



Neumann János Egyetem  
Műszaki és Informatikai  
Kar

# JavaFX beadandó

Forint

Katona Zsolt  
OJZGLU

Sándor Adrián  
GMGCIY

2024

# Tartalomjegyzék

BEVEZETÉS .....	4
1. A FELADATRÓL.....	5
2. ADATBÁZIS .....	6
2.1. ADATBÁZIS FELÉPÍTÉSE.....	7
2.1.1. anyag tábla.....	7
2.1.2. tervezo tábla.....	7
2.1.3. erme tábla.....	8
2.1.4. akod tábla.....	8
2.1.5. tkod tábla.....	8
2.1.6. deviza tábla.....	8
2.1.7. params tábla.....	9
3. CRUD MENÜPONT .....	10
4. MNB ADATSZOLGÁLTATÁS MENÜPONT .....	12
5. PÁRHUZAMOS PROGRAMVÉGREGHJÁTÁS MENÜPONT .....	16
6. FOREX MENÜPONT .....	17
ÖSSZEFOGLALÁS .....	22

## Bevezetés

A JavaFX egy Java alapú keretrendszer, mely lehetővé teszi asztali alkalmazások fejlesztését modern, grafikus felhasználói felülettel. A JavaFX eszközöket kínálja a felhasználói felület komponenseinek létrehozásához, mint például gombok, táblázatok és grafikonok. Támogatja az FXML nyelvet is, ami megkönnyíti a felhasználói felület elkülönítését az alkalmazás logikájától, az alkalmazás vizuális része pedig könnyen testreszabható CSS segítségével.

A JavaFX alkalmazásokban az egyik leggyakrabban alkalmazott architektúra az MVC Model-View-Controller. Ez a minta 3 fő komponenst különít el.

- Model: Az adatokat és üzleti logikát tartalmazza
- View: A felhasználói interfészt képviseli, ezek JavaFX-ben általában FXML fájlkként vannak definiálva.
- Controller: Az a réteg amely kapcsolatot biztosít a modellek és nézetek közt.

Az MVC struktúra előnye, hogy elkülöníti a különböző egységeket, így az alkalmazás karbantartása és bővítése könnyebbé válik. Az FXML pedig, a nézet definiálására különösen jól alkalmazható, mivel lehetővé teszi a felhasználói felület fejlesztését programozási logika nélkül.

## **1. A feladatról**

Az elkészült program egy olyan JavaFX alkalmazás, mely egyrészt a Magyar Nemzeti Bank által, limitált darabszámban kiadott forint érmék adatait tartalmazza, jeleníti meg és engedi meg ezen adatok módosítását, bővítését, másrészt kapcsolatot épít az MNB SOAP szolgáltatásával, így a pontos és aktuális képet kaphatunk a jelenlegi, vagy épp a korábbi árfolyamadatokról. Az alkalmazás harmadik nagyobb funkcionális része az OANDA API-val való integráció, amelyet a devizapiaci (Forex) és más pénzügyi eszközök adataihoz és kereskedési szolgáltatásaihoz való hozzáférésre terveztek. Az API lehetővé teszi, hogy automatizált rendszereket építsünk devizakereskedéshez, valós idejű adatelemzéshez vagy éppen történelmi árfolyam adatok letöltéséhez.

A fejlesztés folyamán, elsődlegesen a funkcionalitás megvalósítását helyeztük előtérbe, emiatt a felhasználói felület egy egyszerű, mégis igényes, átlátható megjelenést kapott.

## 2. Adatbázis

A feladat megvalósításának egyik kritériuma volt, hogy az adatbázis nem függhet külső adatbázistól, így könnyen mozgatható az elkészült alkalmazás egyik gépről a másikra.

Ennek függvényében SQLite adatbázist használ a program, ami egy könnyű, önállóan működő, beágyazott adatbázis-kezelő rendszer, ellentétben a hagyományos szerver alapú adatbázisokkal, mint például a MySQL. Az SQLite nem igényel külön adatbázisszervert, mivel minden adat egyetlen fájlban tárolódik.

Az alkalmazás és az SQLite adatbázis közötti kapcsolódás során úgy döntöttünk, hogy **nem használunk objektum-relációs leképezést (ORM)**, hanem közvetlenül natív SQL-lekérdezésekkel kezeljük az adatbázist. Ez a megközelítés több előnnyel is járhat, különösen kisebb adatbázis-kezelési igényű alkalmazások esetén.

Használatának előnyei közé tartozik:

- Teljes kontroll a lekérdezések felett, ami annyit tesz, hogy pontosan meghatározhatjuk a különböző műveleteket (adatbeolvasás, módosítás), így az optimalizálást is közvetlenül kézben tarthatjuk
- Csökkentett függőségi rendszer. ORM használata esetén további könyvtárakat, eszközöket kell beépíteni az alkalmazásba
- Könnyebb hibakezelés. A natív SQL lekérdezésekkel az adatbázissal kapcsolatos problémák feltárása egyszerűsödik

