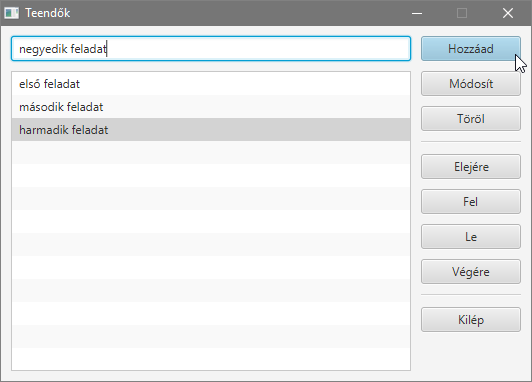
# 13. Teendők

## Feladat

Ebben a leckében egy to-do list programot készítünk, amelyben teendőinket, feladatainkat tarthatjuk nyilván.



A felső szövegmezőbe írhatjuk a következő feladatot, és a Hozzáad gombbal adhatjuk a lista végéhez.

Ha a listában kijelölünk egy elemet, a tartalma beíródik a szövegmezőbe. Ha megváltoztatjuk, majd megnyomjuk a Módosít gombot, a kijelölt elem helyett beíródik a módosított.

Amikor elkészültünk egy feladattal, kijelöljük, majd a Töröl gombbal töröljük a listáról.

A feladatok sorrendjét módosíthatjuk az Elejére, a Fel, a Le és a Végére gombokkal.

Kilépéskor a program elmenti a lista tartalmát egy *teendok.txt* nevű fájlba.

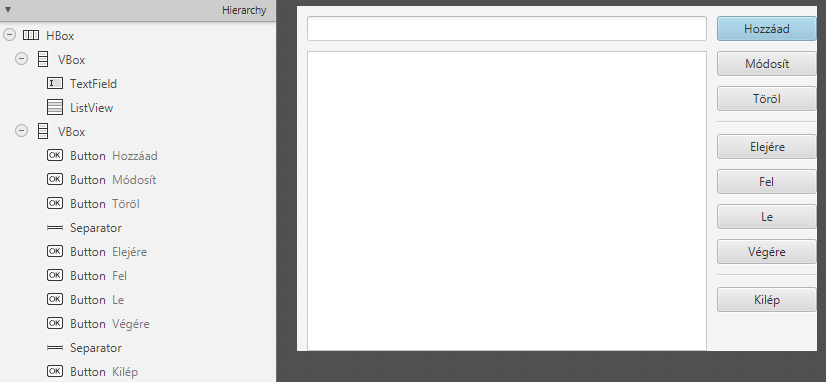
Induláskor megpróbálja betölteni a legutóbb mentett listát. Ha ez nem sikerül, akkor üres listával indulunk.

## Ablak elkészítése

Hogyan rendeznéd el az elemeket az ablakban?

Kezdj egy új JavaFX FXML projektet *teendok* néven!

Készítsd el a program ablakát a Scene Builderben az alábbi minta alapján:



A HBox preferált szélessége számított, térköze 10 pont legyen! Bal oldalon és felül legyen 10 pontos padding-je! A másik két oldalon maradjon 0 a padding! Ez így nem szép, de ha majd letiltjuk az ablak átméretezését, akkor ide kerül két 10 pontos padding.

Mindkét VBox térköze 10 pontos legyen. Ezeknél már nem kell padding és margin. A jobb oldali VBox preferált szélessége 100 képpont legyen!

A TextField preferált szélessége legyen 400 pont, a ListView pedig legyen 400x300-as! A szövegmező azonosítója legyen txtUjFeladat, a listáé pedig lstFeladatok!

A gombok preferált szélessége legyen 100 képpont! Csoportosítsd őket témák szerint a Separator-okkal!

A Hozzáad gomb legyen Default Button, a Kilép gomb pedig Cancel Button!

Az egyes gombok megnyomása a következő metódusokat futtassa: hozzaad, modosit, torol, elejere, fel, le, vegere, kilep!

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 (Teendők), 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!

A ListView-nál add meg a listaelemek típusát (String):  


Töröld az eseménykezelők ActionEvent paramétereit!

Futtasd a programot!

