

2024/25/1

Notebook JavaFX

Dokumentáció

Készítették:

Bitó Tamás - I866YG

Mixtaj Flóra - EETLOL

GitHub projekt URL: <https://github.com/bitotamas/Java-alkalmazasok-Eloadas-Beadando>

Bevezető.....	2
A Felhasználói felületet kialakítása	3
A feladatrészek megvalósítása	4
Adatbázis.....	4
Adatbázis menü: egy CRUD alkalmazás a választott adatbázissal.....	5
Olvas almenü.....	5
Olvas - nézet.....	5
Olvas - kontroller.....	6
Olvas2 almenü.....	6
Olvas2 - nézet.....	6
Olvas2 - kontroller.....	7
Ír almenü	8
Ír - nézet	8
Ír - kontroller	9
Módosít almenü.....	10
Módosít - nézet	10
Módosít - kontroller	11
Töröl almenü	12
Töröl - nézet	12
Töröl - kontroller	13
Párhuzamos menü	13
Forex menü	16
Oanda API.....	16
Számlainformációk almenü	16
Aktuális árak almenü.....	17
Historikus árak	17
Pozíció nyitás.....	19
Pozíció zárás	20
Nytott pozíciók	21
Forex kód, azaz a mögöttes tartalom.....	21
Modellek	23
CpU modell.....	23
Id (Int típusú).....	23
Gyarto (String típusú).....	23
Tipus (String típusú)	23
Konstruktorok	23
Getter és Setter metódusok.....	23

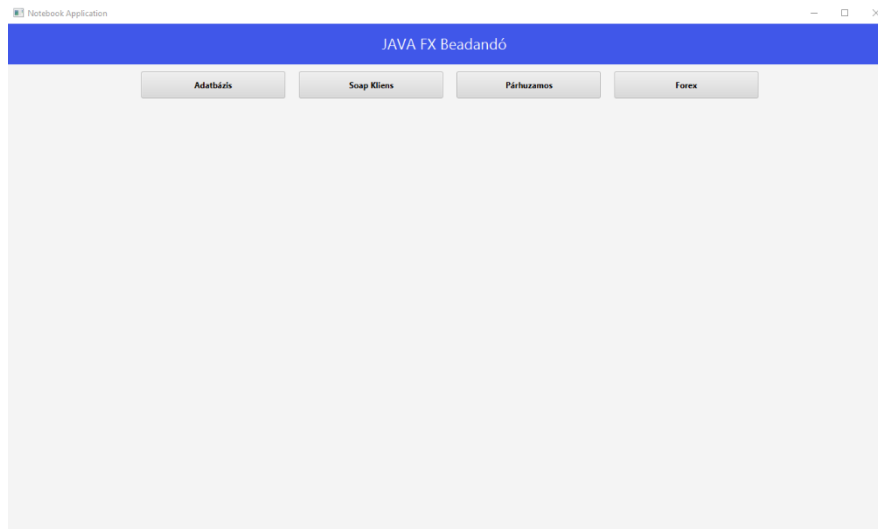
Notebook modell	24
Id (Int típusú).....	24
Gyarto (String típusú).....	24
Tipus (String típusú)	24
Kijelzo (float típusú)	24
Memoria (Int típusú)	24
Merevlemez (Int típusú).....	24
Videovezerlo (String típusú).....	24
Ar (Int típusú)	24
Db (Int típusú)	24
Processzor (CPU)	24
Operációs Rendszer (OS).....	25
Az osztály funkciói:.....	25
Speciális metódusok.....	25
OS modell	25
Id (int típusú).....	25
Nev (String típusú).....	26
Az osztály funkciói:.....	26
Hibernate.cfg.xml.....	26
Kapcsolódás az SQLite adatbázishoz:	26
Entitások kezelésének engedélyezése:	26
Mappingek betöltése:	26
Feladatmegoldásban használt gui elemek	26

BEVEZETŐ

A projekt futtatásához, a LETÖLTÉS.zip fájl kicsomagolva a notebook_app.jar fájl szükséges. Az adatbázis elérésért felelős fájl (feladat.db) a jar fájlal egy mappában kell lennie, ahhoz, hogy az tudja kezelni, így kicsomagolás után rögtön futtatható is.

A FELHASZNÁLÓI FELÜLETET KIALAKÍTÁSA

Az alkalmazás indulásakor megjelenik egy menüválasztó ablak (amit a Main_View.fxml fájl foglal magába), amiben a különböző részfeladatok (Adatbázis, Soap Kliens, Párhuzamos, Forex) megvalósításáért felelős gombok találhatók. Az alkalmazásban az összes .fxml kiterjesztésű nézetet a SceneBuilder segítségével hoztuk létre.



Minden egyes gombhoz tartozó nézetért felelős fájlhoz létre lett hozva egyenként egy-egy controller, ami képes lekezelni, létrehozni, illetve használni a speciális azonosítóval ellátott nézet adatait.

- Main_view.fxml → NotebookController.java
- Db_view.fxml → DBController.java
- Forex_view.fxml → ForexController.java
- Parhuzamos_view.fxml → ThreadsController.java

A főoldalhoz tartozó controller osztály tartalmazza a rajta található gomb eseményeket megvalósító metódusok, amik betöltik a felhasználó által választott aktuális oldalt.

```
//Adatbázis (Db_view) betöltése a Main_view view Pane elemébe
public void loadDatabase(javaafx.event.ActionEvent actionEvent) throws IOException {
    AnchorPane selectedView = FXMLLoader.load(getClass().getResource("Db_view.fxml"));
    view.getChildren().setAll(selectedView);
}

//A számkézelés nézet (Parhuzamos_view) betöltése a Main_view view Pane elemébe
@FXML 1 usage
public void loadParhuzamos(javaafx.event.ActionEvent actionEvent) throws IOException {
    AnchorPane selectedView = FXMLLoader.load(getClass().getResource("Parhuzamos_view.fxml"));
    view.getChildren().setAll(selectedView);
}

//A FOREX kliens (Forex_view) betöltése a Main_view view Pane elemébe
@FXML 1 usage
public void loadForex(javaafx.event.ActionEvent actionEvent) throws IOException {
    AnchorPane selectedView = FXMLLoader.load(getClass().getResource("Forex_view.fxml"));
    view.getChildren().setAll(selectedView);
}
```

A FELADATRÉSZEK MEGVALÓSÍTÁSA

ADATBÁZIS

Az adatbázis egy három táblából álló Notebook és ahhoz tartozó Processzor és Operációs rendszereket képes tárolni. Az adatbázist a DB Browser for SQLite segítségével hoztuk létre, feladat.db néven.

A gep tábla, ami eltárolja az adott notebook azonosítóját, gyártóját, típusát, a kijelző méretét, a memória és a merevlemez nagyságát, a bennlévő videóvezérlőt, az árát, a benne található processzor és operációs rendszer azonosítóját, illetve a raktáron lévő mennyiségét.

Az oprendszer tábla eltárolja az operációs rendszer azonosítóját (ez alapján tudjuk a gep táblában melyik operációs rendszer tartozik hozzá), illetve az operációs rendszer nevét.

A processzor tábla pedig a processzor azonosítóját (hasonló eset az operációs rendszeréhez), gyártóját, illetve típusát tárolja el

Ezáltal az is látszik hogy a táblák össze vannak kötve, készen a több táblás lekérdezésekhez.

Tables (4)	
gep	
id	INTEGER
gyarto	TEXT
tipus	TEXT
kijelzo	REAL
memoria	INTEGER
merevlemez	INTEGER
videovezerlo	TEXT
ar	INTEGER
processzorid	INTEGER
oprendszerid	INTEGER
db	INTEGER
oprendszer	
id	INTEGER
nev	TEXT
processzor	
id	INTEGER
gyarto	TEXT
tipus	TEXT

ADATBÁZIS MENÜ: EGY CRUD ALKALMAZÁS A VÁLASZTOTT ADATBÁZISSAL

OLVAS ALMENÜ

OLVAS - NÉZETE

Az Adatbázis gomb lenyomásával, megvívja a hozzá kötött controller (DBController) osztályt, ami első soron létre hozza az adatbázis kapcsolódást és egyből végre is hajtja az Olvas tabella funkcióját, ahol kiolvassa az adatbázisban található összes Notebook adatát, összekötvé a három táblát, az operációs rendszer és a processzor nevével bővítve.

