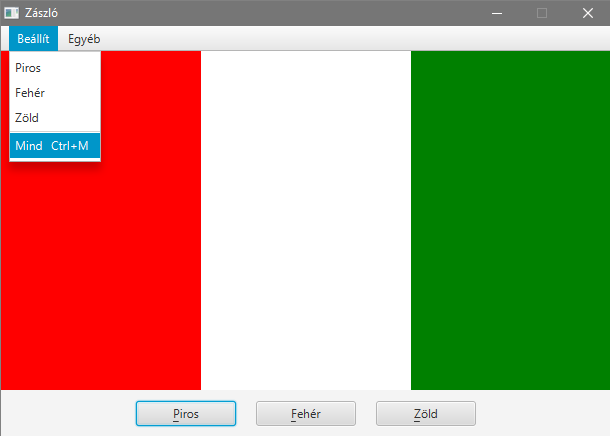
# Menük és párbeszédpanelek

## Feladat

Ebben a leckében egy olyan programot készítünk, amely ablakának középső része háromfelé van osztva. A részek induláskor szürkék, és az alsó részen lévő gombok segítségével külön-külön színezhetők.

A parancsokat menükből is ki lehet választani, és párbeszédpaneleket is készítünk.



## Előkészületek

Kezdj egy új FXML projektet zasz*lo* néven!

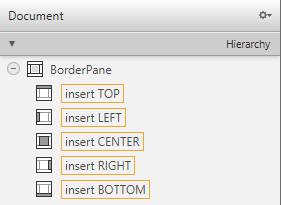
Nyisd meg az FXML fájlt a Scene Builder-rel!

Töröld le az automatikusan létrehozott elemeket! (Mivel ezeknek van fx:id-jük, a törlés előtt figyelmeztet a program.)

Töröld a Controller fájlból is a felesleges kódot!

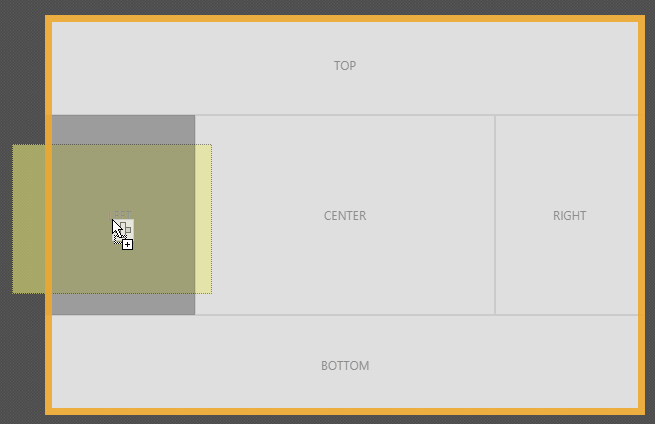
## Jelenet elkészítése

Helyezz el a jeleneten egy BorderPane elemet! Ez öt részből áll, és az egyes részekre különböző elemeket tehetsz.



Állítsd a BorderPane preferált méretét 600x400 pontra!

Helyezz el a bal oldali, a középső és a jobb oldali részbe egy-egy StackPane tárolót! A hierarchiára vagy az ablak megfelelő részére is húzhatod őket.



A StackPane-ek preferált mérete legyen 200 x 400 képpont! fx:id azonosítóik a következők legyenek: spPiros, spFeher, spZold!

Ha a BorderPane valamelyik részére nem teszel semmit, akkor a mellette/fölötte/alatta lévő elemek elfoglalják a helyét. Figyeld meg, hogy a top és a bottom rész eltűnt! A hierarchia segítségével azonban most is tudsz új elemeket tenni ezekre a helyekre.

Helyezz el az alsó részre egy HBox tárolót! Ez a benne elhelyezett elemeket egymás mellé teszi. Figyeld meg, hogy egy tárolóban (BorderPane) helyeztünk el egy másik tárolót (HBox)! (És ezt még lehetne további tárolókkal folytatni.)

Állítsd a HBox Alignment tulajdonságát CENTER-re! Így a rajta lévő elemek középen fognak megjelenni.