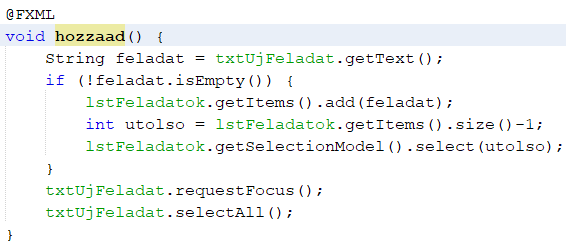
## Feladat hozzáadása, módosítása, törlése

A Hozzáad gomb megnyomásakor a program kiolvassa a szövegmezőbe írt szöveget, és ha ez nem üres, akkor hozzáadja a listához.

Mivel az új elem a lista végére kerül, kijelöli az utolsó elemet.

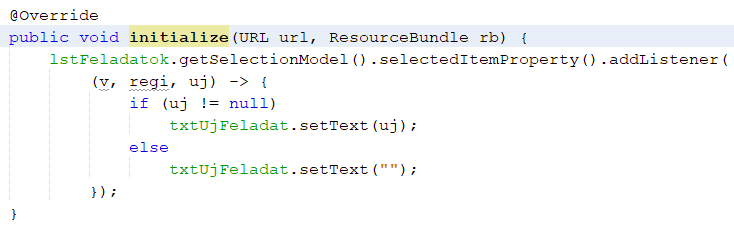
Végül a kurzort a szövegmezőbe viszi a requestFocus metódussal, és kijelöli a szöveget, hogy egyből lehessen írni a következő feladatot.

Készítsd el a hozzaad metódust:



Próbáld ki!

Azt szeretnénk, hogy a listában kijelölt elem megjelenjen a felső szövegdobozban. Ehhez egy eseménykezelőt kell készíteni az initialize metódusban, amely a kijelölt elem változásakor bemásolja az új elemet a szövegdobozba:



Az uj paraméter értéke akkor lesz null, ha letöröljük az utolsó elemet a listából. Ebben az esetben üresre állítjuk a szövegmezőt.

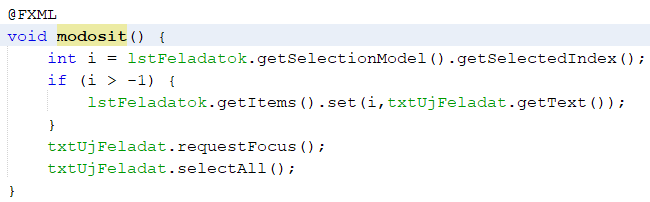
Próbáld ki!

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

Ha ki van jelölve elem, akkor elvégezük a módosítást.

Utána a kurzort visszavisszük a szövegmezőre, és kijelöljük a szöveget.

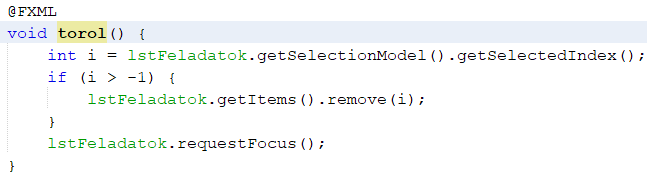
Készítsd el a modosit metódust:



Próbáld ki!

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

Készítsd el a torol metódust:



Próbáld ki!

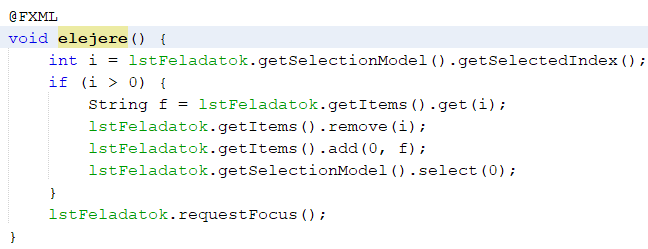
## Listaelemek áthelyezése

Hogyan lehet a listában áthelyezni egy elemet?

Először a lista elejére mozgatást nézzük meg! Ez akkor lehetséges, ha a kijelölt elem indexe 0-nál nagyobb.