Az adatbázis műveletek kezelésére létrehoztunk egy `databaseHelper`-t, mely az adatbázishoz való csatlakozást, CRUD műveletek megvalósítását biztosítja.

```
/**
 * Adatbázis insert a paraméterben megadott sql-el
 * @param sql
 */
public static void executeSql(String sql) { 9 usages  BobessA
    try (Connection connection = connect();
        Statement stmt = connection.createStatement()) {
        if (connection != null) {
            stmt.executeUpdate(sql);
            System.out.println("Successful");
        }
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    }
}
```

Ahhoz, hogy az adatbázisból kapott adatokat modellekkel tudjuk kezelni és ne kelljen minden modellhez külön megírni a mentést egy általános megoldást alakítottunk ki. A

getRecords metódus paraméterül megkapja a modellt amit válaszként kell adnia az szintén paraméterben kapott sql scriptre.

```
/**
 * Generikus metódus, a paraméterben várt modell alapján rakjuk össze a választ
 * @param sql lekérdezés
 * @param modelT modell
 * @return modell típusú lista
 * @param <T>
 */
public static <T> List<T> getRecords(String sql, Class<T> modelT) { 9 usages  ⬆ BobessA
```

## 2.1. Adatbázis felépítése

A feladat megvalósítására 5 alap adatbázis tábla állt rendelkezésünkre, melyek a következők voltak.

### 2.1.1. anyag tábla

A tábla a korlátozott darabszámban kiadott magyar forint értékhez használt nyersanyagokat tartalmazza. Minden rekord egy konkrét fémre reprezentál. A femid mező, a tábla elsődleges kulcsa, mely azonosítja a fémeket.

Name	Type	NN	PK	AI	U
femid	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
femnev	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### 2.1.2. tervező tábla

A tervező tábla a forint érték tervezőinek adatait tárolja. Minden sor egy konkrét tervezőt reprezentál, beleértve a tervező egyedi azonosítóját és nevét. Ez a tábla a tervezők azonosítására szolgál, amely lehetővé teszi az egyes érmék készítőinek nyomon követését.

Name	Type	NN	PK	AI	U
tid	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
nev	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### 2.1.3. erme tábla

Az erme tábla a limitált darabszámban kiadott magyar forint érmék alapvető adatait tárolja. Minden sor egy érmeadatot reprezentál, annak címletével, tömegével, kiadott darabszámával, valamint a kibocsátás és bevonás dátumával. Ez a tábla az érmék részleteinek, információinak tárolására szolgál.

Name	Type	NN	PK	AI	U
ermeid	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
cimlet	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
tomeg	REAL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
darab	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
kiadas	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
bevonas	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### 2.1.4. akod tábla

Az akod tábla egy kapcsolótábla. Az alapanyagoknak használt fémek, azaz anyag tábla és az érmék adatait tartalmazó erme adatbázistábla adatainak kapcsolatait tartalmazza. A kapcsolótáblára azért volt szükség, mert az anyagok és érmék kapcsolata több a többhöz, egy érmehez több alapanyagot is használhattak, valamint egy alapanyagot több érme anyagaként is felhasználtak.

Name	Type	NN	PK	AI	U
ermeid	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
femid	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### 2.1.5. tkod tábla

A tkod tábla szintén egy kapcsolótábla, ami a tervezők és érmék kapcsolatait reprezentálja. Az érmék és tervezők közt szintén több a többhöz a kapcsolat, több tervező is közreműködhetett egy érme megalkotásában, valamint egy-egy tervező több érménél is szerepet játszott.

Name	Type	NN	PK	AI	U
ermeid	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
tervezoid	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### 2.1.6. deviza tábla

A devizanemeket nem tartalmazta az importálandó állomány. Erre azért volt szükség, hogy a devizanemeket könnyen és egyszerűen megjeleníthessük a ComboBoxokban. Az

MNB adatok letöltésekor, a program letölti az elérhető devizanemeket és beszúrja az esetlegesen új devizákat a táblába.

Name	Type	NN	PK	AI	U
devizanem	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### 2.1.7. *params tábla*

Az MNB adatszolgáltatás menüpontban egy felugró ablakkal kiválasztható az eszközön a letöltött adatok mentési helye. Ezt az elérési utat a paraméter táblába mentjük, így a későbbiekben, ha a fájlt módosítani kell, felülírni, vagy beolvasni, nem lesz szükség újra bekérni annak pontos címét, miután az le van tárolva az adatbázisban.

Ez a tábla a program továbbfejlesztésekor további paraméterekkel bővíthető és használható lenne.

Name	Type	NN	PK	AI	U
param_name	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
param_value	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



### 3. CRUD menüpont

Az alkalmazás menüjének első pontjában az adatbáziskapcsolat formázott a felhasználónak könnyen áttekinthető megjelenítése volt a fő cél.

Az első menüpontban az előző pontban részletezett adatbázis táblák (5 fő tábla) adatait táblázatos formában jeleníthetjük meg. A megjelenítést az allData-view biztosítja, a hozzá tartozó controller pedig az AllDataController.

Adatszolgáltatás

MNB adatszolgáltatás

Párhuzamos programvégrehajtás

Forex

Anyag

Anyag-Ér...

Érme

Érme-Ter...

