjük az oszlopok fejlécére kattintva.

Minden módosítás után a program elmenti a lista tartalmát a *sorozatok.txt* nevű fájlba.

A fájl formátuma: cím;évad;rész  
Például:



Ezt a fájlt akár a Jegyzettömbbel is szerkeszthetnénk, de sokkal kényelmesebb egy program használata.

Induláskor a program megpróbálja betölteni a legutóbb mentett táblázatot. Ha ez nem sikerül, akkor üres lesz a táblázat.

A hivatkozásra kattintva megnyílik a hivatkozott oldal (SorozatBarát).

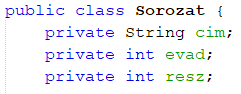
Kezdj egy új JavaFX FXML projektet *sorozatok* néven!

## A Sorozat osztály

Egy-egy sorozat adatait a Sorozat osztály egy-egy példányában fogjuk tárolni. Ez fog megjelenni a táblázat egy-egy sorában.

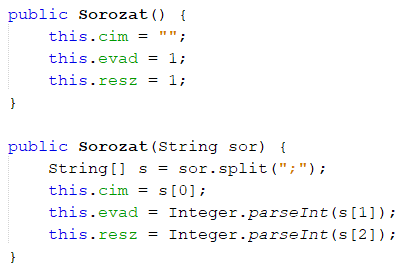
A Sorozat osztály példányainak egy-egy mezője kerül az oszlopokba (cím, evad, resz). Minden mezőhöz készíteni kell egy getter és egy setter metódust. Például ha a mező neve cim, akkor kell egy getCim és egy setCim metódus. Az elnevezésektől nem szabad eltérni!

Minden sorozatról három adatot tárolunk: a címét, valamint az évadot és a részt, amelynél járunk:

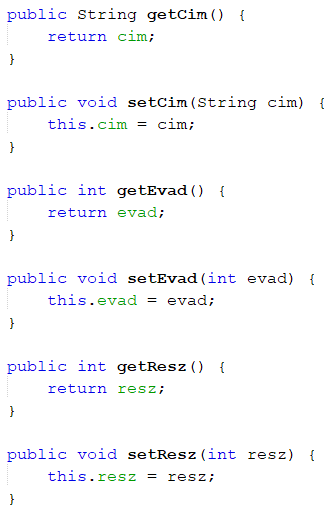


Kétféle konstruktort készítünk. Az alapértelmezett konstruktor nem kap paramétert, és alapértékeket állít be, amelyeket utána módosíthatunk. Ezt új sorozat megadásánál és sorozat módosításánál használjuk.

A másik konstruktor egy beolvasott sort fog kapni, és ebből állítja be az értékeket. Ezt a fájlból történő beolvasásnál használjuk.

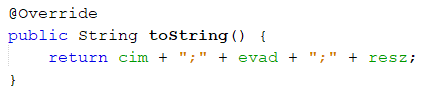


Ezután jönnek a getter és setter metódusok, amelyeket a NetBeans automatikusan elkészít (Alt+Insert, Getter and setter…):

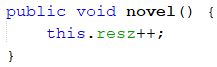


A táblázat elkészítéséhez fontos, hogy pontosan ezeket a metódus neveket használjuk!

A fájlba íráshoz felülírjuk az alapértelmezett toString metódust:



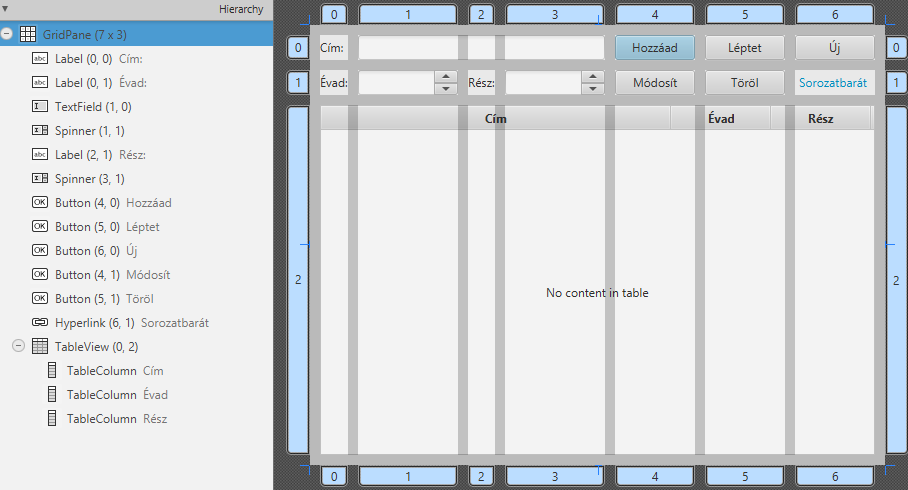
És végül szükségünk lesz egy novel metódusra, amely a részek számát növeli eggyel:



## Az ablak elkészítése

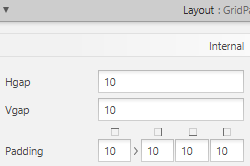
Hogyan rendeznéd el az elemeket az ablakban?

