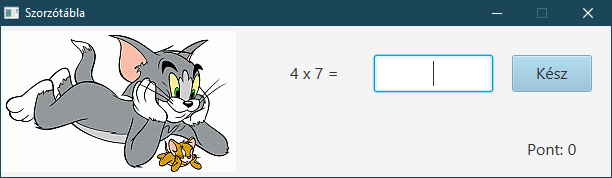
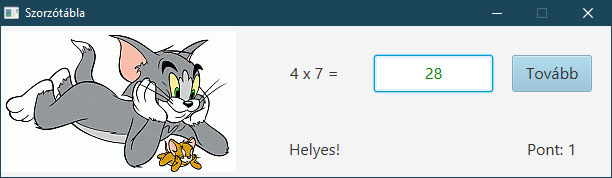
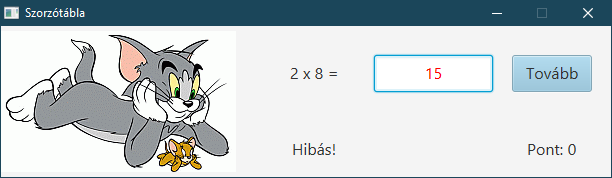
# Szorzótábla

## Feladat

Ebben a leckében egy olyan programot készítünk, amely a szorzótáblát gyakoroltatja egy gyerekkel:







A helyes válaszok eggyel növelik, a hibásak eggyel csökkentik a pontszámot.

A programnak két állapota van:

* Az egyik állapotban várja az eredmény beírását. Ekkor a gomb felirata: Kész. (első kép)
* A másik állapotban értékeli a beírt eredményt. Ekkor a gomb felirata Tovább. (második, harmadik kép)

## Előkészületek

Kezdj egy új JavaFX FXML projektet *szorzotabla* néven!

Nyisd meg az FXML fájlt a Scene Builder-rel!

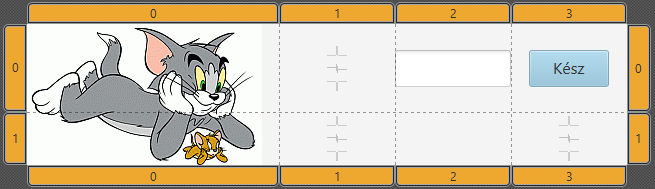
Töröld le az automatikusan létrehozott elemeket! (Mivel ezeknek van fx:id-jük, a törlés előtt figyelmeztet a program.)

Töröld a Controller fájlból is a felesleges kódot!

Másold a kapott képet (*tomesjerry.gif*) a projekt *src* mappájába!

## Jelenet elkészítése

A jelenet elemeit egy GridPane tárolóban fogjuk elhelyezni:



Helyezd el a jeleneten a GridPane tárolót!

Először egy 2 x 3-as táblázatot kapsz, amelyet át kell alakítanod 4 x 2‑esre. Új oszlop vagy sor hozzáadásához kattints a sor vagy az oszlop sorszámára (ez kijelöli), majd a jobb gombos menüből válaszd ki a megfelelő Add … parancsot! Több sort is kijelölhetsz, ha közben nyomod a Shift vagy a Ctrl billentyűket. Így több sort vagy oszlopot tudsz egyszerre beszúrni.

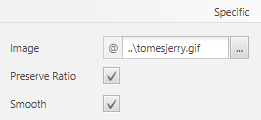
A törlés a sor(ok) vagy oszlop(ok) kijelölése után a Delete paranccsal, az áthelyezés a Move… paranccsal történhet.

A cellák mérete és igazítása még nem megfelelő, de ezt később majd beállítjuk.

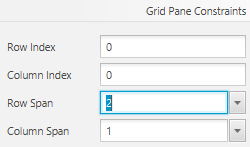
Helyezz el a (0,0) cellában egy ImageView elemet! Ebbe tesszük majd a képet.

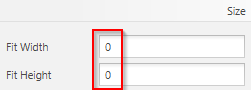
Állítsd a GridPane 0-s oszlopának szélességét és mindkét sorának magasságát számítottra! A többi három oszlop szélessége maradjon az alapértelmezett 100 képpont!

Az ImageView vezérlő image tulajdonságánál add meg a bemásolt képet!



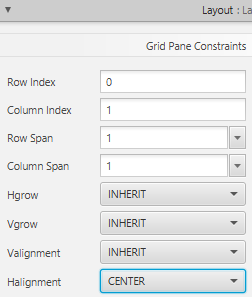
A Layout lapon állítsd be, hogy az ImageView elem két sort foglaljon el, és mérete a kép méretével megegyező legyen!





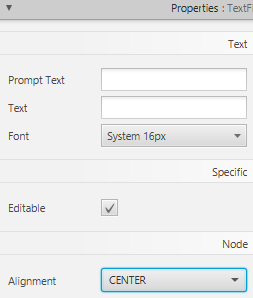
Az 1-es oszlop celláiba tegyél egy-egy címkét (Label), állítsd a betűméretüket 16 képpontra!

Igazítsd mindkettőt középre a cellájában, majd töröld a szövegüket! (Utána a hierarchiában tudod kijelölni őket.)



A felső címke azonosítója (fx:id) legyen lblFeladat, az alsóé lblHelyes!