Tervező

Érme Id	Címlet	Tömeg	Darab	Kiadás dátuma	Bevonás dátuma	
1	1	1.5	227158000	1946-08-01	1995-06-30	
2	1	1.4	431890120	1967-05-12	1995-06-30	
3	1	2.05	483371015	1993-03-29	2008-02-28	
4	2	2.8	13500000	1946-08-01	1951-11-30	
5	2	5.0	57528000	1950-01-20	1971-06-30	
6	2	4.44	303208159	1970-07-01	1995-06-30	
7	2	3.1	467772105	1993-03-29	2008-02-28	
8	5	20.0	39802	1946-08-01	1977-06-30	
10	5	12.09	10004252	1947-05-19	1977-06-30	
11	5	7.4	20029200	1967-05-12	1972-06-30	
12	5	5.73	58284387	1971-08-02	1987-03-31	
13	5	5.0	109668035	1983-04-18	1995-06-30	
14	5	4.2	197772300	1993-06-21		
15	10	8.83	66130376	1971-06-01	1987-03-31	
16	10	6.1	108330025	1983-04-18	1995-06-30	
17	10	6.1	158688505	1993-06-21		
18	20	7.06	108792015	1983-04-18	1995-06-30	
19	20	6.9	171477005	1993-03-29		
20	50	7.7	65406505	1993-03-29		
21	100	9.4	42573505	1993-06-21	1998-12-31	
22	100	8.0	155073000	1996-10-21		

Az adatbázis leginkább látványos megjelenítést biztosító táblája kétségtelenül az érméket reprezentáló tábla. A keresés menüpontban részletes keresést teszünk lehetővé a lenyíló listával, radio gombokkal, jelölőnégyzettel, szöveges beviteli mezőkkel.

[illegible]

A searchErme-view és a SearchErmeController valósítja meg a részletes keresés adatbázislekéréseket.

Az adatrögzítés menüpontban szintén az érme tábla van főszerepben. Szöveges beviteli mezőkkel, dátumválasztókkal adhatók meg az adatok és menthetők az adatbázisba.

A newErme-view és a NewErmeController biztosítja a működést.

Érmék adatainak módosítására is lehetőséget biztosítunk az adatmódosítás menüpontban. Itt táblázatos formában újra megjelennek a jelenlegi rekordok, ezzel segítve a módosítandó adatok bevitelét. A táblázaton dupla kattintásra betöltjük a beviteli mezőkbe a jelenlegi adatokat, ezzel a felhasználó munkáját könnyítve, csak a módosítani kívánt értéket kell megadnia.

Érm...	Cí...	Tö...	Darab	Kiadás dát...	Bevonás dát...
1	1	1.5	227158000	1946-08-01	1995-06-30
2	1	1.4	431890120	1967-05-12	1995-06-30
3	1	2.05	483371015	1993-03-29	2008-02-28
4	2	2.8	13500000	1946-08-01	1951-11-30
5	2	5.0	57528000	1950-01-20	1971-06-30
6	2	4.44	303208159	1970-07-01	1995-06-30
7	2	3.1	467772105	1993-03-29	2008-02-28
8	5	20.0	39802	1946-08-01	1977-06-30
10	5	12.09	10004252	1947-05-19	1977-06-30
11	5	7.4	20029200	1967-05-12	1972-06-30
12	5	5.73	58284387	1971-08-02	1987-03-31
13	5	5.0	109668035	1983-04-18	1995-06-30
14	5	4.2	197772300	1993-06-21	
15	10	8.83	66130376	1971-06-01	1987-03-31
16	10	6.1	108330025	1983-04-18	1995-06-30
17	10	6.1	158688505	1993-06-21	
18	20	7.06	108792015	1983-04-18	1995-06-30
19	20	6.9	171477005	1993-03-29	
20	50	7.7	65406505	1993-03-29	
21	100	9.4	42573505	1993-06-21	1998-12-31
22	100	8.0	155073000	1996-10-21	

A modifiedErme-view a megjelenítésért, a ModifiedErmeController annak kiszolgálásáért felel.

Végül a törlés menüpontban az érme azonosítóját kiválasztva törölhetünk rekordokat az adatbázisból. Itt egy biztonsági funkcióként a beviteli mezőben meg kell adni, az „Érme Törlése!” szöveget, ezzel igyekeztünk védekezni a véletlen adatvesztés ellen.

A törlést a deleteErme-view és DeleteErmeController valósítja meg.

## 4. MNB adatszolgáltatás menüpont

Az MNB SOAP szolgáltatásához csatlakozhatunk és nyerhetünk ki különböző információkat a feladat által megkövetelt módon.

A SOAP kliens létrehozásában a jaxws-maven-plugint használtuk, amely automatikusan képes Java osztályokat generálni egy WSDL (Web Service Definition Language) fájl alapján. Ez a plugin különösen hasznos SOAP webszolgáltatásokkal való integráció során, mivel a generált osztályok leegyszerűsítik a szolgáltatás elérését. A pluginban megadható, hogy hol érhető el a wsdl, ami jelen esetben a <http://www.mnb.hu/arfolyamok.asmx?wsdl> címen található, megadható továbbá hogy a generált osztályokat milyen package-be mentse.