Az elrendezéshez egy GridPane tárolót használunk. Készítsd el a program ablakát a Scene Builderben az alábbi minta alapján:



Alakítsd át a GridPane-t 7 x 3-asra!

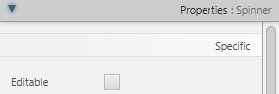
A GridPane térközei mindkét irányban 10 pontosak legyenek, és körben legyen 10 pontos padding:



A szövegmező azonosítója legyen txtCim, és három oszlopot foglaljon el!

Az évadot és a részt Spinner-ek segítségével adjuk meg. A Spinner-ek preferált szélessége legyen 100 pont, azonosítóik: spEvad, spResz!

A Spinner-ek szerkeszthetősége maradjon kikapcsolva, így csak a nyilakkal lehet léptetni őket:



A gombok preferált szélessége legyen 80 képpont! A Hozzáad gomb legyen default, az Új pedig cancel gomb!

A gombok megnyomásakor a következő metódusok fussanak le:   
hozzaad, leptet, ujsorozat, modosit, torol!

A Hyperlinkre (Sorozatbarát) kattintáskor a web metódus fusson le!

A TableView preferált magassága 350 pont, preferált szélessége számított, azonosítója pedig tblSorozatok legyen! A TableView 7 oszlopot foglaljon el a GridPane-ben (Column span: 7)!

A TableView elemre kell ráhúzni az oszlopokat (TableColumn).

A Cím oszlop azonosítója oCim, preferált szélessége 350 pont,   
az Évad oszlop azonosítója oEvad, preferált szélessége 100 pont,   
a Rész oszlop azonosítója oResz, preferált szélessége 100 pont legyen!

Az Évad és a Rész oszlopra állíts be középre igazítást stílus segítségével:



A GridPane szélessége, magassága, oszlopainak szélessége és sorainak magassága is legyen számított!

Add meg a controller osztály nevét a bal alsó sarokban (Document, Controller)!

Ezután nyisd meg a View, Show Sample Controller Skeleton paranccsal a controllerbe másolandó kódot, és másold a vágólapra a szükséges deklarációkat!

Mentsd a jelenetet, és válts vissza a NetBeans-re!

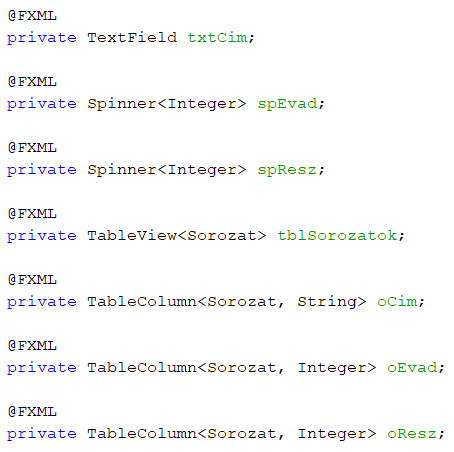
Állítsd be az ablak címét (Sorozatok), és tiltsd le az átméretezését!

Nézd át a generált FXML dokumentumot!

## Controller osztály

Másold be a kódot a vágólapról a controller osztályba! Végezd el a szükséges importálásokat!

Add meg a típusokat a szükséges helyeken:

  
A Spinner-ekben egész számokat fogunk használni.

A TableView sorai Sorozat típusúak lesznek, az oszlopoknál pedig meg kell adni a sor és az oszlop típusát is.

Töröld az eseménykezelők ActionEvent paramétereit!

Futtasd a programot!

## Spinner-ek

A Spinner-ek értéke a Nyíl billentyűkkel vagy nyilakra kattintással léptethető.

Például:



Az Editable tulajdonság határozza meg, hogy be lehessen-e írni más értéket, vagy sem. Mi most kikapcsoltuk a szerkeszthetőséget.

Egy Spinner inicializáláskor meg kell adni azt a metódust, amely a léptetéskor a következő értéket generálja.