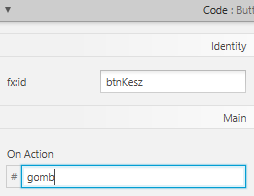
Helyezz el egy TextField vezérlőt a (2,0) cellába! A betűmérete legyen 16px, és vízszintesen is középre igazítson! A mező azonosítója legyen txtMegoldas!



Helyezz el egy gombot (Button) a (3,0) cellába, és igazítsd vízszintesen a cella közepére (mint a címkéket)!

A gomb felirata legyen „Kész”, preferált szélessége pedig 80 képpont! Kapcsold be a Default Button tulajdonságát! Mit jelent ez?

A gomb azonosítója legyen btnKesz, megnyomásakor pedig fusson le a gomb nevű metódus!

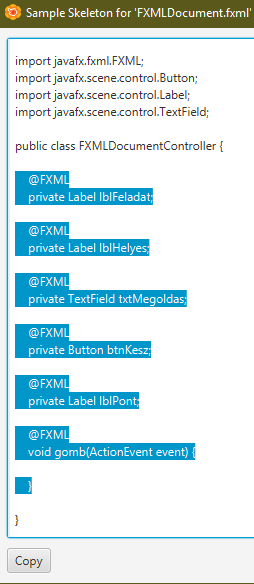


A jobb alsó sarokba, a (3,1) cellába is tegyél egy címkét! Ennek betűmérete is legyen 16px, és legyen középre igazítva! Szövege legyen üres! Azonosítója legyen lblPont!

A GridPane szélessége maradjon 600 képpont, magassága 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!

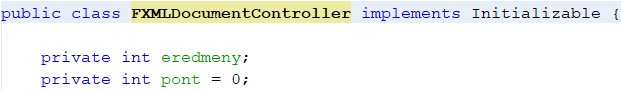
Nézd át a generált FXML dokumentumot!

## Eseménykezelők

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

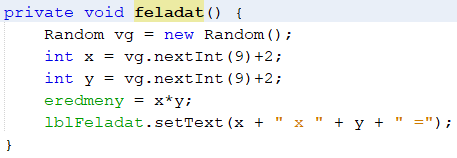
Próbaképpen indítsd el a programot!

Az eseménykezeléshez szükség lesz két változóra, amelyeket az osztály elején deklarálunk:



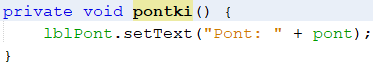
Az eredmeny változó tartalmazza a helyes eredményt, a pont pedig az eddig elért pontszámot, amely induláskor nulla.

A változók után készíts egy feladat nevű metódust, amely a következő feladatot találja ki. Először előállít két véletlen számot 2 és 10 között, a szorzatukat eltárolja az eredmeny változóban, majd a feladatot kiírja a lblFeladat címkébe:

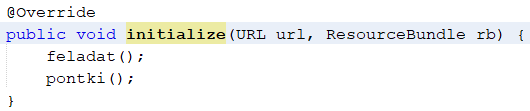


Hívd meg a feladat metódust az initialize metódusból, amely minden programindításkor automatikusan lefut! Próbáld ki! Indítsd el többször is a programot!

Készíts egy pontki nevű metódust is, amely kiírja a pont változó értékét az lblPont nevű címkébe:



Hívd meg ezt is az initialize metódusból:



Futtasd ismét a programot!

Most már „csak” a gomb eseménykezelőjét kell elkészítened. Mint fent, a feladat ismertetésekor írtuk, a programnak két állapota van, amelyeket legegyszerűbben a gomb felirata alapján tudunk megkülönböztetni.

Az egyik állapotban várja az eredmény beírását. Ilyenkor a gomb felirata „Kész”. Ebben az állapotban a következőket kell tenni a gomb megnyomásakor:

* A beírt szöveget átalakítjuk számmá. Ha nem sikerül, akkor a kurzort a szövegmezőre visszük, és a további lépéseket kihagyjuk.
* Ha a megoldás megegyezik a várt eredménnyel, akkor zöldre színezzük a szöveget, kiírjuk, hogy a megoldás helyes, és megnöveljük eggyel a pontszámot!
* Ha nem jó a megoldás, akkor pirosra színezzük, kiírjuk, hogy hibás, és csökkentjük eggyel a pontszámot.
* Az értékelés után kiíratjuk a pontszámot, kikapcsoljuk a szöveg szerkeszthetőségét, és átírjuk a gomb feliratát „Tovább”-ra.

A másik állapotban az értékelést látjuk, és a gomb felirata „Tovább”. Ebben az állapotban a következőket kell tenni a gomb megnyomásakor:

* Ki kell írni egy új feladatot.
* Le kell törölni az előző megoldást, visszakapcsolni a szerkeszthetőségét és a fekete betűszínét, valamint rávinni a kurzort a szövegmezőre.
* Átírni a gomb feliratát „Kész”-re.
* Letörölni az előző értékelést.

Ezeket a feladatokat végzi el a gomb metódus:



Figyeld meg, hogy a setEditable metódussal állíthatjuk egy TextField szerkeszthetőségét, és a requestFocus metódussal tudjuk egy vezérlőre rávinni a kurzort!

Írd be ezt is, és teszteld a programot!

## Feladat

Alakítsd át a programot úgy, hogy 2 és 20 közötti számokkal készítse a feladatokat! Próbáld ki így is!