Notebook Application

JAVA FX Beadandó

Adatbázis Soap Kliens Párhuzamos Forex

Adatbázis menü

Olvas Olvas2 Ír Módosít Töröl

Notebookok

Id	Gyártó	Típus	Képernyő	Memória	Merevlemez	Videóvezérlő	Ár	Processzor	Operációs rendszer	DB
1	HP	COMPAQ 615 NX556EA	15.0	1024	160	ATI Mobility Radeon HD3200 256MB	95120	AMD Athlon 64 X2 QL64	FreeDOS	0
2	ASUS	K51AC-SX001D	15.0	2048	250	ATI Mobility Radeon HD3200 256MB	101200	AMD Athlon 64 X2 QL64	nincs	0
3	HP	COMPAQ 615 NX560EA	15.0	2048	320	ATI Mobility Radeon HD3200 256MB	114800	AMD Athlon 64 X2 QL64	Microsoft Vista Home Basic HU	0
4	HP	Pavilion DV6-1110EH NL956EA	15.0	3072	250	ATI Mobility Radeon HD4530 512MB	167920	AMD Athlon 64 X2 QL64	Microsoft Vista Home Premium HU	3
5	ACER	Aspire 5536G-642G25MN	15.0	2048	250	ATI Mobility Radeon HD4570 512MB	111920	AMD Athlon 64 X2 QL64	Linux	3
6	ACER	Aspire 5536G-643G32MN	15.0	3072	320	ATI Mobility Radeon HD4570 512MB	117520	AMD Athlon 64 X2 QL64	Linux	2
7	MSI	X410-019HU	14.0	2048	320	ATI Radeon Xpress 1250	111920	AMD Athlon TM Neo MV-40	Microsoft Vista Home Premium HU	4
8	ASUS	F83T-VX005X	14.0	4096	500	ATI Mobility Radeon HD4570 512MB	115920	AMD Athlon TM Neo MV-40	nincs	1
9	MSI	VR630XL-004HU	16.0	2048	320	NVIDIA GeForce Go 9100M-GS	90800	AMD Mobil Sempron SL-40	FreeDOS	1
10	ASUS	N60DP-JX012V	16.0	4096	500	ATI Mobility Radeon HD4670 512MB	183920	AMD Turion64 X2 TL60	Windows 7 Home Premium HU 64Bit	4
11	ASUS	K50AB-SX045D	15.0	4096	500	ATI Mobility Radeon HD4570 512MB	134320	AMD Turion64 X2 TL64	nincs	5
12	FUJITSU	Amilo S3650	13.0	2048	250	ATI Mobility Radeon HD3200 256MB	223920	AMD Turion64 X2 TL62	Microsoft Vista Home Premium HU	4
13	MSI	WIND U200-064HU	12.0	2048	320	Intel Graphics X4500M / 256MB	112400	Intel Celeron M ULV723	Microsoft Vista Home Premium HU	3
14	ACER	eMachine E525-901G16Mi	15.0	1024	160	Intel Graphics 4500MHD	82800	Intel Celeron 900	Linux	0
15	DELL	Inspiron 1545-106208 RED	15.0	1024	160	Intel Graphics 4500MHD	103120	Intel Celeron 900	Linux	3
16	TOSHIBA	Satellite L500-1EN	15.0	2048	320	Intel Graphics X4500M / 256MB	118800	Intel Celeron 900	FreeDOS	1
17	MSI	CR500X-012HU	15.0	2048	320	NVIDIA GeForce Go 8200M 128MB	94800	Intel Celeron 900	FreeDOS	1
18	MSI	CR500X-008HU	15.0	4096	320	NVIDIA GeForce Go 8200M 128MB	95920	Intel Celeron 900	FreeDOS	3
19	LENOVO	SL500 2746P5G	15.0	1024	160	NVIDIA GeForce Go 9300M 256MB	139920	Intel Celeron Dual-Core T1600	Microsoft Vista Home Basic HU	0
20	TOSHIBA	Satellite L300-24P	15.0	2048	160	Intel Graphics 4500MHD	98000	Intel Celeron Dual-Core T1600	FreeDOS	0
21	MSI	VR603X-094HU	15.0	4096	320	Intel Graphics 4500MHD	99600	Intel Celeron Dual-Core T1600	FreeDOS	5

```
@FXML
public void initialize(URL url, ResourceBundle resourceBundle) {
    if (session == null || !session.isOpen()) {
        StandardServiceRegistry registry = new StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();
        Metadata metadata = new MetadataSources(registry).buildMetadata();
        factory = metadata.getSessionFactoryBuilder().build();
        session = factory.openSession();
    }
    setNotebookTable(tv1);
}
```

```
@FXML 1 usage
public void menuReadClick(Event event) {
    setNotebookTable(tv1);
}
```

OLVAS - KONTROLLER

A `setNotebookTable` egy paraméteres metódus, ami a paraméterben megadott Notebook típusú TableView sorait, oszlopait állítja be, ami későbbi feladatokban is alkalmazva lett. Az adatokat egy lekérdezéssel egy Notebook típussal ellátott Listában tároljuk és onnan töltjük be.

```
@FXML 3 usages
public void setNotebookTable(TableView<Notebook> tv) {
    tv.getColumns().clear();
    tv.getItems().clear(); // Előző oszlopok törlése
    // Oszlopok létrehozása a Notebook osztály mezői alapján
    IDCol = new TableColumn<>("Id");
    gyartoCol = new TableColumn<>("Gyártó");
    tipusCol = new TableColumn<>("Típus");
    kijelzoCol = new TableColumn<>("Képernyő");
    memoriaCol = new TableColumn<>("Memória");
    merevlemezCol = new TableColumn<>("Merevlemez");
    videovezerloCol = new TableColumn<>("Videóvezérlő");
    arCol = new TableColumn<>("Ár");
    processzorCol = new TableColumn<>("Processzor");
    oprendszerCol = new TableColumn<>("Operációs rendszer");
    dbCol = new TableColumn<>("DB");
    // Oszlopok hozzáadása a táblához
    tv.getColumns().addAll(IDCol, gyartoCol, tipusCol, kijelzoCol, memoriaCol, merevlemezCol, videovezerloCol, arCol, processzorCol, oprendszerCol, dbCol);
    processzorCol.setCellValueFactory(new PropertyValueFactory<>("processzorNev"));
    oprendszerCol.setCellValueFactory(new PropertyValueFactory<>("operaciosRendszerNev"));
    // Beállítjuk, hogy az oszlopokhoz melyik Notebook mezőt társítjuk
    IDCol.setCellValueFactory(new PropertyValueFactory<>("Id"));
    gyartoCol.setCellValueFactory(new PropertyValueFactory<>("Gyarto"));
    tipusCol.setCellValueFactory(new PropertyValueFactory<>("Típus"));
    kijelzoCol.setCellValueFactory(new PropertyValueFactory<>("Kijelzo"));
    memoriaCol.setCellValueFactory(new PropertyValueFactory<>("Memoria"));
    merevlemezCol.setCellValueFactory(new PropertyValueFactory<>("Merevlemez"));
    videovezerloCol.setCellValueFactory(new PropertyValueFactory<>("Videovezerlo"));
    arCol.setCellValueFactory(new PropertyValueFactory<>("Ar"));
    dbCol.setCellValueFactory(new PropertyValueFactory<>("Db"));
    Transaction transaction = session.beginTransaction();
    List<Notebook> notebookLista = session.createQuery("FROM Notebook", Notebook.class).list();
    for (Notebook notebook : notebookLista) {
        tv.getItems().add(notebook);
    }
    transaction.commit();
}
```

OLVAS2 ALMENÜ

OLVAS2 - NÉZET

Az olvas2 almenü egy az oldalon megjelenő űrlap segítségével/kitöltésével tud szűrni az alatta lévő, adatbázisban található Notebookokra. Az űrlap tartalmaz egy szöveg beviteli mezőt a kulcsszavas kereséshez, egy lenyíló listát, ami tartalmazza az összes gyártót, 4 rádió gombot, amivel ki tudja választani milyen márkájú videóvezérlőt szeretne, egy jelölőnégyzetet, ami a raktári elérhetőséget adja meg és mind ezek után egy gomb, ami megvalósítja a szűrést.

Java FX Beadandó

Adatbázis Soap Kliens Párhuzamos Forex

Adatbázis menü

Olvas Olvas2 fr Módosít Töröl

Szűrt Notebookok Összesen: 246 db.

Keresés...

