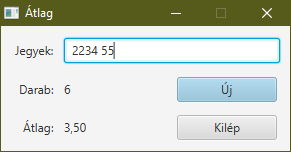
# Átlag

## Feladat

Korábban már írtunk egy parancssoros programot, amely osztályzatok átlagát számítja. Most elkészítjük ennek grafikus felületű változatát:



Ha a karakterek között az 1 és 5 közötti számjegyeken kívül más is van (pl. szóköz), azt a program nem veszi figyelembe. Az átlagot két tizedesjeggyel írja ki.

Az Új gomb vagy ENTER billentyű megnyomásakor törlődnek az eddig beírt jegyek, a Kilép gomb vagy az ESC billentyű megnyomása pedig bezárja a programot.

## Előkészületek

Kezdj egy új JavaFX FXML Java projektet *atlag* 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 egy GridPane tárolót, és alakítsd 3x3-asra! Állítsd a sorok és az oszlopok közötti távolságot 10 képpontra! Legyen körülötte 10 pontos padding!

Helyezz el egy-egy címkét (Label) a 0-s oszlop celláiba, írd be a szövegüket, és igazítsd őket jobbra (Layout lap, Halignment, Right)!

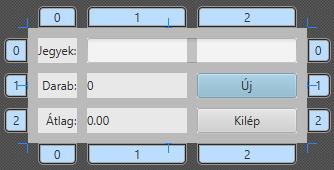
Húzz az (1,0) cellába egy TextField-et! Két oszlopot foglaljon el (Layout lap, Column Span, 2), és azonosítója legyen txtJegyek!

A középső oszlop alsó két cellájába is tegyél egy-egy címkét, és írd be a szövegüket (0, 0.00)! A felső azonosítója legyen lblDb, az alsóé pedig lblAtlag! Az Új gomb legy Default Button, a Kilép gomb pedig Cancel Button!

A jobb oldali oszlop alsó két cellájába helyezz gombokat (Button), és írd át a feliratukat (Új, Kilép)! Mindkét gomb legyen 100 képpont széles! A felső gomb megnyomásakor az uj, az alsó megnyomásakor a kilep metódus fusson le!

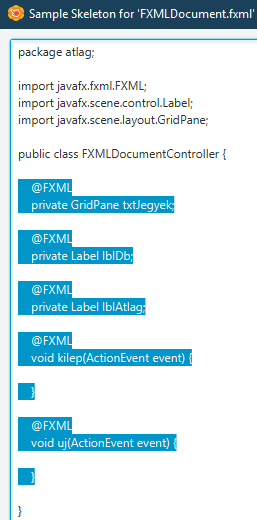
Állítsd a GridPane preferált szélességét és magasságát számítottra!

Legyen az első oszlop szélessége számított, a másik kettőé pedig 100-100 képpont! Legyen minden sor magassága 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!



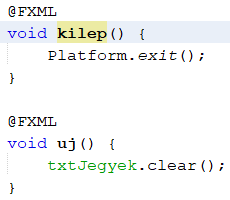
Mentsd a jelenetet, és válts vissza a NetBeans-re!

Nézd át a generált FXML dokumentumot!

## Eseménykezelők

Másold be a kódot a vágólapról a controller osztályba! Javítsd az importálásokat, és töröld az ActionEvent paramétereket! Futtasd a programot!

Készítsd el a kilep és az uj metódusokat:

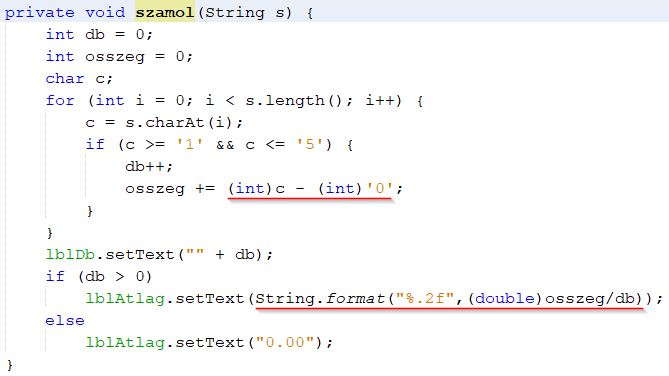


Próbáld ki! Melyik billentyűket lehet használni a gombok helyett?