A generált osztályok és a kontroller közt az MNBSoapClientHelper osztály van segítségünkre. Ebben az osztályban valósítottuk meg többek közt az aktuális árfolyamok lekérdezését, az elérhető devizanemek lekérését is. A helper osztály a jaxws-maven-plugin által, a wsdl alapján generált osztályokat a konstruktorban implementálja.

```
public MNBSoapClientHelper() { 3 usages  ⤴ BobessA
    MNBArfolyamServiceSoapImpl impl = new MNBArfolyamServiceSoapImpl();
    this.service = impl.getCustomBindingMNBArfolyamServiceSoap();
}
```

A letöltések menüpontban lehetőségünk van egy txt fájlba menteni a letöltött aznapi árfolyamokat. Ezt egyszerűen megtehetjük a Letöltés gombra kattintva és a felugró ablakban kiválasztva a mentés helyét. A fájl alapértelmezetten MNB néven jön létre, de a FileChooser segítségével más néven is menthető. Ahogy korábban az adatbázisok menüpontban említettük, az itt kiválasztott elérési utat mentjük arra az esetre ha a fájlt később módosítani, vagy beolvasni kell.

A downloadMNBExchanges-view és a DownloadMNBExchangesController áll a menüpont mögött.

A fájl mentéséhez FileChooser-t használtunk, aminek megvalósítást a következő kódrészlet mutatja.

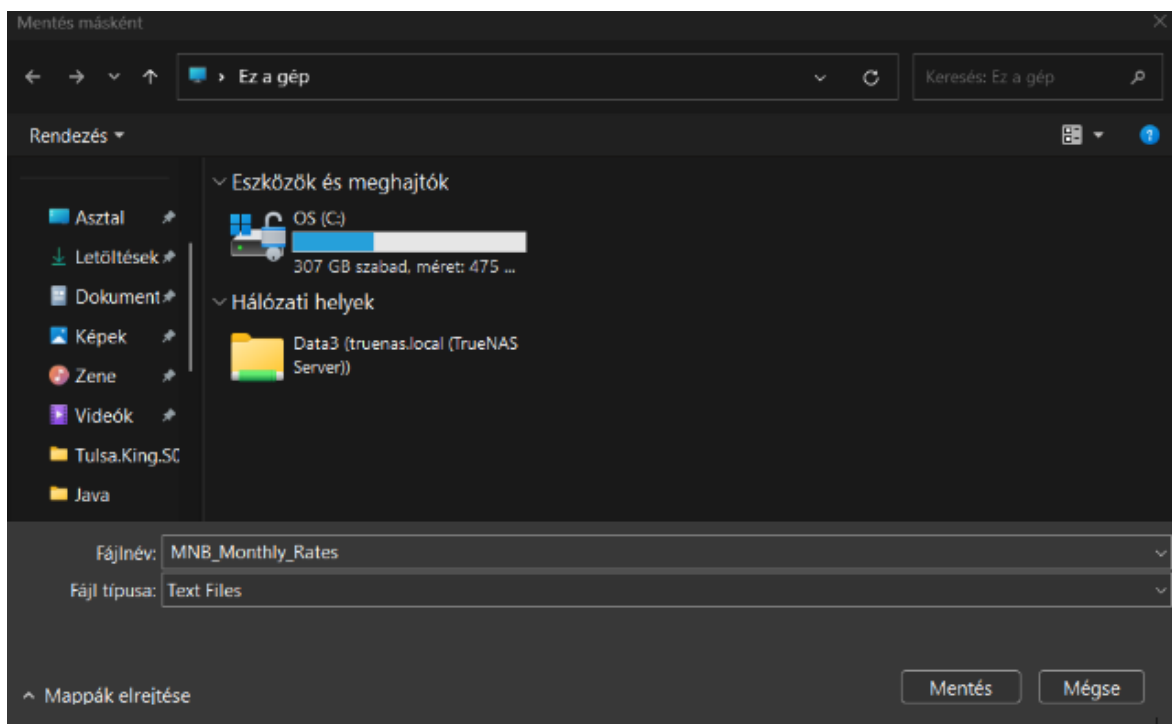
```

FileChooser fileChooser = new FileChooser();
fileChooser.getExtensionFilters().add(new FileChooser.ExtensionFilter("Text Files", "*.txt"));
fileChooser.setInitialFileName("MNB.txt");

File file = fileChooser.showSaveDialog(stage);
try {
    if (file != null) {
        AppConfig.setSavedMNBFilePath(file.getAbsolutePath());
        Params param = new Params("savedMNBFilePath", AppConfig.getSavedMNBFilePath());
        Params.saveParam(param);

        saveExchangesToFile(file);
        resultText.setText("A mentés sikeres volt, a fájl elérhető a\n"+file.getAbsolutePath()+"\ncimen");
    }
} catch (Exception e) {
    showMessage(Alert.AlertType.ERROR, "A fájl mentése sikertelen.", "Hiba");
}
}

```



### Részletes letöltés menüpont

A felhasználónak lehetősége van letölteni egyszerre több árfolyamot is egy fájlba.

A „ListView”-ben akármennyi valuta kiválasztható (ctrl+click).

A „letöltés és mentés” gombra kattintva a felhasználó kiválaszthatja, hogy hova és milyen néven mentse le a szöveges állományt.

A nézetet a Letoltes2Controller vezényli és a downloadMNBExchange2-view.fxml nézetfájl jeleníti meg.