Válasszon egy gyártót

Videóvezérlő gyártója

Intel ATI NVIDIA VIA Raktáron van

Szűrés

Id	Gyártó	Típus	Képernyő	Memória	Merevlemez	Videóvezérlő	Ár	Processzor	Operációs rendszer	DB
1	HP	COMPAQ 615 NX556EA	15.0	1024	160	ATI Mobility Radeon HD3200 256MB	95120	AMD Athlon 64 X2 QL64	FreeDOS	0
2	ASUS	K51AC-SX001D	15.0	2048	250	ATI Mobility Radeon HD3200 256MB	101200	AMD Athlon 64 X2 QL64	nincs	0
3	HP	COMPAQ 615 NX560EA	15.0	2048	320	ATI Mobility Radeon HD3200 256MB	114800	AMD Athlon 64 X2 QL64	Microsoft Vista Home Basic HU	0
4	HP	Pavilion DV6-1110EH NL956EA	15.0	3072	250	ATI Mobility Radeon HD4530 512MB	167920	AMD Athlon 64 X2 QL64	Microsoft Vista Home Premium HU	3
5	ACER	Aspire 5536G-64G2G25MN	15.0	2048	250	ATI Mobility Radeon HD4570 512MB	111920	AMD Athlon 64 X2 QL64	Linux	3
6	ACER	Aspire 5536G-64G32MN	15.0	3072	320	ATI Mobility Radeon HD4570 512MB	117520	AMD Athlon 64 X2 QL64	Linux	2
7	MSI	X410-019HU	14.0	2048	320	ATI Radeon Xpress 1250	111920	AMD Athlon TM Neo MV-40	Microsoft Vista Home Premium HU	4
8	ASUS	F83T-VX005X	14.0	4096	500	ATI Mobility Radeon HD4570 512MB	115920	AMD Athlon TM Neo MV-40	nincs	1
9	MSI	VR630XL-004HU	16.0	2048	320	NVIDIA GeForce Go 9100M-GS	90800	AMD Mobil Sempron SI-40	FreeDOS	1
10	ASUS	N60DR-X0012V	16.0	4096	500	ATI Mobility Radeon HD4670 512MB	183920	AMD Turion64 X2 TL60	Windows 7 Home Premium HU 64bit	4
11	ASUS	K30AB-SX045D	15.0	4096	500	ATI Mobility Radeon HD4570 512MB	134320	AMD Turion64 X2 TL64	nincs	5
12	FUJITSU	Amlio S6350	13.0	2048	250	ATI Mobility Radeon HD3200 256MB	223920	AMD Turion64 X2 TL62	Microsoft Vista Home Premium HU	4
13	MSI	WIND U200-064HU	12.0	2048	320	Intel Graphics X4500M / 256MB	112400	Intel Celeron M ULV723	Microsoft Vista Home Premium HU	3
14	ACER	eMachine E525-901G16Mi	15.0	1024	160	Intel Graphics 4500MHD	82800	Intel Celeron 900	Linux	0
15	DELL	Inspiron 1545-106208 RED	15.0	1024	160	Intel Graphics 4500MHD	103120	Intel Celeron 900	Linux	3
16	TOSHIBA	Satellite L500-1EN	15.0	2048	320	Intel Graphics X4500M / 256MB	118800	Intel Celeron 900	FreeDOS	1
17	MSI	CR500X-012HU	15.0	2048	320	NVIDIA GeForce Go 8200M 128MB	94800	Intel Celeron 900	FreeDOS	1

OLVAS2 - KONTROLLER

A táblázat betöltéséhez egy `setNotebook2` metódust hoztunk létre, ami paraméterül vár egy lekérdezést. Ezt a lekérdezést a Szűrés gomb generálja le, ami végig vizsgálja az űrlap elemeit és egymáshoz fűzi egy egész lekérdezésbe az értékeit.

```
public void btSearch(ActionEvent actionEvent) { 1 usage

//Megnézzük melyik van kiválasztva, az true értékkel tér vissza
List<Boolean>radios=new ArrayList<>();
radios.add(radioButtonINTEL.isSelected());
radios.add(radioButtonATi.isSelected());
radios.add(radioButtonNVIDIA.isSelected());
radios.add(radioButtonVIA.isSelected());
if(radios.get(0)){
    hql += "AND videovezerlo LIKE 'Intel%'";
}
if(radios.get(1)){
    hql += "AND videovezerlo LIKE 'ATi%'";
}
if(radios.get(2)){
    hql += "AND videovezerlo LIKE 'NVIDIA%'";
}
if(radios.get(3)){
    hql += "AND videovezerlo LIKE 'VIA%'";
}
if(!searchText.isEmpty()) {
    hql += " AND (gyarto LIKE :searchText OR tipus LIKE :searchText OR videovezerlo LIKE :searchText)";
}
if(comboGyartok.getSelectionModel().getSelectedIndex()!=0){
    hql += " AND gyarto = :getGyarto";
}
if(checkBox.isSelected()){
    hql += " AND db>0";
}
// Adatok lekérése a Hibernate segítségével
Transaction transaction = session.beginTransaction();
Query query = session.createQuery(hql,Notebook.class);
if (searchText!="") {
    query.setParameter(s: "searchText", searchText); // searchText paraméter
}
if (getGyartoIndex!=0) {
    query.setParameter(s: "getGyarto", getGyarto);
}
transaction.commit();
```


A setNotebook2Table pedig elvégzi az adatlekérést

```
radioButtonINTEL.setOnAction(e -> {
    if (radioButtonINTEL.isSelected()) {
        radioButtonATI.setSelected(false); //
        radioButtonNVIDIA.setSelected(false);
        radioButtonVIA.setSelected(false);
    }
});
radioButtonATI.setOnAction(e -> {
    if (radioButtonATI.isSelected()) {
        radioButtonINTEL.setSelected(false);
        radioButtonNVIDIA.setSelected(false);
        radioButtonVIA.setSelected(false);
    }
});
radioButtonNVIDIA.setOnAction(e -> {
    if (radioButtonNVIDIA.isSelected()) {
        radioButtonATI.setSelected(false); //
        radioButtonINTEL.setSelected(false);
        radioButtonVIA.setSelected(false);
    }
});
radioButtonVIA.setOnAction(e -> {
    if (radioButtonVIA.isSelected()) {
        radioButtonATI.setSelected(false); //
        radioButtonNVIDIA.setSelected(false);
        radioButtonINTEL.setSelected(false);
    }
});
});

List<String> gyartoLista=new ArrayList<>();
gyartoLista.add("Válasszon egy gyártót");
List<String> getGyartoLista = session.createQuery
    (s: "SELECT DISTINCT n.Gyarto FROM Notebook n", String.class).list();
for(var item : getGyartoLista) {
    gyartoLista.add(item.toString());
}
comboGyartok.setItems(FXCollections.observableArrayList(gyartoLista));
comboGyartok.getSelectionModel().selectFirst();

setNotebook2Table(null);
```

Az Olvas2 megjelenését oldalt pedig a menuSearchReadClick eventtel ellátott metódus valósítja meg, amiben megvan valósítva kóddal, hogy egyszerre csak egy rádió gombra lehessen csak rákattintani, illetve itt gyűjti ki a lenyíló listába a gyártók nevét.

ÍR ALMENÜ

ÍR - NÉZET

Az Ír almenü lehetővé teszi a felhasználó számára, hogy az adatbázisban eltárol operációs rendszerek közé, fel tudjon venni egy újat.

Az oldalon megjelenik egy szövegbeviteli mező ahol megtudja adni a nevét az operációs rendszernek, illetve ezt a műveletet megvalósító Hozzáad gomb. Ezek alatt pedig az eltárolt adatbázis jelennek meg.

Id	Név
1	FreeDOS
2	Linux
3	Microsoft Vista Business
4	Microsoft Vista Home Basic HU
5	Microsoft Vista Home Premium
6	Microsoft Vista Home Premium HU
7	Microsoft Vista Home Premium HU notebook
8	nincs
9	Windows 7 Home Premium HU 32bit
10	Windows 7 Home Premium HU 64bit
11	Windows 7 Starter Edition HU
12	Windows XP Home Magyar
13	Windows 11

A megjelenést az menuCreateClick gomb eseménye váltja ki (Ír tabella) amiben a setOsTable metódus, beállítja és előkészíti az operációs rendszerek megjelenítéséhez a táblázatot.

```
public void menuCreateClick(Event event) { 1 usage
    setOsTable(tv3);
    lb3.setText("Összesen: " + (long) tv3.getItems().size() + " db.");
}

public void setOsTable(Table<OS> tv) { 4 usages
    tv.getColumns().clear();
    tv.getItems().clear(); // Előző oszlopok törlése
    idCol = new TableColumn<>(s: "Id");
    nevCol = new TableColumn<>(s: "Név");
    idCol.setCellValueFactory(new PropertyValueFactory<>(s: "Id"));
    nevCol.setCellValueFactory(new PropertyValueFactory<>(s: "Név"));
    tv.getColumns().addAll(idCol, nevCol);
    List<OS> getOsLista = session.createQuery(s: "FROM OS", OS.class).list();
    for(OS os : getOsLista) {
        tv.getItems().add(os);
    }
}
```

ÍR - KONTROLLER

Miután a felhasználó beírta az új operációs rendszer nevét és a Hozzáad gombra kattint, lefut a hozzáadásért felelős Create metódus, feltölti az adatot, a tábla frissül a hozzáadott elemmel és vissza állítja az alapértelmezett nézetet.

```
public void Create(ActionEvent actionEvent) { 1 usage

    Session sessionForNewOs = factory.openSession();

    System.out.println(newOS.getText());
    OS osCreate;
    if(!newOS.getText().trim().isEmpty()) {
        osCreate = new OS(newOS.getText());
        sessionForNewOs.save(osCreate);
        sessionForNewOs.close();
    }
    setOsTable(tv3);

    // Adatok lekérése a Hibernate segítségével
    tv3.getItems().clear();
    setOsTable(tv3);
}
```