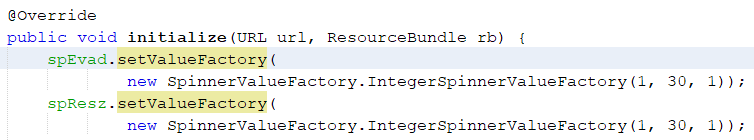
Ha például egész számokat szeretnénk léptetni 1 és 30 között, 10 kezdőértékkel, akkor ezt kell írni az initialize metódusba (lehet egy sorba is):

spResz.setValueFactory(  
 new SpinnerValueFactory.IntegerSpinnerValueFactory(1, 30, 10));

Ezután így lehet a spinner értékét lekérdezni, illetve beállítani például 15-re:

Integer r = spResz.getValue();  
spResz.getValueFactory().setValue(15);

Írd most ezt az initialize metódusba:



Az egész értékek 1 és 30 között változhatnak, és 1 lesz az induláskor beállított érték.

Próbáld ki!

## TableView

A TableView objektum egy táblázatot jelenít meg.

Az oszlopokat TableColumn típusú objektumok alkotják, amelyeket egy ObservableList tárol.

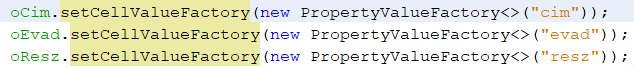
A sorok egy-egy objektumot jelenítenek meg (pl. a Sorozat osztály egy-egy példányát). Ennek egy-egy mezője kerül az oszlopokba (pl. cím, evad, resz). A táblázat sorait is egy ObservableList tárolja.

Alapértelmezésben egyszerre csak egy sor jelölhető ki, de ez módosítható.   
Egy sor kijelölése: kattintás, nyilak, Home, End, PgUp, PgDn.

Szükség esetén automatikusan megjelennek a görgetősávok.

Induláskor azt is meg kell adni, hogy a táblázat egyes oszlopaiban a Sorozat osztály melyik mezőjének értékét kell megjeleníteni.

Add hozzá a következőket az initialize metódushoz:



Ennek hatására az oCim oszlopban a Sorozat osztály cim mezője, az oEvad oszlopban az evad mezője, az oResz oszlopban pedig a resz mezője jelenik meg.

A sorok kezeléséhez le kell kérni az elemek listáját. Például:

// új sorozat hozzáadása  
tablazat.**getItems().add**(new Sorozat());  
// i-dik sor módosítása   
tablazat.**getItems().set**(i, new Sorozat());   
// i-dik sor törlése   
tablazat.**getItems().remove**(i);  
// sorok száma  
tablazat.**getItems().size(**);

A kijelöléssel kapcsolatos műveletekhez le kell kérni a kijelölésmodellt (mint a listánál):

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

Eseménykezelő, amely a kijelölés változásakor fut le:

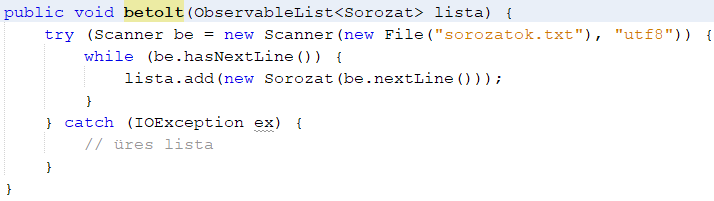
tablazat.**getSelectionModel().selectedItemProperty().addListener**(  
 (p, regi, uj) -> {  
 // *utasítások*

});

## Betöltés

A program indításakor be kell tölteni a *sorozatok.txt* fájlból az adatokat. Ha nem tudja megnyitni a fájlt, akkor üres táblázattal induljon a program!

Készítsd el először a controller osztályban a betolt metódust, amely paraméterként kap egy listát, és ehhez fűzi hozzá az adatokat:



A lista paraméter helyett használhattuk volna a tblSorozatok táblázatot, de így általánosabb a metódus: bármelyik vezérlővel használható, amelynek tartalma egy ObservableList-ben tárolódik.

Figyeld meg, hogy a metódus hiba esetén nem csinál semmit!

Ezután hívd meg a betolt metódust az initialize metódusból!



Másold a *sorozatok.txt* fájlt a projekt mappájába, majd indítsd el a programot!

Ha betöltődtek az adatok a táblázatba, próbáld ki a következőket:

* oszlopszélesség állítása húzással,
* oszlopok sorrendjének módosítása húzással,
* oszlop rendezése a fejlécre kattintással.

