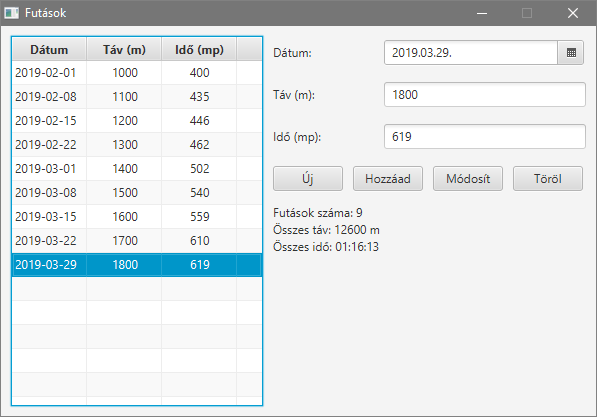
# **Feladat**

Ebben a feladatban egy olyan alkalmazást kell készítenie, amellyel futó edzéseket naplózhat. A futások adatait egy MySQL adatbázisban tárolja.

Az alkalmazást az alábbi ábra mutatja:



Induláskor a program beolvassa az adatokat a *futas* adatbázisból.

Az ablak bal oldalán lévő táblázat mutatja az eddigi futásokat. Minden futásnál meg lehet adni a dátumot, a távot méterben és az időt másodpercben.

Ha kijelölünk egy sort a táblázatban, akkor a program kimásolja a jobb oldali részre a kijelölt sorban lévő adatokat. Ezek módosíthatók, és a módosított változat a Módosít gomb megnyomásával rögzíthető az adatbázisban.

A kijelölt sor törölhető az adatbázisból és a táblázatból a Töröl gombbal.

Az Új gomb megnyomásakor a jobb ldali részben lévő adatok törlődnek, és a dátumhoz a mai dátum íródik be. A beírt új adatokat a Hozzáad gombbal lehet hozzáadni a táblázathoz és az adatbázishoz.

A jobb oldalon, a gombok alatt egy összegzés jelenik meg a futások számával, összes hosszával és idejével.

## **Adatbázis elkészítése**

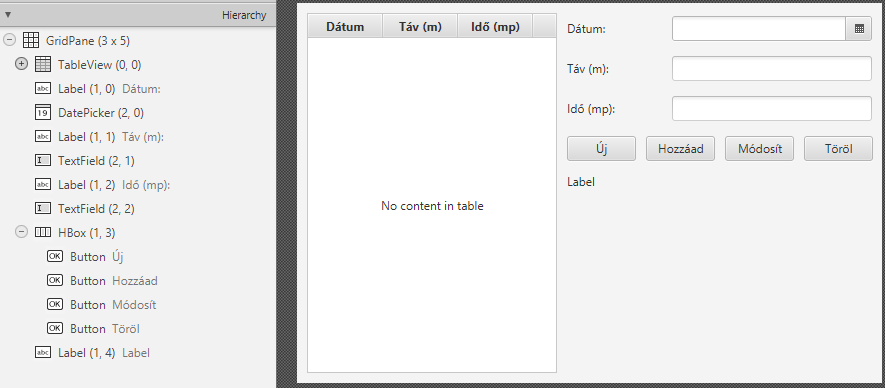
1. Indítsa el a WAMP szervert, majd nyissa meg a PHPMyAdmin programot! Jelentkezzen be (név: root, jelszó nincs), majd készítsen egy új adatbázist *futas* néven, magyar illesztéssel!
2. Importálja a *futas.sql* fájl tartalmát! Nézze meg a kapott tábla szerkezetét és tartalmát!
3. Állítsa be, hogy a *futasID* mező legyen az elsődleges kulcs, automatikus számozással!
4. Állítsa be, hogy ugyanazt a dátumot ne lehessen kétszer megadni!
5. Hozzon létre egy új felhasználót *futo* néven! A jelszava is legyen *futo*! Csak helyben tudjon bejelentkezni, és a *futas* adatbázisban tudjon minden adatokkal kapcsolatos műveletet végezni, de máshoz ne legye joga!
6. Exportálja a *futo* felhasználó jogosultságait a *futo.sql* fájlba!
7. Exportálja az adatbázis tartalmát *megoldas.sql* néven!