Az átlag meghatározásához először készítünk egy metódust szamol néven, és utána ezt majd hozzárendeljük a szövegmező megváltozásához.

A szamol metódus paraméterként fogja megkapni a szöveget, amelyből meg kell határoznia a darabszámot és az átlagot. Az eredményt a megfelelő címkékbe fogja kiírni.

Írd be a controllerbe a szamol metódust az alábbiak szerint:



A metódus egy for ciklus segítségével végigmegy a kapott string karakterein (c), de csak az '1' és '5' közötti számjegyeket veszi figyelembe.

A ciklusban a korábban már megismert megszámlálást és összegzést alkalmazza.

A karakter értékét úgy határozza meg, hogy a karakter kódjából kivonja a '0' karakter kódját (első aláhúzás).

Ezután kiírja a darabszámot a felső címkébe.

Előfordulhat, hogy egy osztályzat sincs a stringben, ekkor 0.00-t ír ki a metódus alsó címkébe, egyébként pedig az összeg és a darab hányadosát valós számként, két tizedesre kerekítve (2. aláhúzás).

Elkészíthetnénk a programot úgy is, hogy egy gomb megnyomásakor fusson le a szamol metódus. Most azonban úgy szeretnénk megoldani, hogy amikor gépelünk a szövegmezőbe, minden karakter után automatikusan végezze el a számítást.

A megoldáshoz meg kell ismerkednünk a **property** (tulajdonság) fogalmával. A property egy osztály részeként egy értéket vagy egy objektumot tárol. Ezt le lehet kérdezni és be lehet állítani egy-egy metódussal, és a változását is lehet figyelni egy eseménykezelővel. Az egyes metódusok nevét meghatározott szabályok szerint kell képezni.

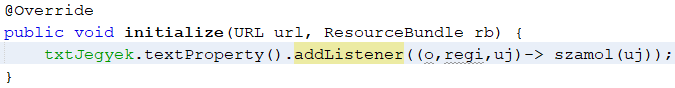
Például tegyük fel, hogy egy property egy keresztnevet tárol. A property neve lehet keresztNev. Ezt a getKeresztNev metódussal lehet lekérdezni és a setKeresztNev metódussal lehet beállítani. Ezen kívül a property-nek rendelkeznie kell egy keresztNevProperty nevű metódussal is, amely magát a keresztnevet adja vissza. Ez utóbbihoz eseménykezelőt is kapcsolhatunk, amely a property változásakor fut le.

A JavaFX osztályainak jó része is használ ilyen property-ket. (Mi is készíthetnénk saját property-ket, de erre most nincs szükségünk, inkább csak az előre elkészítetteket használjuk.)

Például a TextField osztály a text property-t használja egy szövegmező szövegének tárolásához. A szöveget a getText metódussal tudjuk lekérdezni és a setText metódussal beállítani. Ezen kívül van egy textProperty metódusa, amellyel hozzá lehet férni a text property-hez, és eseménykezelőt is lehet a változásához kapcsolni az addListener metódussal. Az eseménykezelő a szöveg tartalmának változásakor fut le (change listener).

Az eseménykezelő három paramétert kap: a property-t, az előző értékét, és az új értékét. Legegyszerűbben Lambda kifejezéssel lehet megadni a controller initialize metódusában, amely induláskor automatikusan lefut.

Írd ezt az initialize metódusba:



Az eseménykezelő három paramétert kap: az o maga a property, a regi a változás előtti értéket tartalmazza, az uj pedig a változás utánit.

Az uj paraméter egy String, a szöveg minden változásakor ezzel hívjuk meg a szamol metódust.

Próbáld ki a programot!

Válts az Atlag osztályra, állítsd be az ablak címét, és tiltsd le az átméretezését! Próbáld ki!

## Kérdések

1. Mi a property? Milyen metódusai vannak?
2. Hogyan lehet eseménykezelőt rendelni egy property-hez? Milyen paramétereket kap az eseménykezelő? Mikor fut le?