Adatszolgáltatás	MNB adatszolgáltatás	Párhuzamos programvégrehajtás	Forex
Valuta árfolyamok letöltése			
Hónap (YYYY-MM):	<input type="text" value="2024-10"/>		
Válasszon devizákat:			
<div>HUF</div> <div>EUR</div> <div>AUD</div> <div>BGN</div> <div>BRL</div> <div>CAD</div> <div>CHF</div> <div>CNY</div>			
<div>Letöltés és mentés</div>			

A letöltött fájl tartalma:

Valuta: Valuta

Árfolyam adatok a –Dátum– dátumról:

1 Valuta = XY Ft

Kiválasztott valuták egymás alatt kerülnek kiíratásra.

A vezérlő az adatokat a MNBSOAPClientHelper osztálytól veszi.

Az adatok a helper osztályban kerülnek lekérdezésre és formázásra.

```
Valuta: AUD
Árfolyam adatok a 2022-10-28 dátumról:
1 AUD = 265,96 Ft
Árfolyam adatok a 2022-10-27 dátumról:
1 AUD = 262,18 Ft
Árfolyam adatok a 2022-10-26 dátumról:
1 AUD = 265,49 Ft
Árfolyam adatok a 2022-10-25 dátumról:
1 AUD = 264,65 Ft
Árfolyam adatok a 2022-10-24 dátumról:
1 AUD = 264,33 Ft
Árfolyam adatok a 2022-10-21 dátumról:
1 AUD = 264,78 Ft
Árfolyam adatok a 2022-10-20 dátumról:
1 AUD = 264,59 Ft
Árfolyam adatok a 2022-10-19 dátumról:
1 AUD = 264,91 Ft
Árfolyam adatok a 2022-10-18 dátumról:
1 AUD = 263,54 Ft
Árfolyam adatok a 2022-10-17 dátumról:
1 AUD = 268,66 Ft
Árfolyam adatok a 2022-10-14 dátumról:
1 AUD = 271,09 Ft
Árfolyam adatok a 2022-10-13 dátumról:
1 AUD = 280,14 Ft
Árfolyam adatok a 2022-10-12 dátumról:
1 AUD = 277,27 Ft
Árfolyam adatok a 2022-10-11 dátumról:
1 AUD = 276,31 Ft
Árfolyam adatok a 2022-10-10 dátumról:
1 AUD = 277,51 Ft
Árfolyam adatok a 2022-10-07 dátumról:
1 AUD = 277,43 Ft
Árfolyam adatok a 2022-10-06 dátumról:
1 AUD = 277,48 Ft
Árfolyam adatok a 2022-10-05 dátumról:
1 AUD = 275,23 Ft
Árfolyam adatok a 2022-10-04 dátumról:
1 AUD = 275,02 Ft
Árfolyam adatok a 2022-10-03 dátumról:
1 AUD = 277,73 Ft

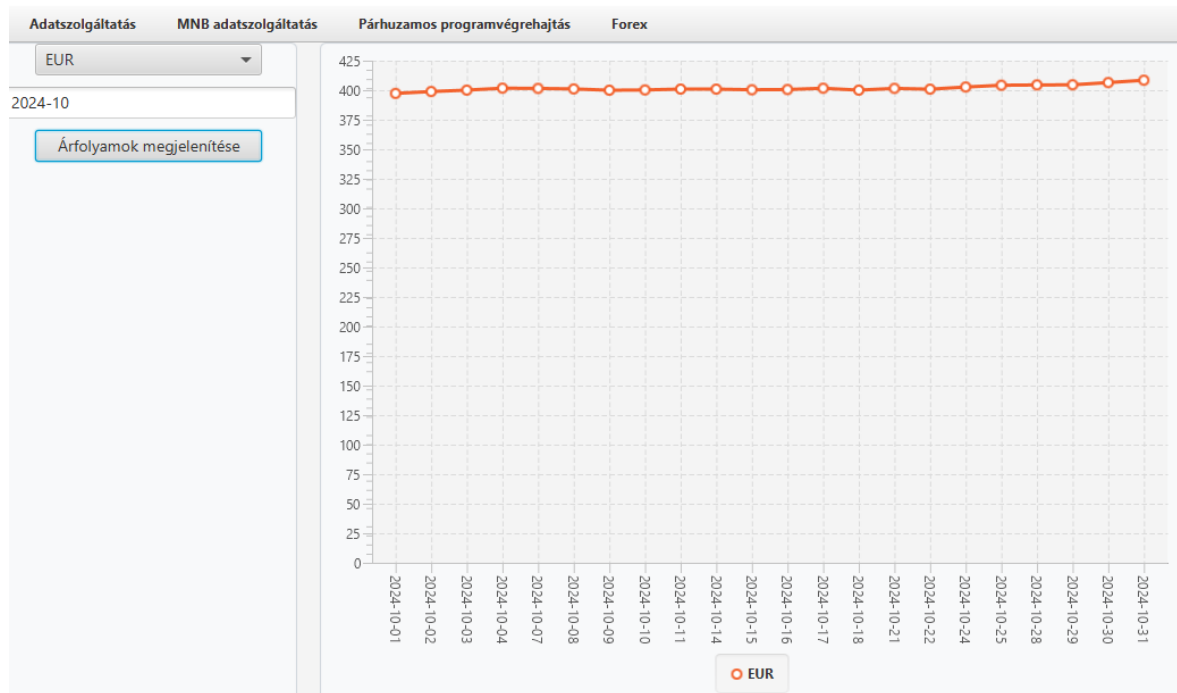
Valuta: BRL
Árfolyam adatok a 2022-10-28 dátumról:
1 BRL = 77,60 Ft
Árfolyam adatok a 2022-10-27 dátumról:
1 BRL = 75,26 Ft
Árfolyam adatok a 2022-10-26 dátumról:
1 BRL = 76,93 Ft
Árfolyam adatok a 2022-10-25 dátumról:
```