Az áthelyezést úgy végezzük, hogy a lista i-dik helyéről töröljük az elemet, majd beszúrjuk a 0-dik helyre, és ki is jelöljük.

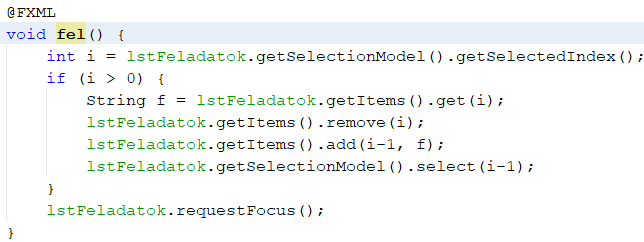
Készítsd el az elejere metódust:



Próbáld ki!

A felfelé mozgatás hasonlóan történik, csak most az i-1-dik helyre szúrjuk be az i-dik helyről törölt elemet.

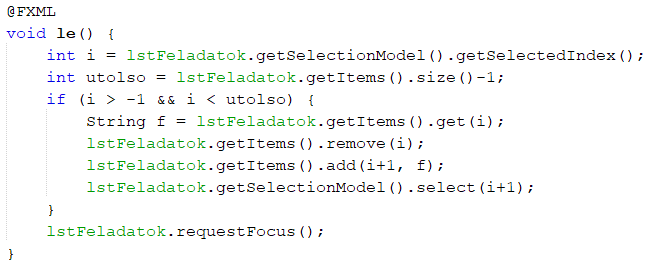
Készítsd el a fel metódust:



Próbáld ki!

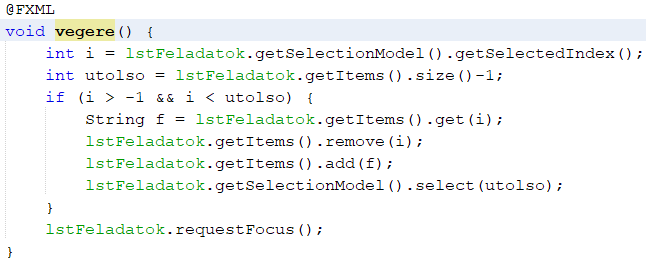
Lefelé mozgatásnál az i-dik helyről törölt feladatot az i+1-dikre szúrjuk be. Ez azonban csak akkor lehetséges, ha nem az utolsó elem volt kijelölve.

Készítsd el a le metódust:



Próbáld ki!

A kijelölt feladat lista végére helyezése is hasonlóan végezhető el:

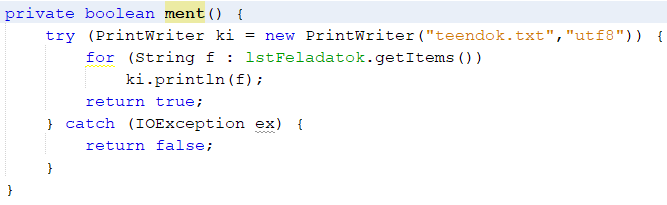


Próbáld ki!

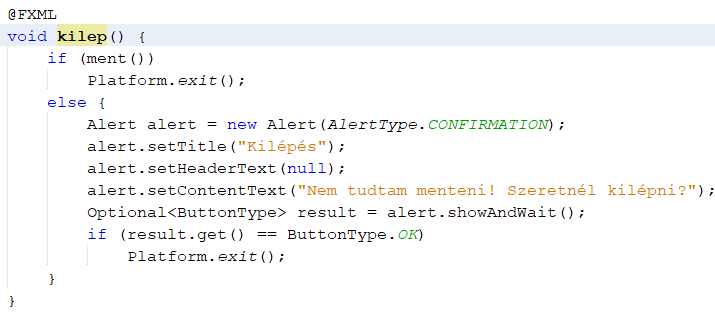
## Mentés és betöltés

A Kilépés gomb megnyomásakor a kilépés előtt szeretnénk megőrizni a lista tartalmát egy *teendok.txt* nevű, UTF-8 kódolású szöveges fájlban.

Készítsd el a ment metódust, amely a listában szereplő feladatokat a szöveges fájlba írja! Ha nem sikerül a mentés, hamis értékert adjon vissza, egyébként igazat!

