



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.....	6
2.1.1. anyag tábla.....	6
2.1.2. tervezo tábla.....	7
2.1.3. erme tábla.....	7
2.1.4. akod tábla.....	7
2.1.5. tkod tábla.....	8
2.1.6. deviza tábla.....	8
2.1.7. params tábla.....	8
3. CRUD MENÜPONT	9
4. MNB ADATSZOLGÁLTATÁS MENÜPONT	11
5. PÁRHUZAMOS PROGRAMVÉGREHAJTÁS MENÜPONT	14
6. FOREX MENÜPONT	15
ÖSSZEFOGLALÁS	20

A szöveg gépelése és formázása után a tartalomjegyzéket frissíteni kell. A frissítéshez kattintson az egér jobb gombjával a tartalomjegyzék tetszőleges elemére, majd válassza a *Mezőfrissítés* menüpontot! A megjelenő párbeszédablakon válassza a *Teljes jegyzék frissítése* lehetőséget!

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 databaseHelpert, mely az adatbázishoz való csatlakozást, CRUD műveletek megvalósítását biztosítja.

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 érmékhez használt nyersanyagokat tartalmazza. Minden rekord egy konkrét fémre reprezentál. A fémid 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. tervezo tábla

A tervezo 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 érték 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 érték alapvető adatait tárolja. Minden sor egy érmeadatot reprezentál, annak címletével, tömegével, kiadott darabszámával, valamint a kibocsátás és bevonás dátumával. Ez a tábla az érték 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 érték adatait tartalmazó erme adatbázistábla adatainak kapcsolatait tartalmazza. A kapcsolótáblára azért volt szükség, mert az anyagok és érték 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.

Adatszolgáltatás		MNB adatszolgáltatás		Párhuzamos programvégrehajtás		Forex	
		Érme Id	Címlet	Tömeg	Darab	Kiadás dátuma	Bevonás dátuma
<div>Anyag</div> <div>Anyag-Ér...</div> <div>Érme</div> <div>Érme-Ter...</div> <div>Tervező</div>	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]

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

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.

Adatszolgáltatás

MNB adatszolgáltatás

Párhuzamos programvégrehajtás

Forex

Érme módosítása

Érme azonosító

4

Címlet

2

Tömeg

2.8

Darab

13500000

Kiadás dátuma

1946. 08. 01.

Tervezett bevonás

1951. 11. 30.

Módosítás

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	

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.

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-pluginint 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 lekérdezését is.

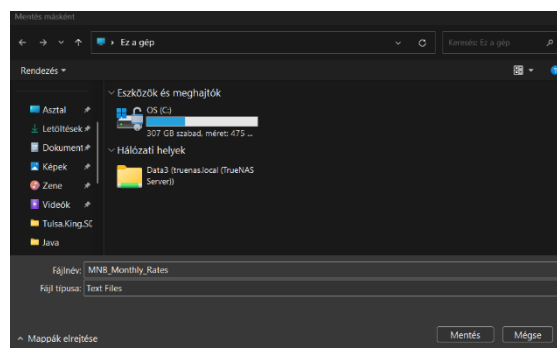
A letöltések menüpontban lehetőségünk van egy txt fájlba menteni a letöltött aznapi árfolyamokat. Ezt egyszerűen megtehetjük a Letöltés gombra kattintva és a felugró ablakban kiválasztva a mentés helyét. A fájl alapértelmezetten MNB néven jön létre, de a FileChooser segítségével más néven is menthető.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

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.



Adatszolgáltatás MNB adatszolgáltatás Párhuzamos programvégrehajtás Forex

Valuta árfolyamok letöltése

Hónap (YYYY-MM):

Válasszon devizákat:

- HUF
- EUR
- AUD
- BGN
- BRL
- CAD
- CHF
- CNY

Letöltés és mentés

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

A letöltött fájl tartalma:

Valuta: Valuta

Árfolyam adatok a –Dátum– dátumról:

1 Valuta = XY Ft

Kiválasztott valuták egymás alatt kerülnek kiírásra.

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.

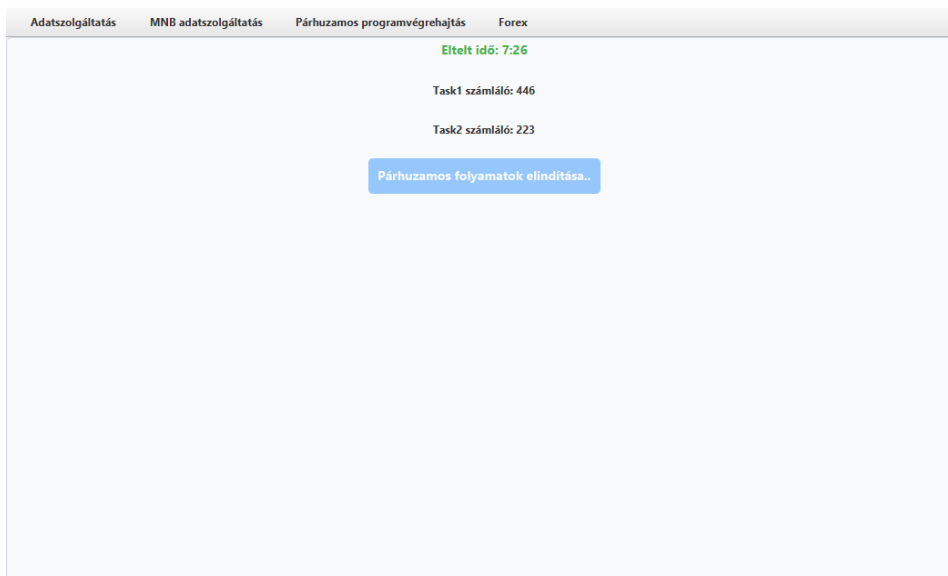
A grafikon a javaFX által kínált beépített megoldás.

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);
```


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

```
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; }
}
```

Pozíció nyitás

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

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 1 Zsolti44
    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.

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);
}
```

Az adatok lekérését ebben az esetben is az Oanda által lefejlesztett metódus végzi. A „Frissítés” gomb megnyomásával a **onCloseTradeButtonClick** függvény kerül meghívásra.

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 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
- Megjelenítés 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