## Grafikonos megjelenítés menüpont

Ezen a felületen a felhasználó az általa kiválasztott valuta egy hónapra levetített értékeit grafikonos formában jelenítheti meg.

A felhasználó által megadható adatok:

- Valuta: A lenyíló listából
- Megtekinteni kívánt hónap. (YYYY-MM)



A nézetet a ChartViewController vezényli és a Chart-view.fxml nézetfájl jeleníti meg.

A vezérlő az adatokat a MNBSOapClientHelper osztálytól veszi.

Az adatok a helper osztályban kerülnek lekérdezésre és formázásra.

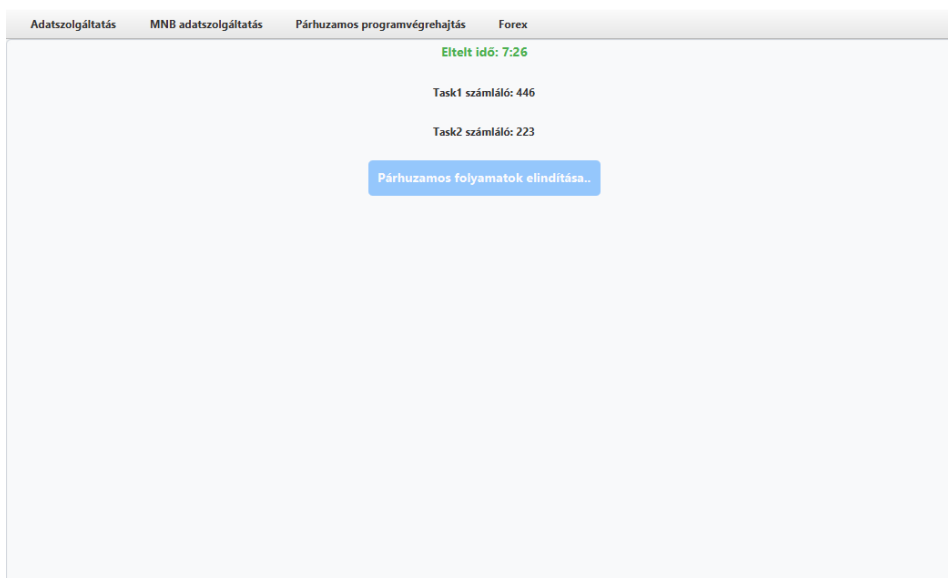
A grafikon a javaFX által kínált beépített megoldás.

## 5. Párhuzamos programvégrehajtás menüpont

A menüpontban szemléltetjük a párhuzamos programvégrehajtást, ahol is egy gomb megnyomására két külön task futása látható. A `startMultiTaskAction()` metódus indítására két Runnable feladat, egy `task1` és egy `task2` indul. A két feladat egymástól függetlenül működik, előbbi az indítás óta eltelt másodperceket jeleníti meg, mivel másodpercenként frissíti a felhasználói felületen megjelenő értéket, míg utóbbi értéke 2 másodpercenként módosul, ezzel szemléltetve, hogy a két task külön szálon fut. Az ütemezés az `executorService` segítségével valósul meg, mely a megadott időközönként futtatja a két független task-ot.

A kód párhuzamos végrehajtást demonstrál a következő módon:

- Két különálló szálon futnak a feladatok, így azok egymástól függetlenül végezhetők el.
- A két feladat eltérő gyakorisággal (1 másodperc és 2 másodperc) hajtódik végre, ami jól mutatja az ütemezés rugalmasságát.
- A GUI-frissítés szálbiztosan történik a `Platform.runLater()` segítségével, amely biztosítja a stabil működést.



A nézetet a `MultitaskController` vezényli és a `multitask-view.fxml` nézetfájl jeleníti meg. A két Task lényegében két időzítő.

```
// Feladatok ütemezése
executorService.scheduleAtFixedRate(task1, initialDelay: 0, period: 1, TimeUnit.SECONDS);
executorService.scheduleAtFixedRate(task2, initialDelay: 0, period: 2, TimeUnit.SECONDS);
```

## 6. Forex menüpont

### Oanda API

Az alkalmazás az OANDA által kínált előre megírt modulokat (<https://github.com/oanda/v20-java/tree/master/src/com/oanda/v20>) használja

A fő azonosítók (token, api url, accountID) a /config/AppConfig fájlban vannak definiálva.

```
/*Forex OANDA*/
public static final String FOREX_API_URL = "https://api-fxpractice.oanda.com"; 6 usages
public static final String FOREX_API_TOKEN = "REDACTED"; 6 usages
public static final AccountID FOREX_API_ACCOUNTID = new AccountID("101-004-30380107-001"); 6 usages
```

Élő tőzsdei adatok révén a hétfégi pozíció nyitás és zárás a devizák esetén nem lehetséges, hisz maga a tőzsdék sem tartanak nyitva. Ilyenkor az alkalmazásban ezen menüpontok funkcionálitása szünetel.