MÓDOSÍT ALMENÜ

MÓDOSÍT - NÉZET

A Módosít tabella lenyomásával meghívódik a menuUpdateClick esemény, ami betölti a Módosításhoz kellő űrlapot és a táblázatot, a setOsTable metódussal megjelenítjük a táblákat. A benne lévő getComboOsList pedig hozzáadja szerkesztve az lenyíló listához az operációs rendszereket.

```
public void menuUpdateClick(Event event) { 1 usage
    setOsTable(tv4);
    getComboOsList();
    lb4.setText("Összesen: " + (long) tv4.getItems().size() + " db.");
}

@FXML 2 usages
public void getComboOsList(){
    List<String> osLista = new ArrayList<>();
    List<OS> getOsLista = session.createQuery("s: FROM OS", OS.class).list();
    for(OS os : getOsLista) {
        osLista.add(os.getId() + "." + os.getNev());
    }
    comboOS.setItems(FXCollections.observableArrayList(osLista));
}
```

Notebook Application

JAVA FX Beadandó

Adatbázis Soap Kliens Párhuzamos Forex

Adatbázis menü

Olvas Olvas2 Ír Módosít Töröl

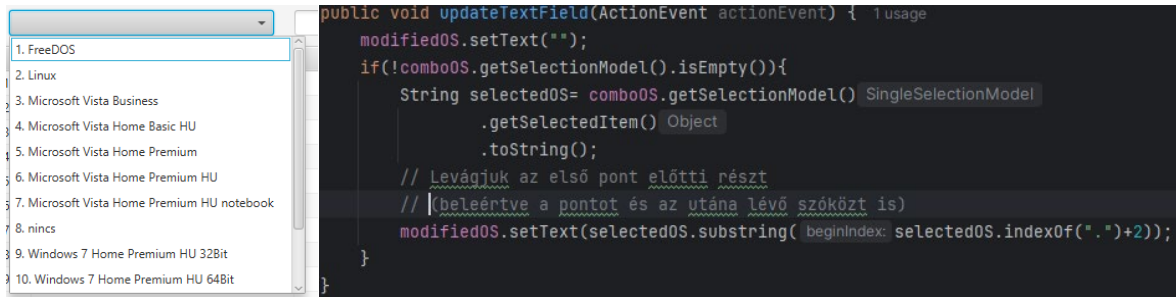
Operációs rendszerek listája Összesen: 13 db.

Kérjük válassza ki melyik Operációs rendszert szeretné módosítani!

Id	Név
1	FreeDOS
2	Linux
3	Microsoft Vista Business
4	Microsoft Vista Home Basic HU
5	Microsoft Vista Home Premium
6	Microsoft Vista Home Premium HU
7	Microsoft Vista Home Premium HU notebook
8	nincs
9	Windows 7 Home Premium HU 32Bit
10	Windows 7 Home Premium HU 64Bit
11	Windows 7 Starter Edition HU
12	Windows XP Home Magyar
13	Windows 11

MÓDOSÍT - KONTROLLER

A legördülő listában azonosítóval ellátva ki lehet választani egy operációs rendszert, amit a kiválasztás időpontjában az updateTextField metódus segítségével bele is tesz a mellette lévő TextField-be, ahol a felhasználó tudja azt módosítani.



Amennyiben bármilyen változás történt az operációs rendszer régi és új neve között, úgy a Módosít gomb lenyomásával megtörténik a módosítás.

```
public void Update(ActionEvent actionEvent) { 1 usage
    if(!modifiedOS.getText().trim().isEmpty() || !modifiedOS.getText()
        .trim()
        .equals(comboOS.getSelectionModel() SingleSelectionModel
            .getSelectedItem() Object
            .toString())){
        Transaction transaction = session.beginTransaction();|
        String selectedOS=comboOS.getSelectionModel().getSelectedItem().toString();
        int dotIndex = selectedOS.indexOf('.');
        int selectedOsIndex = Integer.parseInt(selectedOS.substring(0, dotIndex).trim());

        OS os=session.get(OS.class, selectedOsIndex);
        os.setNev(modifiedOS.getText().trim());
        session.update(os);
        tv4.refresh();
        comboOS.getSelectionModel().clearSelection();
        getComboOsList();

        modifiedOS.setText("");

        transaction.commit();
    }
}
```

TÖRÖL ALMENÜ

TÖRÖL - NÉZET

A Töröl tabella megnyomásával meghívódik a menuDeleteClick metódus, ami a setNotebookTable metódussal létrehozza a táblázatot, a getNotebookList metódussal pedig feltölti a lenyíló listát az összes Notebookkal, az azonosítójával ellátva.

```
public void menuDeleteClick(Event event) { 2 usages
    setNotebookTable(tv5);
    getComboNotebookList();
}
```

```
@FXML 1 usage
public void getComboNotebookList(){

    List<String> notebookLista=new ArrayList<>();
    List<Notebook>getNotebookLista = session.createQuery(s: "FROM Notebook", Notebook.class).list();
    for(Notebook nb : getNotebookLista) {
        notebookLista.add(nb.getId()+" "+nb.getGyarto()+" "+nb.getTipus());
    }

    comboNotebook.setItems(FXCollections.observableArrayList(notebookLista));
    lb5.setText("Az adatbázisban található összes notebook. Összesen: "+(long) tv5.getItems().size()+" db.");
}
```

Notebook Application

JAVA FX Beadandó

Adatbázis Soap Kliens Párhuzamos Forex

Adatbázis menü

Olvas Olvas2 Ír Módosít Töröl

Operációs rendszerek listája Az adatbázisban található összes notebook. Összesen: 246 db.

Válassza ki a törölni kívánt Operációs rendszert!

1. HP COMPAQ 615 NX556EA

Törölés

Id	Memória	Merevlemez	Videóvezérlő	Ár	Processzor	Operációs rendszer	DB
1	1024	160	ATI Mobility Radeon HD3200 256MB	95120	AMD Athlon 64 X2 QL64	FreeDOS	0
2	2048	250	ATI Mobility Radeon HD3200 256MB	101200	AMD Athlon 64 X2 QL64	nincs	0
3	2048	320	ATI Mobility Radeon HD3200 256MB	114800	AMD Athlon 64 X2 QL64	Microsoft Vista Home Basic HU	0
4	3072	250	ATI Mobility Radeon HD4530 512MB	167920	AMD Athlon 64 X2 QL64	Microsoft Vista Home Premium HU	3
5	2048	250	ATI Mobility Radeon HD4570 512MB	111920	AMD Athlon 64 X2 QL64	Linux	3
6	3072	320	ATI Mobility Radeon HD4570 512MB	117520	AMD Athlon 64 X2 QL64	Linux	2
7	2048	320	ATI Radeon Xpress 1250	111920	AMD Athlon TM Neo MV-40	Microsoft Vista Home Premium HU	4
8	4096	500	ATI Mobility Radeon HD4570 512MB	115920	AMD Athlon TM Neo MV-40	nincs	1
9	2048	320	NVIDIA GeForce Go 9100M-GS	90800	AMD Mobil Sempron SI-40	FreeDOS	1
10	4096	500	ATI Mobility Radeon HD4670 512MB	183920	AMD Turion64 X2 TL60	Windows 7 Home Premium HU 64Bit	4
11	4096	500	ATI Mobility Radeon HD4570 512MB	134320	AMD Turion64 X2 TL64	nincs	5
12	2048	250	ATI Mobility Radeon HD3200 256MB	223920	AMD Turion64 X2 TL62	Microsoft Vista Home Premium HU	4
13	2048	320	Intel Graphics X4500M / 256MB	112400	Intel Celeron M ULV723	Microsoft Vista Home Premium HU	3
14	1024	160	Intel Graphics 4500MHD	82800	Intel Celeron 900	Linux	0
15	1024	160	Intel Graphics 4500MHD	103120	Intel Celeron 900	Linux	3
16	2048	320	Intel Graphics X4500M / 256MB	118800	Intel Celeron 900	FreeDOS	1
17	2048	320	NVIDIA GeForce Go 8200M 128MB	94800	Intel Celeron 900	FreeDOS	1