A Layout lapon a Spacing tulajdonságot állítsd 20 képpontra! Ekkora helyet fog hagyni a HBox az egyes elemek között.

A HBox preferált magassága legyen 46 képpont!

Helyezz el a HBox-ban három gombot Piros, Fehér és Zöld felirattal! Preferált szélességük legyen 100-100 képpont!

Nézd meg az ablak előnézeti képét (Preview, Show Preview in Window)! Mi történik, ha változtatod az ablak méretét?

Az egyes gombok megnyomásakor a pirosAction, a feherAction, illetve a zoldAction metódusok fussanak le!

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!

## Eseménykezelők

Másold be a kódot a vágólapról, töröld az ActionEvent paramétereket az eseménykezelőkből, és javítsd az importálásokat!

Egy elem stílusát a setStyle metódussal tudjuk beállítani. Ez ahhoz hasonló, mint amikor egy weblapelemre inline stílust alkalmaztunk. Például a következő sor fehérre állítja az spFeher háttérszínét:  
spFeher.setStyle("-fx-background-color: white");

Az egyes elemekhez rendelt stílusokat a getStyle metódussal tudjuk lekérdezni.

Egészítsd ki a metódusok kódját az alábbiak szerint:



Hogyan működnek a metódusok? Figyeld meg a getStyle és a setStyle metódusok használatát!

Nézd át a generált FXML dokumentumot!

A Zászló osztályban állítsd be az ablak címét Zászló-ra!

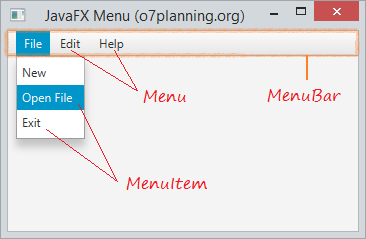
Futtasd a programot!

Próbáld meg átméretezni a program ablakát! Mit tapasztalsz, milyen szélesek lesznek az egyes sávok?

Tiltsd le az ablak átméretezését, majd futtasd így is!

## Menük

Összetettebb programoknál a parancsokat menükből lehet kiválasztani. Ennek részeit az alábbi ábra szemlélteti:



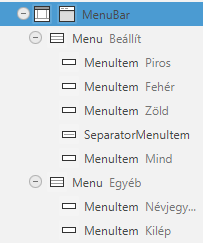
Általában az ablak felső részén van egy menüsor (MenuBar), ezen vannak a menük (Menu) és a menükben a menüpontok (MenuItem).

Nyisd meg az FXML fájlt a Scene Builderrel, és helyezz el a BorderPane felső részén egy MenuBar-t a controls lapról!

Ezen rögtön van néhány menü, és azokban néhány menüpont, amelyeket át kell alakítanod. Ezt az elemek hierarchiájában tudod elvégezni. (A + jelre kattintva nyithatod meg a menü tartalmát.)

A menük elemeit a Menu lapon találod.

Az alábbi ábra mutatja, hogy milyen menüket és menüpontokat kell készítened:



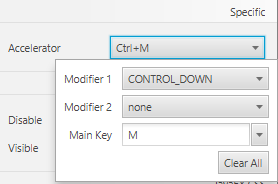
A SeparatorMenuItem egy elválasztó vonal a parancsok között.

Nézd meg az előnézeti képet, és próbáld ki a menüt! (A billentyűzetről a lefelé nyíllal lehet megnyitni egy menüpontot.)

A gyakori menüpontok kiválasztását meg lehet gyorsítani billentyűkombinációkkal.

Például most azt szeretnénk, hogy a Beállít/Mind menüpontot a CTRL+M billentyűkkel is ki lehessen választani.