### Számlainformációk almenü:

A fiók / számla információk egy táblázat **szerű** gridben jelennek meg.

Adatszolgáltatás	MNB adatszolgáltatás	Párhuzamos programvégrehajtás	Forex
Számlainformációk			
ID:	101-004-30380107-001		
Alias:	Primary		
Pénznem:	GBP		
Egyenleg:	99999.9879		
NAV:	100000.0064		
Kereskedések száma:	3		
Hedging Engedélyezve:	false		
Pozíciók száma:	2		
Garantált Stop Loss Mód:	ALLOWED		

A lekérdezést egy előre megírt funkció kezeli, melyek megjelenítése a /forex/account.fxml fájlban történik.

```
Context ctx = new Context(
    AppConfig.FOREX_API_URL,
    AppConfig.FOREX_API_TOKEN
);
AccountSummary summary = ctx.account.summary(
    new AccountID(AppConfig.FOREX_API_ACCOUNTID)
).getAccount();
```

### Aktuális árak

Ez a nézet a tőzsdén található és lekérhető, összes valutapárt és az eladási, illetve vételi árat megjeleníti. Valutapárok a nézet megnyitásakor az OandaAPI-ból frissülnek.



Az adathalmaz a fenti keresőmező segítségével szűrhető. A lent elhelyezett „Frissítés” gombbal frissíthető.

Adatszolgáltatás	MNB adatszolgáltatás	Párhuzamos programvégrehajtás	Forex
Aktuális árak			
usd			
Instrum...	Bid Ár	Ask Ár	
BCO_USD	72.977	73.007	
NZD_USD	0.58916	0.58931	
CORN_USD	4.133	4.143	
USD_ZAR	18.0894	18.0981	
EUR_USD	1.05529	1.05537	
SOYBN_U...	9.89	9.908	
XPT_USD	931.48	933.377	
USD_DKK	7.06555	7.06685	
NAS100_...	20849.7	20851.0	
USD_PLN	4.0811	4.083	
USD_CHF	0.88268	0.88284	
USD_THB	34.379	34.421	
WTICO_U...	69.166	69.206	
USD_JPY	151.49	151.512	
USD_HUF	391.081	391.375	
USD_ZAR	18.0894	18.0981	

Frissítés

Az aktuális árak megjelenítését az actualPrices vezérlő kezeli. A megjelenítést az actualPrices.fxml végzi. Az árakat táblázatos nézetben jeleníti meg a program. A táblázat oszlopai a vezérlő inicializálásánál kerülnek beállításra.

Az adatok lekérését egy, már az Oanda által lefejlesztett metódus végzi.

Az adatok kezelésének egyszerűsítése érdekében létrehozásra került egy modell, amely a PriceData nevet viseli. A frissítés gomb megnyomásával a onLoadActualPricesButtonClicked függvény kerül meghívásra.

```
public static class PriceData { 7 usages 1 Zsolt44
    private final String instrument; 2 usages
    private final Double bidPrice; 2 usages
    private final Double askPrice; 2 usages

    public PriceData(String instrument, Double bidPrice, Double askPrice) { 1 usage 1 Zsolt44
        this.instrument = instrument;
        this.bidPrice = bidPrice;
        this.askPrice = askPrice;
    }

    public String getInstrument() {return instrument;}

    public Double getBidPrice() {return bidPrice;}

    public Double getAskPrice() {return askPrice;}
}
```

## Historikus árak

A menüpontban a tőzsdén található és lekérhető összes valutapárt és az elmúlt 10 óra záró árait mutatja meg táblázatos formában. Valutapárok a nézet megnyitásakor az OandaAPI-ből frissülnek. Az adathalmaz a lent elhelyezett „Frissítés” gombbal frissíthető.

Az aktuális árak megjelenítést a historicPrices vezérlő és historicprices.fxml végzi. Az árakat táblázatos érhetők el, melynek oszlopai a vezérlő inicializálásánál kerülnek beállításra.

Adatszolgáltatás	MNB adatszolgáltatás	Párhuzamos programvégrehajtás	Forex
EUR_USD			
Time	Close Price		
2024-11-28 09:00:00	1.05320		
2024-11-28 10:00:00	1.05508		
2024-11-28 11:00:00	1.05500		
2024-11-28 12:00:00	1.05452		
2024-11-28 13:00:00	1.05340		
2024-11-28 14:00:00	1.05492		
2024-11-28 15:00:00	1.05526		
2024-11-28 16:00:00	1.05540		
2024-11-28 17:00:00	1.05528		
2024-11-28 18:00:00	1.05566		
Frissítés			

Az adatok lekérését itt is egy Oanda által lefejlesztett metódus végzi.

Az adatok kezelésének egyszerűsítése érdekében létrehozásra került egy modell, amely a `CandleData` nevet viseli. A frissítés gomb megnyomásával a `onLoadHistoricPricesButtonClicked` függvény kerül meghívásra.