## **Osztályok létrehozása**

1. Indítsa el a NetBeans fejlesztőrendszert, és kezdjen egy új JavaFX FXML projektet *futas* néven!
2. A projektben lévő *futas* nevű csomagban hozza létre az alábbi osztályokat:  
   - *Edzes*: a táblázat soraihoz,  
   - *DB*: ebbe kerülnek az adatbázist közvetlenül kezelő metódusok.
3. Adja hozzá a projekt könyvtáraihoz a MySQL JDBC drivert!
4. Hozzon létre a projektben egy panel nevű csomagot, és másolja bele a kapott Panel.java fájlt! Nézze meg a forráskódot! A továbbiakban a Panel osztály metódusait alkalmazza a párbeszédpanelek megjelenítéséhez!
5. Az Edzes osztályban hozza létre a következő privát adattagokat:  
   - futasID: egész szám,  
   - datum: dátum,  
   - tav: egész szám,  
   - ido: egész szám.
6. Készítsen konstruktort az Edzes osztályhoz, amely a paraméterként kapott értékeket tárolja az adattagokban!
7. Készítsen getter és setter metódusokat az Edzes osztály adattagjaihoz!

## **GUI elkészítése**

1. Készítse el a program grafikus felületét a következő ábra alapján:



1. A GridPane méretei számítottak, az egyes oszlopok szélességei: 250, 95 és 200 képpont, a sorok magasságai: 30, 30, 30, 30 és 200 képpont. A cellák közötti távolságok mindkét irányban 10 pontosak legyenek! A padding minden oldalon 10 pontos legyen!
2. A TableView elem azonosítója tblEdzesek, méretei számítottak legyenek, és 5 sort foglaljon el! Az oszlopok nevei oDatum, oTav és oIdo, és mindhárom szélessége 75 képpont legyen! A táv és az idő oszlop tartalmát CSS alkalmazásával igazítsa középre!
3. A DAtePicker azonosítója dpDatum, szélessége 200 képpont legyen! A dátumot csak kiválasztani lehessen!
4. Az alatta lévő két szövegdoboz azonosítói txtTav és txtIdo legyenek!
5. A gombokat egy HBox olyan elemben helyezze egymás mellé, amely két oszlopot foglal el! A gombok közötti térköz 10 képpont legyen!
6. A gombok szélessége 70 képpont legyen! Az általuk elindított metódusok nevei: uj, hozzaad, modosit, illetve torol legyen!
7. A gombok alatti címke két oszlopot foglaljon el, és felülre legyen igazítva a cellában! Szövegét majd a programból fogjuk beállítani. Azonosítója lblStat legyen!
8. Állítsa be a controller osztályt, és mentse a jelenetet!
9. Másolja át a szükséges kódot a controller osztályba! Végezze el a szükséges importálásokat! A kérdőjelek helyére írja be a megfelelő típusokat!
10. Az initialize metódusban adja meg, hogy a TableView oszlopaiba az Edzes osztály melyik mezőit kell megjeleníteni!
11. Állítsa az ablak címét „Futások”-ra, és kapcsolja ki az átméretezhetőségét!