TÖRÖL - KONTROLLER

Amennyiben a felhasználó kiválasztott egy számára tetszőleges Notebookot a lenyíló listából és rányom a Töröl gombra, meghívódik a Delet metódus, ami kitörli az adatbázisból a kiválasztott Notebookot. Miután ez megtörtént, vissza állítja a nézetet a kiindulási helyzetébe, frissítve a listát.

```
public void Delete(ActionEvent actionEvent) { 1 usage
    if(!comboNotebook.getSelectionModel().isEmpty()){
        Transaction transaction = session.beginTransaction();
        String selectedNotebook=comboNotebook.getSelectionModel().getSelectedItem().toString();
        int dotIndex = selectedNotebook.indexOf('.');
        int selectedNotebookIndex = Integer.parseInt(selectedNotebook.substring(0, dotIndex).trim());

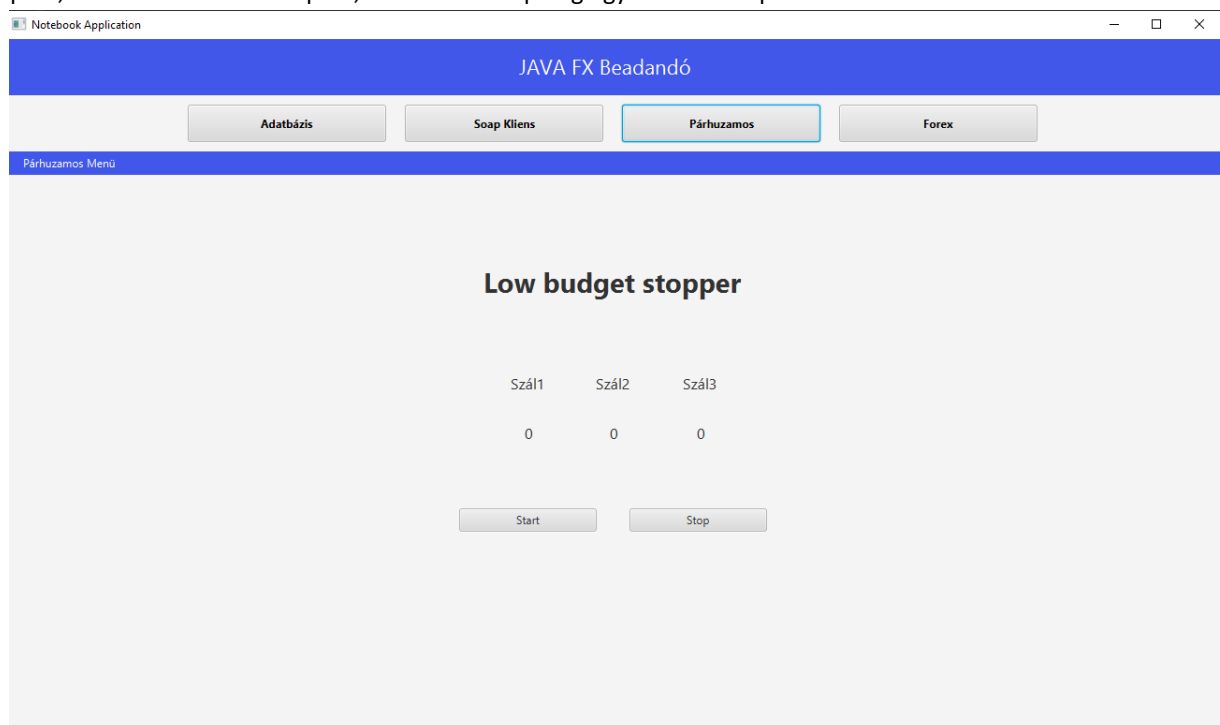
        Notebook nb=session.get(Notebook.class, selectedNotebookIndex);
        session.delete(nb);

        comboNotebook.getSelectionModel().clearSelection();
        transaction.commit();

        menuDeleteClick(actionEvent);
    }
}
```

PÁRHUZAMOS MENÜ

Ezen az oldalon a feladatkiírás érdekében a Párhuzamos gomb megnyomására megjelenik egy száakezelést használó nézet (Parhuzamos_view), amiben három szá fut egyszerre, reprezentálva egy stoppert. Első szá a perc, Második szá a másodperc, Harmadik szá pedig egy tizedmásodperc számláló.



A hozzá tartozó kontroller osztályban (ThreadsController) létrehoztunk 1-1 külön osztályt a három szálnak, amik kiegészülnek a Thread Runnable osztállyal (ez teszi lehetővé a szálkezelést), amiben az első szál percenként (60000ms), a második másodpercenként (1000ms), a harmadik pedig tízed másodpercenként (100ms) frissül.

```
class Thread1 extends Thread { 2 usages
    @Override
    public void run() {
        while (running) {
            String message1 = counter1 + " (min)";
            Platform.runLater(() -> thread_one_txt.setText(message1));
            counter1 = (counter1 == 59) ? 0 : counter1 + 1;
            try {
                Thread.sleep(millis: 60000);
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
                break;
            }
        }
    }
}
```

```
class Thread2 extends Thread { 2 usages
    @Override
    public void run() {
        while (running) {
            String message2 = counter2 + " (sec)";
            Platform.runLater(() -> thread_two_txt.setText(message2));
            counter2 = (counter2 == 59) ? 0 : counter2 + 1;
            try {
                Thread.sleep(millis: 1000);
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
                break;
            }
        }
    }
}
```

```
class Thread3 extends Thread { 2 usages
    @Override
    public void run() {
        while (running) {
            String message3 = "0." + counter3 + " (sec)";
            Platform.runLater(() -> thread_three_txt.setText(message3));
            counter3 = (counter3 == 10) ? 1 : counter3 + 1;
            try {
                Thread.sleep(millis: 100);
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
                break;
            }
        }
    }
}
```

A Start gomb lenyomásával példányosítjuk a szálak osztályait, és a start() metódussal elindítjuk a szálkezelést.

```
@FXML
private void onStartButtonClick() {
    if (!running) {
        running = true;
        thread1 = new Thread1();
        thread2 = new Thread2();
        thread3 = new Thread3();

        thread1.start();
        thread2.start();
        thread3.start();
    }
}
```

Amennyiben le akarjuk állítani úgy a Stop gomb megnyomásával leállítja a szálkezeléseket és vissza állít mindent az alapértelmezett helyzetére.

```
@FXML 1 usage
public void onStopButtonClick(ActionEvent actionEvent) {
    running = false;
    if (thread1 != null) thread1.interrupt();
    if (thread2 != null) thread2.interrupt();
    if (thread3 != null) thread3.interrupt();
    counter1 = 0;
    counter2 = 0;
    counter3 = 0;
    Platform.runLater(() -> {
        thread_one_txt.setText("0 (min)");
        thread_two_txt.setText("0 (sec)");
        thread_three_txt.setText("0 (ms)");
    });
}
```


FOREX MENÜ

OANDA API

Az Oanda egy Forex devizapiacra fejlesztett szolgáltatás, amely API-t biztosít az alkalmazások fejlesztéséhez. Az alkalmazásunkhoz szükséges az Oanda API integrációja, amelyhez letölthető az oanda.v20 API csomag az Oanda weboldaláról. Ezt a csomagot beillesztve a programba, elérhetővé válnak azok az osztályok, amelyek segítségével hozzáférhetünk az adatokhoz.

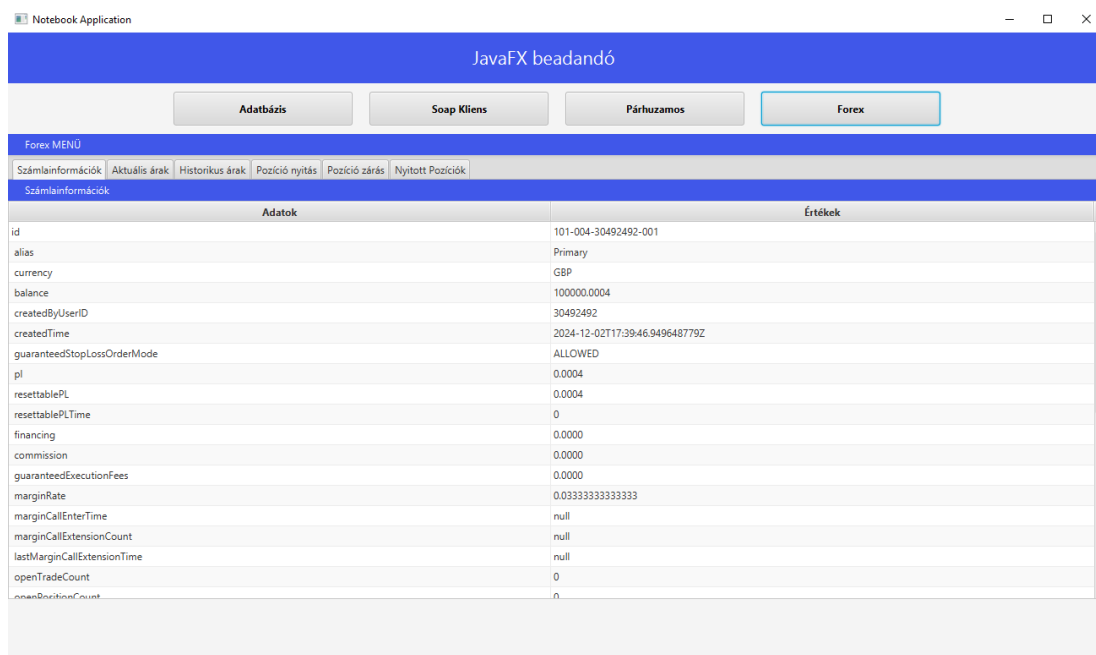
Az API használatához először regisztrálnunk kell egy felhasználót az Oanda platformján, hogy API kulcsot kapjunk. Ezt a kulcsot az alkalmazásunk ForexConfig osztályában tároljuk statikus adatként, ami megkönnyíti az elérését más osztályokból. Ha a kulcsot módosítani kell, elegendő csak ezt a fájlt frissíteni.

```
package com.notebook;  
  
import com.oanda.v20.account.AccountID;  
  
public class ForexConfig {  
    private ForexConfig() {} no usages  
    public static final String URL = "https://api-fxpractice.oanda.com"; 1 usage  
    public static final String TOKEN = "7bc36878e2ac6cd30101020044385ce2-a9f225209e219215e9d74dc0278c07d0"; 1 usage  
    public static final AccountID ACCOUNTID = new AccountID("101-004-30492492-001"); 2 usages  
}
```

A Forex gomb kattintásával nem egy üres tér jelenik meg, hanem egyből lefut a lentebb említett Számlainformációk almenü funkciója.