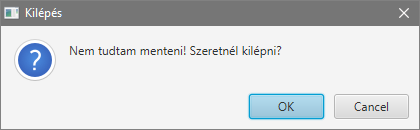


A kilep metódusból meghívjuk a ment metódust. Ha sikerül a mentés, akkor kilépünk, egyébként megjelenítünk egy hibaüzenetet:



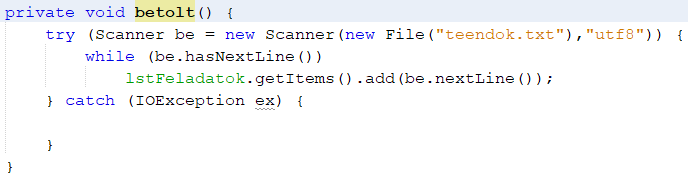
Próbáld ki! Keresd meg és ellenőrizd a szoveg.txt fájlt!

Tedd írásvédetté a *teendok.txt* fájlt, és próbálj meg így is menteni!



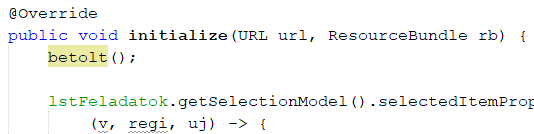
A program indításakor be kell tölteni a *teendok.txt* fájlból az adatokat. Ha nem tudja megnyitni a fájlt, akkor üres listával induljon a program!

Készítsd először a betolt metódust:



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

Ezután hívd meg a betolt metódust az initialize metódusból:



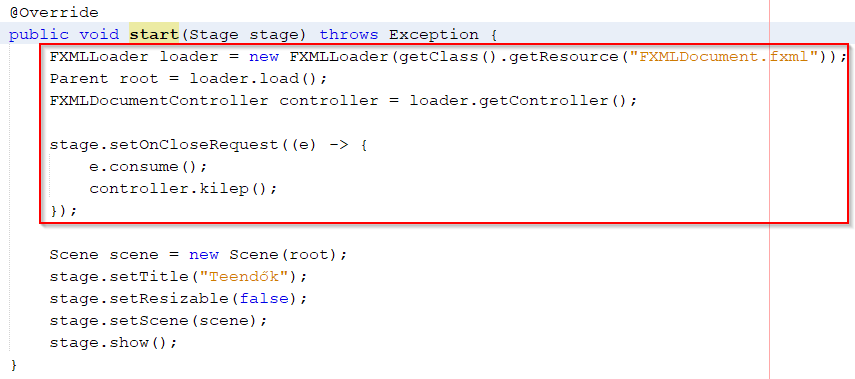
Próbáld ki! Zárd be a programot, majd indítsd el újra!

## Kilépés

Az ablak bezárásakor meg kell hívni a kilep metódust.

Mivel az eseménykezelőt a Teendok osztályban hozzuk létre, és a a kilep metódus a controller osztályban van, szükségünk lesz egy változóra, amely a controller osztályra mutat.

Ehhez sajnos a Teendok osztályban át kell írnunk a start metódus elejét, és utána tudjuk elkészíteni az eseménykezelőt:



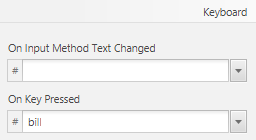
Az esemény consume metódusa megakadályozza az esemény továbbadását.

Próbáld ki!

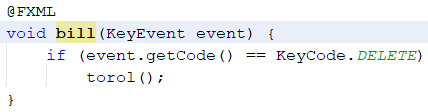
## Billentyűk kezelése

Szeretnénk, hogy a listában a kijelölt elemet a Delete billentyűvel is le lehessen törölni.

Ehhez először állítsd be, hogy a listában egy billentyű lenyomásakor fusson le a bill metódus:



Ezután készítsd el a bill metódust a controller osztályban:



Próbáld ki!

Fontos! A Delete billentyű eseménykezelőjét csak akkor hívja meg a program, ha a listában állva nyomjuk meg a Delete billentyűt. Ha másik elemen állva is szeretnénk használni, akkor ahhoz az elemhez is hozzá kell rendelni a megelelő eseménykezelőt.