Hogyan lehet több oszlop szerinti rendezést megadni? (Shift+kattintás)

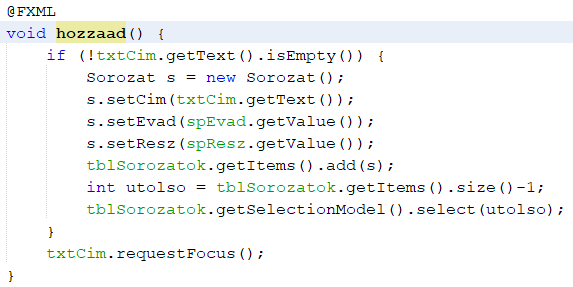
## Sorozatok kezelése

A Hozzáad gomb megnyomásakor a program megvizsgálja a címet. Ha a cím nem üres, létrehoz egy új példányt a Sorozat osztályból, és ebben beállítja a szövegmezőbe írt címet, valamint a spinnerekben beállított évadot és részt. Végül hozzáad egy sort a táblázathoz a sorozat adataival.

Mivel az új sor a táblázat végére kerül, kijelöli az utolsó sort. (A rendezés nem frissül automatikusan.)

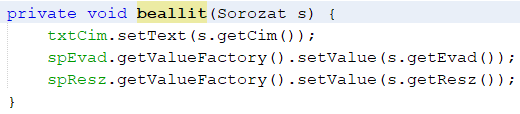
Végül a kurzort a szövegmezőbe viszi a requestFocus metódussal.

Készítsd el a hozzaad metódust:

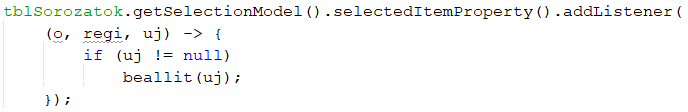


Próbáld ki, adj meg egy új sorozatot!

Azt szeretnénk, hogy a táblázatban kijelölt sor adatai megjelenjenek a felső részen. Ehhez először egy olyan metódust készítünk, amely egy Sorozat objektumot kap, és beállítja ennek adatait a felső részen:



Ezután egy eseménykezelőt készítenünk az initialize metódus végére, amely a kijelölt sor változásakor bemásolja az új elem adatait a beallit metódus segítségével:



Az uj paraméter értéke akkor lesz null, ha egy sor sincs kijelölve a táblázatban.

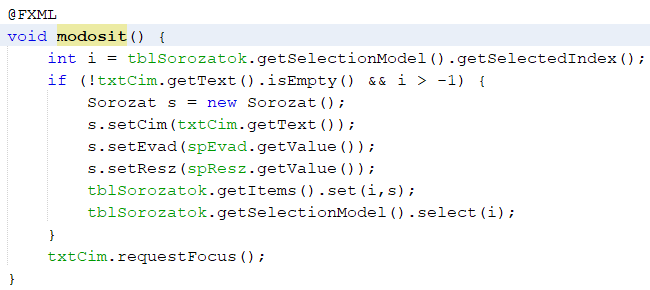
Próbáld ki, jelölj ki sorokat a táblázatban!

A kijelölt sor módosításához először meghatározzuk a listában kijelölt elem indexét. Ha nincs semmi kijelölve, akkor ez -1, egyébként ennél nagyobb.

Ha nem üres a cím, és ki van jelölve sor, akkor elvégezzük a módosítást, és kijelöljük a módosított sort.

Utána a kurzort visszavisszük a szövegmezőre.

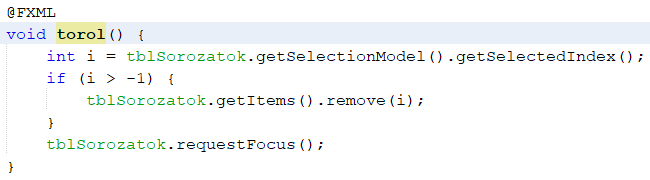
Készítsd el a modosit metódust:



Próbáld ki, módosítsd az egyik sorozat adatait!

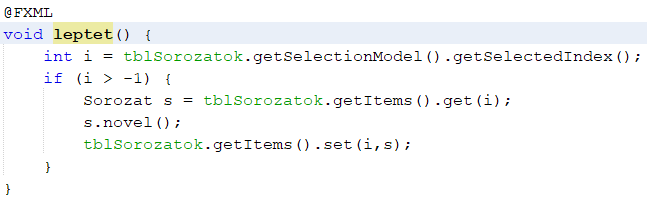
A törlést is akkor tudjuk elvégezni, ha ki van jelölve egy sor. A törlés után a táblázatra visszük a kurzort.