Ehhez a Mind menüpont Accelerator tulajdonságát kell beállítani az ábra szerint:



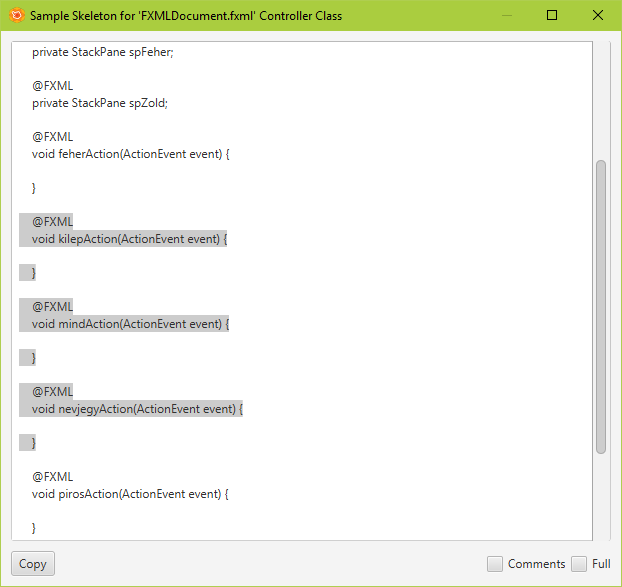
Ezután állítsd be önállóan, hogy a CTRL+X megnyomásával is ki lehessen választani az Egyéb/Kilép parancsot!

Nézd meg az előnézeti képen, hogy ezek a billentyűkombinációk meg is jelennek a menüpontok mellett!

A menüpontok működéséhez jelöld ki őket a hierarchiában, majd a Code lapon add meg, hogy melyik metódusokat hívják meg (On Action):

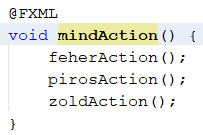
|  |  |
| --- | --- |
| Piros | pirosAction |
| Fehér | feherAction |
| Zöld | zoldAction |
| Mind | mindAction |
| Névjegy… | nevjegyAction |
| Kilép | kilepAction |

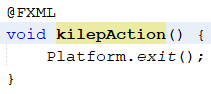
Mentsd a jelenetet, és másold a vágólapra az új elemeket:



Válts vissza a Controller osztályra, töröld az ActionEvent paramétereket, és javítsd az importálásokat!

Egészítsd ki az alábbi metódusokat:



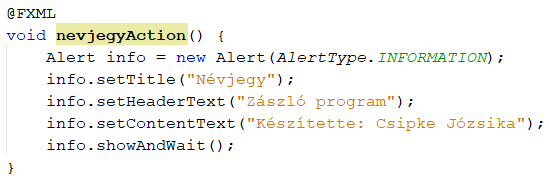


Próbáld ki a menüpontok működését egérrel, billentyűzettel és a gyorsbillentyűkkel is! (A névjegy még nem működik.)

## Párbeszédpanelek

A névjegyet egy párbeszédpanelen szeretnénk megjeleníteni. Ehhez szerencsére nem kell külön ablakot készítenünk, mert készen megkapjuk az Alert osztályban.

Írd ezt a Controller osztályba (a szerzőhöz a saját nevedet add meg):

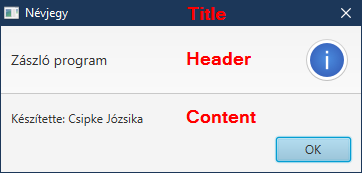


Többféle párbeszédpanel van, az INFORMATION típusú egy i ikont jelenít meg.

Az ablak címe Névjegy. A fejléc (header) szövege felül, a tartalom (content) szövege alatta jelenik meg.

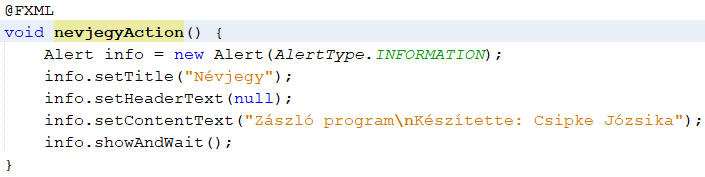
A showAndWait metódus megjeleníti a párbeszédpanelt, és vár, amíg a felhasználó be nem zárja. Az Alert ablakok modálisak, vagyis a program másik ablakára nem tudunk átmenni, amíg be nem zárjuk őket.

Próbáld ki!