SZÁMLAINFORMÁCIÓK ALMENÜ

Ebben a feladatban a felhasználónk számlainformációinak a részleteit tudjuk kilistázni egy 2 oszlopos táblázatos formában, egyik oszlopban a nevei, másikban pedig az értékei jelennek meg.

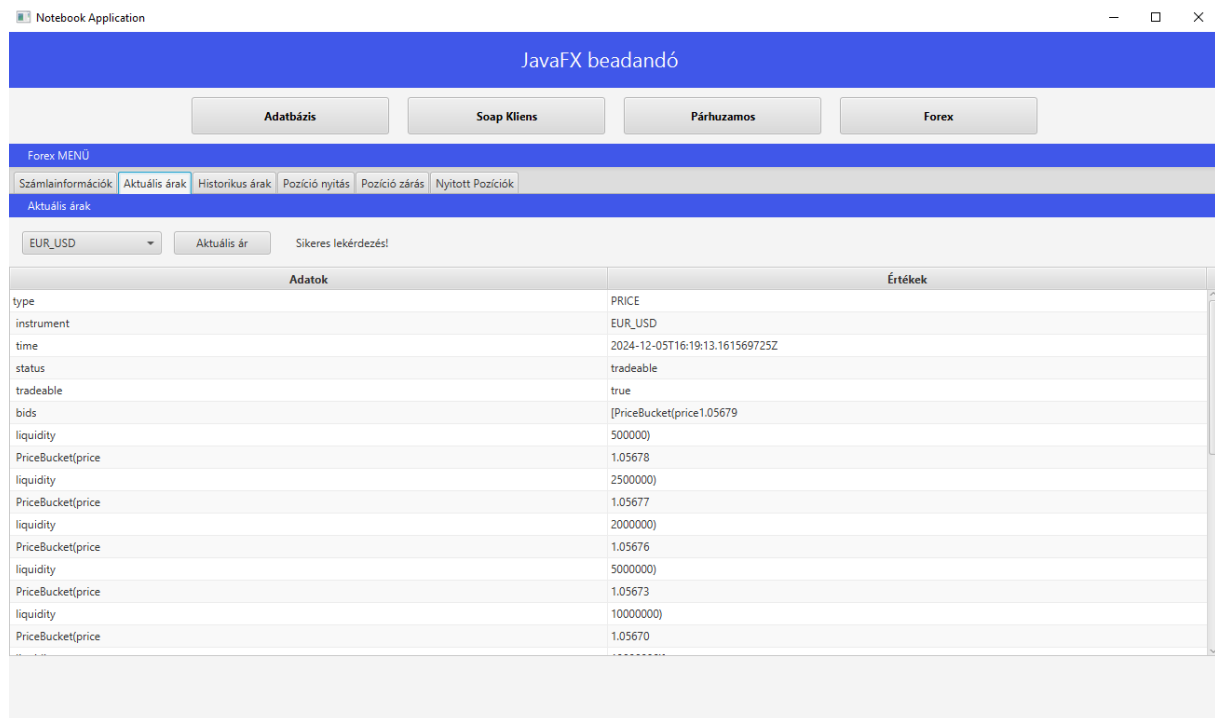


The screenshot shows a JavaFX application window titled "JavaFX beadandó". It has four buttons: "Adatbázis", "Soap Kliens", "Párhuzamos", and "Forex". The "Forex" button is selected. Below the buttons is a "Forex MENÜ" section with a tabbed interface. The "Számlainformációk" tab is active, showing a table of account information.

Adatok	Értékek
id	101-004-30492492-001
alias	Primary
currency	GBP
balance	100000.0004
createdByUserID	30492492
createdTime	2024-12-02T17:39:46.949648779Z
guaranteedStopLossOrderMode	ALLOWED
pl	0.0004
resettablePL	0.0004
resettablePLTime	0
financing	0.0000
commission	0.0000
guaranteedExecutionFees	0.0000
marginRate	0.03333333333333333
marginCallEnterTime	null
marginCallExtensionCount	null
lastMarginCallExtensionTime	null
openTradeCount	0
marginBalanceCount	0

AKTUÁLIS ÁRAK ALMENÜ

Ebben a feladatban egy lenyíló lista segítségével devizapárokat tudunk kiválasztani, az Aktuális ár gombbal pedig lekérni a kiválasztott devizapár információit és az aktuális árát. Ezeket mind egy két oszlopos táblában jeleníti meg.



HISTORIKUS ÁRAK

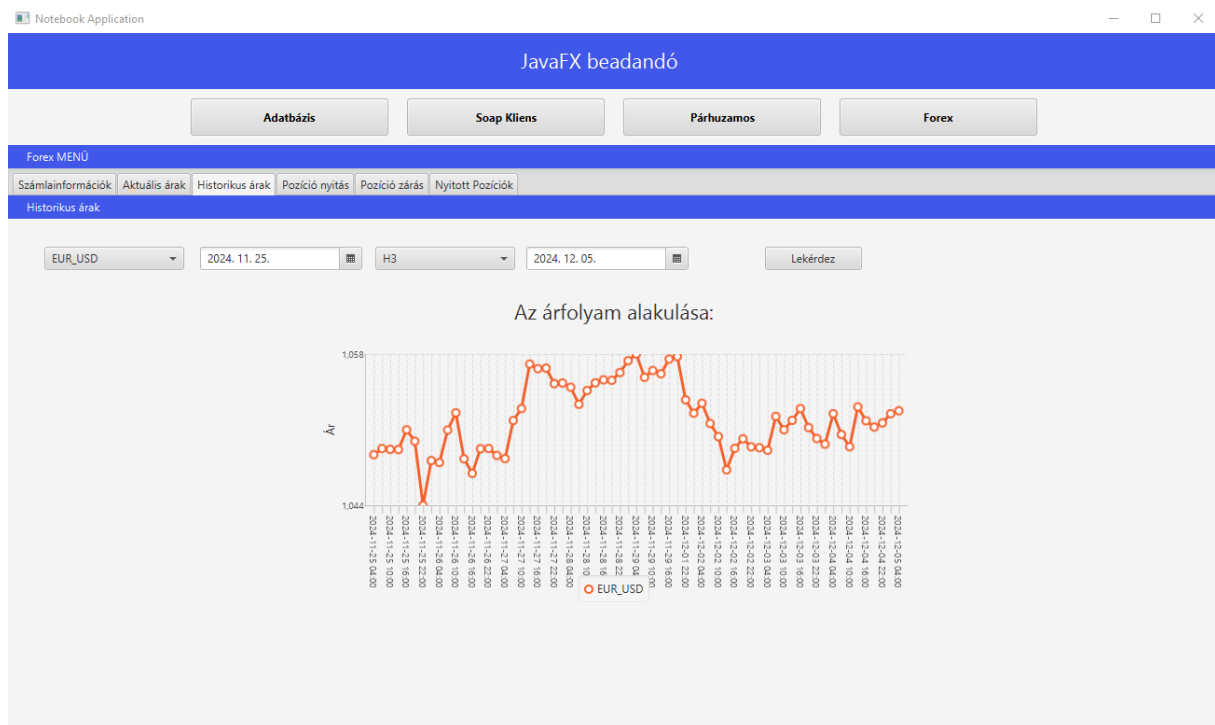
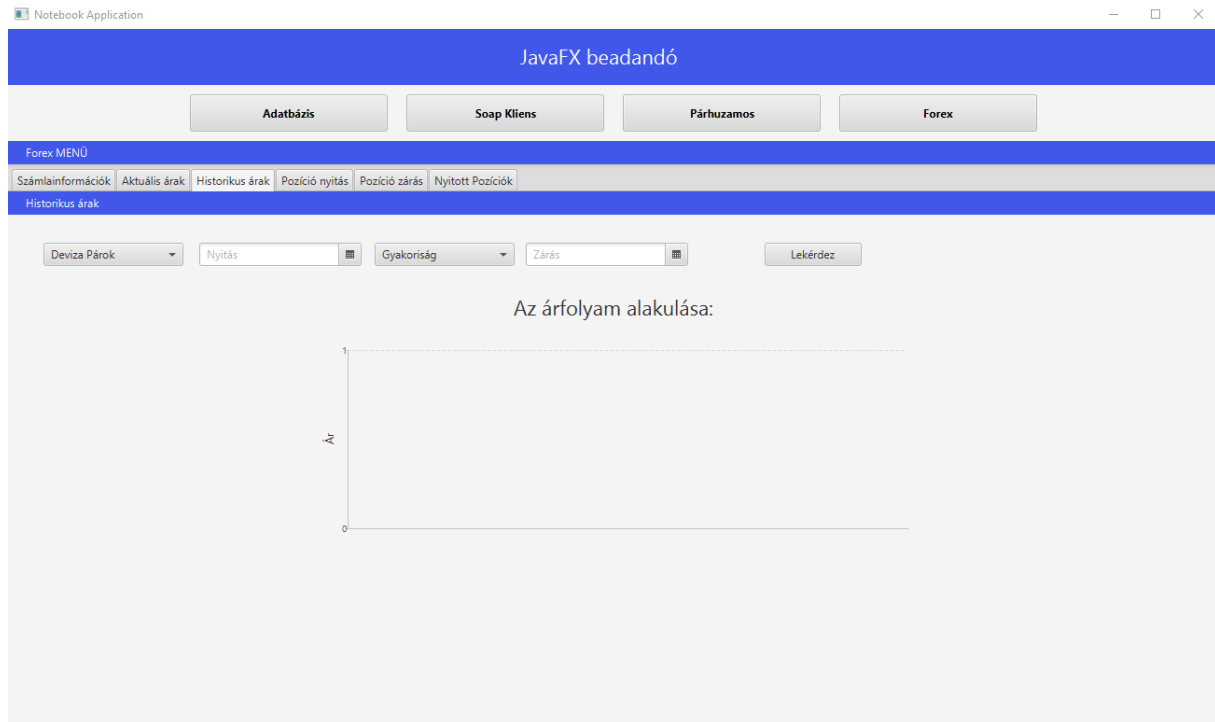
Ebben a menüpontban, ha kiválasztanunk egy devizapárt, egy mérési gyakoriságot, valamint két időpontot (nyitó- és záró dátumot), az alkalmazás ezek alapján elkészít egy diagramot, amely az árfolyam változását ábrázolja a megadott időszakban.