Készítsd el a torol metódust:



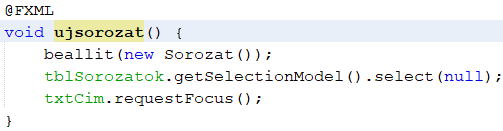
Próbáld ki!

A leptet metódusban is először a kijelölt sort kérdezzük le. Ha ki van jelölve sor, akkor a táblázatból kiolvassuk a hozzá tartozó adatokat, megnöveljük a rész értékét eggyel, majd visszaírjuk a táblázatba:



Próbáld ki, léptesd az egyik sorozatot!

Az Új gomb megnyomásakor visszaállítjuk a felső rész értékeit az alapértékekre, megszüntetjük a kijelölést a táblázatban, és a címre visszük a kurzort:

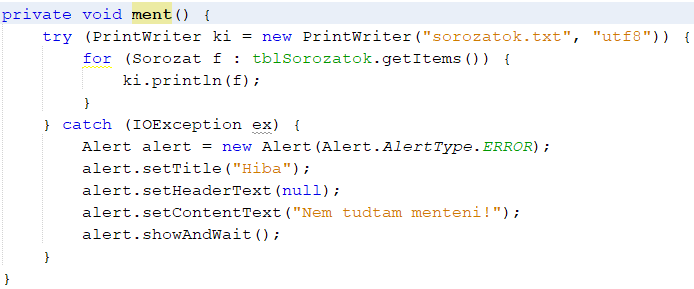


A táblázatban a kijelölést úgy tudjuk megszüntetni, hogy null-ra állítjuk. Erre azért van szükség, hogy véletlenül se lehessen a Módosít gombbal átírni a kijelölt sort.

## Mentés

Ebben a programban a mentést nem kilépéskor, hanem minden adatmódosítás után elvégezzük. Ez a legbiztosabb módszer, de cserébe ennél van a legtöbb lemezművelet.

Készítsd el a ment metódust, amely a táblázatban szereplő sorokat a *sorozatok.txt* nevű, UTF-8 kódolású szöveges fájlba írja! Ha nem sikerül a mentés, jelenjen meg egy hibaüzenet! (Mivel csak ezen az egy helyen van szükség párbeszédpanelre, most nem használjuk a Panel osztályt.)



Ezután minden olyan metódus végén, amely módosítja az adatokat, hívd meg a ment metódust!  
(hozzaad, modosit, leptet, torol)

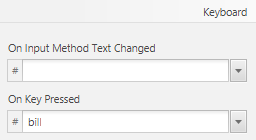
Próbáld ki! Keresd meg és ellenőrizd a *sorozatok.txt* fájlt!

Tedd írásvédetté a *sorozatok.txt* fájlt, és próbáld ki úgy is! Utána szüntesd meg az írásvédettséget!

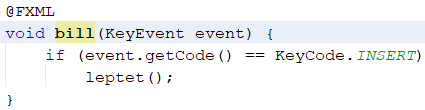
## Billentyűk kezelése

Azt szeretnénk, hogy a táblázatban a kijelölt sorozat részeit az Insert billentyűvel is lehessen léptetni.

Ehhez először állítsd be, hogy a táblázatban egy billentyű lenyomásakor fusson le a bill metódus (Scene Builder, Code lap):



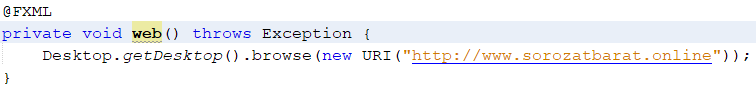
Ezután készítsd el a bill metódust a controller osztályban:



Próbáld ki!

## Weblap megnyitása

Végül készítsd el a web nevű metódust, amely a hivatkozásra kattintva fut le:



Ehhez a Desktop osztályt kivételesen a java.awt csomagból kell importálni!

Próbáld ki!

Állítsd be az ablak címét (Sorozatok), és tiltsd le az átméretezését önállóan!

## Kérdések

1. Mire való a Spinner vezérlő? Hogyan lehet megadni a lehetséges értékeket? Hogyan lehet lekérdezni és beállítani az értéküket?
2. Hogyan adjuk meg egy TableView vezérlő sorainak típusát? Hogyan állítjuk be, hogy melyik mező melyik oszlopban jelenjen meg?
3. Hogyan lehet egy TableView vezérlőhöz sorokat hozzáadni, törölni, módosítani?
4. Hogyan lehet egy TableView vezérlőben a kijelölt sort lekérdezni, a kijelölést módosítani? Hogyan lehet a kijelölés változását figyelni?
5. Hogyan lehetne továbbfejleszteni a programot?