## Pozíció nyitás

A menüpontban a felhasználónak lehetősége van eladási(-) és vételi (+) pozíciót létrehozni.

```
public static class CandleData { 5 usages 1 Zsolti44
    private final String time; 2 usages
    private final String closePrice; 2 usages

    public CandleData(String time, String closePrice) { 1 usage 1 Zsolti44
        this.time = time;
        this.closePrice = closePrice;
    }

    public String getTime() { return time; }

    public String getClosePrice() { return closePrice; }
}
```

A felhasználónak az alábbi adatokat kell megadni a pozíció nyitásához:

- Instrument: A kiválasztott valutapár. Pld: EUR\_USD, EUR\_AUD. Az első érték az amit cserélni szeretne a felhasználó, a második pedig amire cserélni szeretné.
- Units: Hány cserét szeretne. Ha az összeg negatív az short pozíció (eladás), ha pozitív akkor long pozíció (vétel).

AdatszolgáltatásMNB adatszolgáltatásPárhuzamos programvégrehajtásForex

Open Trade

Instrument: EUR\_USD

Units: 100

Open Trade

Active Trades

ID	Instrument	Open Time	Units	Price	Unrealized ...
29	EUR_AUD	2024-11-21 20:09:33	1	1.60937	0.0072
25	EUR_USD	2024-11-21 19:37:27	1	1.04765	0.0060
23	EUR_USD	2024-11-21 19:37:26	1	1.04765	0.0060

A fenti táblázatban a felhasználó láthatja, hogy jelenleg milyen nyitott pozíciói vannak.

A pozíció nyitás az OpenTradeController-rel és az openTrade.fxml-el lehetséges. Az aktuális pozíciók betöltését az ActiveTradeService szolgálja.

```
private void loadActiveTrades() { 2 usages  Zsotti44
    ObservableList<TradeData> activeTrades = tradeService.loadActiveTrades();
    activeTradesTable.setItems(activeTrades);
}
```

A pozíciókat táblázatos nézetben jeleníti meg a program. A táblázat oszlopai a vezérlő inicializálásánál kerülnek beállításra. Az „Open Trade” gomb megnyomásával a **onOpenPositionButtonClicked** függvény kerül meghívásra.

## Pozíció zárás

Itt nyílik lehetőség eladási(-) és vételi (+) pozíciót lezárni.

A felhasználónak az alábbi adatokat kell megadni a pozíció zárásához:

- Trade ID: Egy nyitott pozíció azonosítója. (A lenti táblázatból)

AdatszolgáltatásMNB adatszolgáltatásPárhuzamos programvégrehajtásForex

Close Trade

Trade ID: Enter Trade ID

Close Trade

Active Trades

ID	Instrument	Open Time	Units	Price	Unrealized ...
29	EUR_AUD	2024-11-21 20:09:33	1	1.60937	0.0071
25	EUR_USD	2024-11-21 19:37:27	1	1.04765	0.0061
23	EUR_USD	2024-11-21 19:37:26	1	1.04765	0.0061

```
private void loadActiveTrades() { 2 usages Zsotti44
    ObservableList<TradeData> activeTrades = tradeService.loadActiveTrades();
    activeTradesTable.setItems(activeTrades);
}
```

## Nyitott pozíciók

[illegible]

A pozíciók felületének megjelenítését az ActiveTradesController vezérlő kezeli és az activeTrades.fxml végzi. Az aktuális pozíciók betöltését az ActiveTradeService szolgálja ki.

```
private void loadActiveTrades() { 2 usages  Zsotti44
    ObservableList<TradeData> activeTrades = tradeService.loadActiveTrades();
    activeTradesTable.setItems(activeTrades);
}
```

21

## Összefoglalás

A projekt megvalósításában igyekeztünk pontról, pontra követni a feladat által támasztott követelményeket.

A teljes fejlesztés alatt, az első lépésétől kezdve a Github verziókezelőt használtuk, ahova rendszeresen felkerültek a változások. A repository a [https://github.com/BobessA/Java\\_Forint](https://github.com/BobessA/Java_Forint) címen érhető el.

A feladatokat projektmunka módszert alkalmazva végeztük. A Githubon Katona Zsolt felhasználóneve Zsolti44, míg Sándor Adrián felhasználóneve BobessA.

A Git repository gyökér könyvtárában megtalálható a LETOLTES.zip fájl, ami tartalmazza a jar fájlt, valamint az adatok.db fájlt.

Részfeladatok megjelölése:

Katona Zsolt

- MNB letöltés 2 menüpont megvalósítása
- Grafikonon történő megjelenítés
- OANDA API integráció létrehozása
- Forex menüpont teljes megvalósítása
- Párhuzamos programvégrehajtás megvalósítása
- Meg jelenítés finomhangolása

Sándor Adrián

- A projekt alapjainak létrehozása
- Adatbázis megvalósítása
- Adatbáziskapcsolat kialakítása (databaseHelper alapjai)
- SOAP kliens generálása jaxws-maven-plugin segítségével
- SOAP helper megvalósítása (alapjai)
- CRUD menüpontok megvalósítása
- MNB letöltés megvalósítása