A devizapárok és a mérési gyakoriságok egy ComboBox segítségével választhatók ki, míg az időpontokat két DatePicker vezérlővel adhatjuk meg. A kiválasztott dátumok formátuma az alábbi: yyyy-MM-ddT00:00:00. Az árfolyam változása a diagram Y tengelyén jelenik meg, a minimum és maximum értékek között. Az X tengelyen pedig az adott időpontok láthatók.

Fontos megjegyezni, hogy az alkalmazás nem képes túl nagy sűrűségű adatokat megjeleníteni, ezért a túl gyakori mérési intervallumokat kerülni kell.

A Lekérdez gomb nyomására történik meg a lekérdezés és a diagram készítése.

Ez egy időigényesebb procedúra, így lehet, hogy emiatt **az alkalmazás egy ideig nem fog válaszolni!**



POZÍCIÓ NYITÁS

Az alkalmazás lehetőséget biztosít új pozíciók nyitására. Ehhez ki kell választanunk egy devizapárt, meg kell adnunk a mennyiséget, valamint az irányt (Eladás vagy Vásárlás). A Pozíció megnyitása gomb megnyomásával hozhatjuk létre a pozíciót.

A program először ellenőrzi, hogy minden szükséges adat helyesen meg lett-e adva, és csak ezután engedélyezi a pozíció megnyitását. Fontos megjegyezni, hogy hétvégén a piac nem elérhető, így új pozíciókat sem lehet nyitni. Ha ilyenkor próbálkozunk, a rendszer figyelmeztetést ad.

Sikeres pozíció nyitás esetén a program visszajelzést küld, és megjeleníti a létrehozott pozíció azonosítóját.

JavaFX beadandó

Adatbázis Soap Kliens Párhuzamos Forex

Forex MENÜ

Számlainformációk Aktuális árak Historikus árak Pozíció nyitás Pozíció zárás Nyitott Pozíciók

Pozíció nyitás

Deviza Párok Mennyiség Irány Pozíció megnyitása

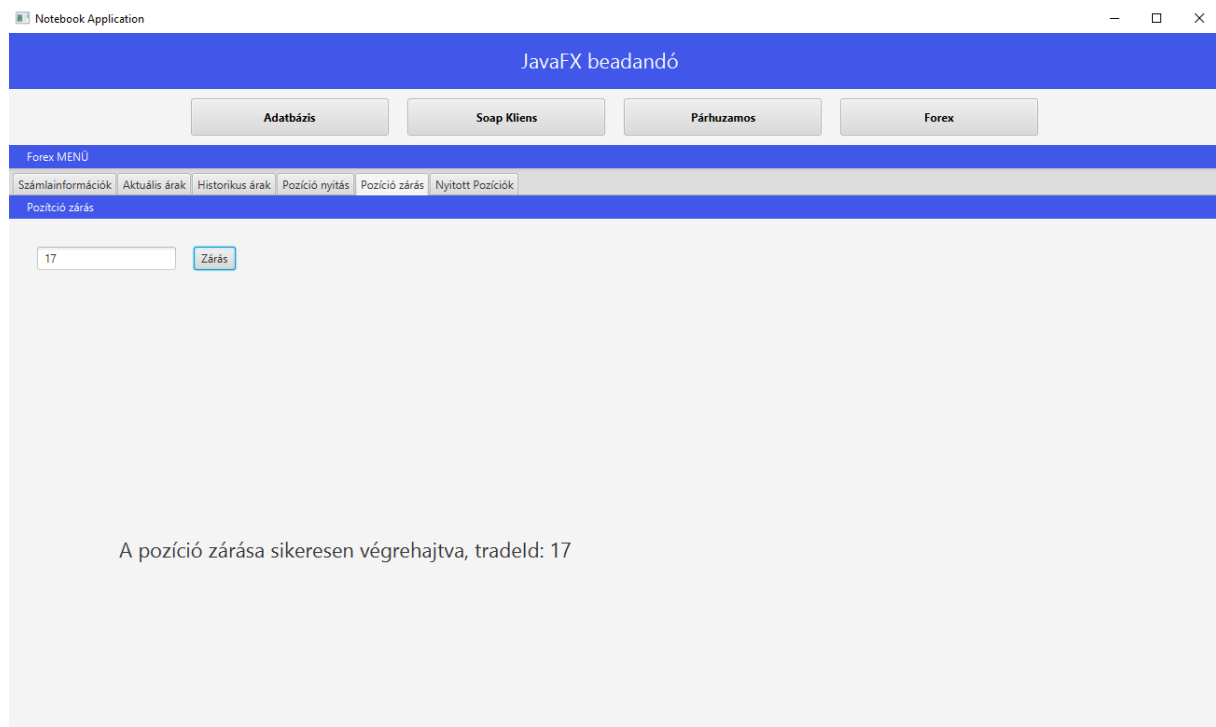
EUR_USD 20 Eladás Pozíció megnyitása

Sikeres nyitottál egy pozíciót tradeld: 17

POZÍCIÓ ZÁRÁS

Az alkalmazás egy egyszerű felületet biztosít, amelyen egy TextField és egy gomb található. A TextField mezőbe adhatjuk meg a törlendő pozíció azonosítóját, amely kizárólag számokat tartalmazhat.

A gomb megnyomása után a rendszer végrehajtja a törlési műveletet, majd visszajelzést ad arról, hogy a törlés sikeres volt-e.



NYITOTT POZÍCIÓK

A létrehozott pozíciók ebben menüpontban tekinthetők meg. A megjelenített táblázat hatféle adatot tartalmaz minden pozícióról: tradeID, Valutapár, Nyitás ideje, Current Units, Ár, és UnrealizedPL.

Ez a táblázat átlátható módon nyújt információt a nyitott pozíciókról, beleértve azok azonosítóját (tradeID), amely alapján a pozíciók törölhetők.

[illegible]

FOREX KÓD, AZAZ A MÖGÖTTES TARTALOM

A Forex menü vizuális elemeit a FeladatController.java tölti be, a main-view.fxml fájl ContentView ID-jú AnchorPane elemébe. A teljes Forex menü grafikus felépítése a Forex_main.fxml fájlban található, amely a ForexController.java osztályhoz kapcsolódik.