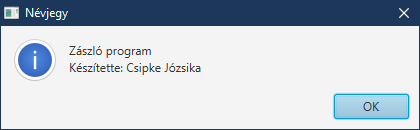


Ha párbeszédpanelen nem szeretnénk fejlécet, akkor null-ra állítjuk az értékét.

Írd át a nevjegyAction metódust így, és próbáld ki!

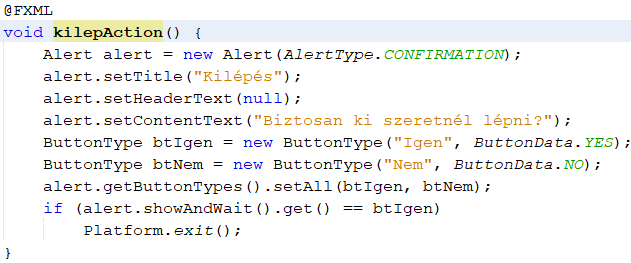


Figyeld meg, hogyan tördeltük két sorba a szöveget!



A kilépés előtt szeretnénk megerősítést kérni. Ehhez készítünk egy új párbeszédpanelt.

Alakítsd át a kilepAction metódust így:

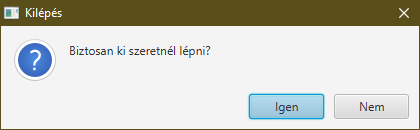


A CONFIRMATION típusú párbeszédpanel kérdőjelet jelenít meg.

Itt saját gombtípusokat (ButtonType) definiáltunk (btIgen, btNem), hogy magyar feliratuk legyen (ne legyen Cancel). Utána hozzáadtuk őket a párbeszédpanelhez (setAll).

A showAndWait metódus eredményéből a get metódussal lehet kiolvasni a megnyomott gombot.

Próbáld ki! Működik Enter-rel és Esc -pel is? (Ez a ButtonData.YES, illetve a ButtonData.NO paramétereknek köszönhető.)



## Kiegészítés

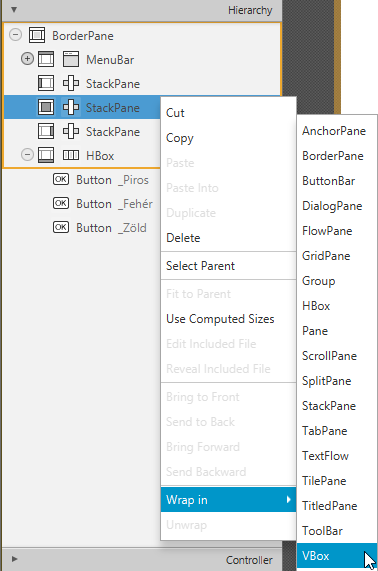
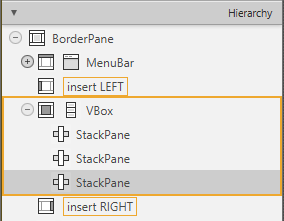
A párbeszédpanelek használatát mutatja be néhány példával a következő oldal:

<http://code.makery.ch/blog/javafx-dialogs-official/>

Nézd meg!

## Vízszintes sávok

Ebben a feladatban a függőleges sávokat vízszintes sávokká alakítjuk. Ehhez egy VBox tárolót használunk, amely egymás alá helyezi a rajta lévő elemeket.

1. Kattints a jobb gombbal a középső StackPane tárolóra, majd válaszd a menüből a Wrap in, VBox parancsot! Ez beleteszi egy VBox tárolóba a kijelölt StackPane-t.  
   
2. Ezután húzd át a hierarchiában a másik két StackPane tárolót is a VBox-ba! (Az egyiket felülre, a másikat alulra.)  
   
3. Állítsd a három StackPane preferált magasságát 150-re, szélességét pedig számítottra!
4. Próbáld ki a programot!

## Kérdések

1. Hogyan használható a BorderPane tároló?
2. Hogyan állítható be és hogyan kérdezhető le egy elem stílusa?
3. Milyen elemekből állnak a menük?
4. Mit jelent az, hogy a párbeszédpanelek modálisak?
5. Mi a különbség a HBox és a VBox tárolók között?