## **Beolvasás és szűrés**

1. A *DB* osztályban készítsen három változót, amely az adatbázis használatához szükséges nevet (futo), jelszót (futo) és URL címet tartalmazza!
2. Készítsen egy *beolvas* nevű metódust a *DB* osztályba, amely beolvassa az adatokat a *naplo* táblából dátum szerinti sorrendben egy paraméterként kapott ObservableList-be! Előtte törölje a lista korábbi tartalmát! Hiba esetén írja ki a kapott hibaüzenetet egy párbeszédpanelen!
3. A controller osztályban hozzon létre *ab* néven egy példányt a *DB* osztályból!
4. A controller osztályban készítsen egy metódust *opm* néven, amely a paraméterként másodpercekben kapott időértékből egy formázott stringet állít elő óra:perc:másodperc formában! Például: 12:08:03
5. A controller osztályban készítsen egy *beolvas* nevű metódust, amely meghívja az *ab* objektum *beolvas* metódusát, majd a táblázat adatai alapján kiírja a futások számát, az összes távot és az összes időt az *lblStat* címkébe a minta szerint! Az idő kiírásához használja az előző pontban elkészített opm metódust!
6. A program indulásakor hívja meg a *beolvas* metódust!
7. Amikor kijelölünk egy sort a táblázatban, ki kell másolni a kijelölt sor adatait a jobb oldali mezőkbe. Készítsen ehhez egy *tablabol* nevű metódust a controller osztályba! A metódus paraméterként kapja meg a kijelölt sor indexét! Ha nincs kijelölt sor, a metódus ne csináljon semmit!
8. Rendelje hozzá a *tablabol* metódust ahhoz az eseményhez, amikor a táblában megváltozik a kijelölés!

## **Új futás hozzáadása**

1. Az Új gomb megnyomásakor a következők történjenek: a DatePickerben jelenjen meg a mai dátum és kerüljön rá a kurzor, a táv és az idő szövegmezők tartalma törlődjön, és a táblázatban szűnjön meg a kijelölés!
2. Készítsen a *DB* osztályban egy *hozzaad* nevű metódust, amely paraméterként kapja meg a *datum*, a *tav* és az *ido* mezők értékeit, majd hozzáadja egy új rekordban az adatbázishoz! A visszaadott érték siker esetén üres String, egyébként a hibaüzenet legyen!
3. Módosítsa a controller osztályban lévő *hozzaad* metódust úgy, hogy a jobb oldali mezőkből olvassa be és ellenőrizze az értékeket, majd hívja meg a *DB* osztályban lévő *hozzaad* nevű metódust ezekkel!   
   A dátumot akkor fogadja el, ha meg van adva, és nem újabb a mai dátumnál!  
   A távot akkor fogadja el, ha számmá lehet alakítani és 100 és 50000 között van, az időnél pedig azt ellenőrizze, hogy 10 vagy annál nagyobb szám-e!  
   Ha sikerült a hozzáadás, olvassa be újra az adatokat az adatbázisból, majd hívja meg az *uj* metódust!

## **Futás módosítása**

1. Készítsen a *DB* osztályban egy *modosit* nevű metódust, amely paraméterként megkapja egy futás azonosítóját, a *datum*, a *tav* és az *ido* mezők értékeit, majd módosítja a rekordot az adatbázisban! A visszaadott érték siker esetén üres String, egyébként a hibaüzenet legyen!
2. Az ablakon lévő Módosít gomb megnyomásakor az eseménykezelő vizsgálja meg, hogy van-e kijelölt sor! Ha van, akkor olvassa be a táblázatból a hozzá tartozó évazonosítót, majd hívja meg a *modosit* metódust az azonosítóval és a jobb oldali mezőkből kiolvasott és ellenőrzött értékekkel! Ugyanúgy végezze az ellenőrzést, mint a hozzáadásnál!  
   Ha sikerült a módosítás, olvassa be újra az adatokat az adatbázisból!
3. Egészítse ki a Módosít gomb eseménykezelőjét azzal, hogy sikeres módosítás után jelölje ki a rekordot a táblázatban!

## **Futás törlése**

1. Készítsen a *DB* osztályban egy *torol* nevű metódust, amely paraméterként megkapja egy futás azonosítóját, majd törli a futást az adatbázisból! A visszaadott érték siker esetén üres String, egyébként a hibaüzenet legyen!
2. Az ablakon lévő Töröl gomb megnyomásakor az eseménykezelő vizsgálja meg, hogy van‑e kijelölt sor! Ha van, akkor kérdezzen rá a törlésre, majd igen válasz esetén olvassa be a táblázatból a hozzá tartozó futásazonosítót, és hívja meg a *torol* metódust az azonosítóval! Végül olvassa be újra az adatokat az adatbázisból!