Az Oanda API-hoz való csatlakozáshoz szükséges adatokat a `ForexConfig.java` osztály tárolja. A Controller ebből az osztályból éri el az adatokat, hogy létrehozza a Context (cxt) és AccountID (accountID) példányokat, amelyek biztosítják az adatokhoz való hozzáférést.

```
Context ctx = new ContextBuilder(ForexConfig.URL).setToken(ForexConfig.TOKEN).setApplication("PricePolling").build();
AccountID accountId = ForexConfig.ACCOUNTID; 4 usages
```

A Controller osztályban definiálásra kerülnek azok a JavaFX komponensek, amelyeket a program során a ForexMenü működtetéséhez használunk. Ide tartoznak például a táblázatok, táblázatoszlopok, lenyíló menük és egyéb vizuális elemek, amelyek biztosítják a menü funkcionalitását és felhasználói interakcióját.

```

@FXML
private TableView<ForexData> AccountDataTableView;
@FXML
private TableColumn<ForexData, String> AccountNameColumn;
@FXML
private TableColumn<ForexData, String> AccountValueColumn;
//Aktualis arak
@FXML
private ComboBox<String> InstrumentsComboBox;
@FXML
private TableView<ForexData> InstrumentInformationsTableView;
@FXML
private TableColumn<ForexData, String> InstrumentDataNameColumn;
@FXML
private TableColumn<ForexData, String> instrumentDataValueColumn;
@FXML
private Label InstrumentResponseLabel;
//Historikus arak
@FXML
private ComboBox<String> HistoricInstrumentsComboBox;
@FXML
private ComboBox<CandlestickGranularity> HistoricGranularityComboBox;
@FXML
private DatePicker StartDatePicker;
@FXML
private DatePicker CloseDatePicker;
@FXML
private Label HistoricResponseLabel;
@FXML
private LineChart ForexLineChart;
@FXML
private NumberAxis ForexYAxis;

```

```

//Pozicio Nyitas
@FXML
private ComboBox<String> OpenPositionInstrumentsComboBox;
@FXML
private ComboBox<Integer> OpenPositionAmountComboBox;
@FXML
private ComboBox<String> OpenPositonDirrectionComboBox;
@FXML
private Label OpenPositionInstrumentsErrorLabel;
@FXML
private Label OpenPositionAmountErrorLabel;
@FXML
private Label OpenPositionDirrectionErrorLabel;
@FXML
private Label OpenPositionResponseLabel;
//Pozicio zaras
@FXML
private TextField ClosePostionIdTextField;
@FXML
private Label ClosePostionIdErrorLabel;
@FXML
private Label ClosePostionResponseLabel;

```

```

//Nyitott Poziciok

```

```

@FXML
private TableView<Trade> TradesTableView;
@FXML
private TableColumn<Trade, String> TradeIdTableColumn;
@FXML
private TableColumn<Trade, String> TradeInstrumentTableColumn;
@FXML
private TableColumn<Trade, String> TradeDateTableColumn;
@FXML
private TableColumn<Trade, String> TradeCurrentUnitsTableColumn;
@FXML
private TableColumn<Trade, String> TradePriceTableColumn;
@FXML
private TableColumn<Trade, String> TradeUnrealizedPLTableColumn;

```

Az almenü funkcióért pedig az alábbi a ForexController-ben található függvények felelnek

```
public void loadAccountInformations() {...}

public void loadInstrumentInformations() {...}

public void loadHistoric(){...}

public void openPositon() {...}

public void closePosition(){...}

public void listPostions(){...}
```

MODELLEK

CPU MODELL

Az osztály egy **CPU** (processzor) nevű entitást definiál, amelyet egy adatbázisban tárolnak. Az osztály a **JPA (Java Persistence API)** segítségével van megjelölve entitásként, amely lehetővé teszi, hogy közvetlenül kapcsolódjon egy adatbázis megfelelő táblájához. Az osztályhoz tartozó adatbázis tábla neve "**processzor**", amit a `@Table(name = "processzor")` annotáció határoz meg.

ID (INT TÍPUSÚ)

Ez az osztály egyedi azonosítója, amely az adatbázisban az **id** nevű oszlophoz kapcsolódik. Az azonosító értékét automatikusan generálja az adatbázis az **IDENTITY** generációs stratégia szerint (`@GeneratedValue(strategy = GenerationType.IDENTITY)`).

GYARTO (STRING TÍPUSÚ)

A processzor gyártóját tárolja, például Intel vagy AMD. Ez az adat az adatbázis **gyarto** nevű oszlopához kapcsolódik.

TIPUS (STRING TÍPUSÚ)

A processzor típusát (modelljét) tartalmazza, például "Core i7" vagy "Ryzen 5". Ez az adat az adatbázis **típus** nevű oszlopához kapcsolódik. Az osztály a következő elemeket tartalmazza:

KONSTRUKTOROK

Egy alapértelmezett konstruktor (`public CPU()`), amely lehetővé teszi, hogy példányt hozzunk létre anélkül, hogy értékeket adnánk meg. Egy paraméterezett konstruktor (`public CPU(Integer id, String gyarto, String tipus)`), amely lehetővé teszi, hogy azonosítóval, gyártóval és típussal együtt hozzunk létre egy új CPU példányt.

GETTER ÉS SETTER METÓDUSOK

Minden mezőhöz tartozik getter és setter, amelyek lehetővé teszik az értékek lekérdezését és beállítását. Ezek:

getId() és **setId(Integer id)** az azonosító kezeléséhez.

getGyarto() és **setGyarto(String gyarto)** a gyártó kezeléséhez.

getTipus() és **setTipus(String tipus)** a típus kezeléséhez.

NOTEBOOK MODELL

Az osztály egy **Notebook** nevű entitást definiál, amelyet egy adatbázisban tárolnak. Ez az osztály a **JPA (Java Persistence API)** segítségével kapcsolódik az adatbázis megfelelő táblájához. Az entitás az adatbázisban a **"gep"** nevű táblához tartozik, amit a `@Table(name = "gep")` annotáció jelez.

ID (INT TÍPUSÚ)

Az osztály egyedi azonosítója, amelyet az adatbázis generál az IDENTITY stratégia szerint. Az adatbázisban az id nevű oszlophoz kapcsolódik.

GYARTO (STRING TÍPUSÚ)

A notebook gyártóját tárolja, például Dell, HP, vagy Lenovo. Az adat az adatbázis gyarto nevű oszlopában található.

TIPUS (STRING TÍPUSÚ)

A notebook modelljét tárolja a "XPS 13" vagy "ThinkPad T14" modelleket. Az adat az adatbázis tipus nevű oszlopában található.

KIJELZO (FLOAT TÍPUSÚ)

A kijelző méretét (hüvelykben) tárolja. Az adat az adatbázis kijelzo nevű oszlopához kapcsolódik.

MEMORIA (INT TÍPUSÚ)

A notebook RAM memóriáját gigabájtban tárolja. Az adat az adatbázis memoria nevű oszlopához kapcsolódik.

MEREVLEMEZ (INT TÍPUSÚ)

A notebook merevlemezének méretét gigabájtban tárolja. Az adat az adatbázis merevlemez nevű oszlopához kapcsolódik.

VIDEOVEZERLO (STRING TÍPUSÚ)

A videovezérlő (GPU) típusát tárolja a "NVIDIA GeForce" vagy "Intel Iris" típusokat is. Az adat az adatbázis videovezerlo nevű oszlopához kapcsolódik.

AR (INT TÍPUSÚ)

A notebook árát tárolja forintban vagy más pénznemben. Az adat az adatbázis ar nevű oszlopában található.

DB (INT TÍPUSÚ)

A notebook készletének darabszámát tárolja. Az adat az adatbázis db nevű oszlopában található.

PROCESSZOR (CPU)

Kapcsolatot hoz létre a notebook és a CPU entitás között. Az adatbázisban a processzorid oszlop tárolja a kapcsolatot. Ez a kapcsolat a @ManyToOne annotációval van megjelölve.

OPERÁCIÓS RENDSZER (OS)

Kapcsolatot hoz létre a notebook és az OS (operációs rendszer) entitás között. Az adatbázisban az oprendszerid oszlop tárolja a kapcsolatot. Ez a kapcsolat szintén a @ManyToOne annotációval van megjelölve.

AZ OSZTÁLY FUNKCIÓI:

Konstruktorok

Alapértelmezett konstruktor(public Notebook()):

Egy üres konstruktor, amely lehetővé teszi az objektum létrehozását értékek megadása nélkül.

Paraméterezett konstruktor(public Notebook(String gyarto, String tipus, float kijelzo, Int memoria, Int merevlemez, String videovezerlo, Int ar, Int db, CPU processzor, OS operaciosRendszer)):

Ez a konstruktor lehetővé teszi az összes mező inicializálását, beleértve a kapcsolódó CPU és OS objektumokat.

Getter és Setter metódusok

Az összes mezőhöz tartozik getter és setter, amelyek lehetővé teszik az értékek lekérdezését és beállítását.

Ezek például:

getId() és **setId(Integer id)** az azonosító kezelésére.

getGyarto() és **setGyarto(String gyarto)** a gyártó lekérdezésére és módosítására.

SPECIÁLIS METÓDUSOK

getProcesszorNev():

Visszaadja a processzor nevét a gyártó és a típus összefűzésével. Ha a processzor értéke null, akkor "N/A" szöveget ad vissza.

getOperaciosRendszerNev():

Visszaadja az operációs rendszer nevét. Ha az operációs rendszer nincs megadva, akkor "N/A" szöveget ad vissza.

OS MODELL

Az osztály egy **OS** (operációs rendszer) nevű entitást definiál, amelyet egy adatbázisban tárolnak. Ez az osztály a **JPA (Java Persistence API)** segítségével van megjelölve, és az **"oprendszer"** nevű adatbázis táblához kapcsolódik, amit a @Table(name = "oprendszer") annotáció határoz meg.

ID (INT TÍPUSÚ)

- Az osztály egyedi azonosítója, amelyet az adatbázis automatikusan generál az **IDENTITY** stratégia szerint.
- Az adatbázisban az **id** nevű oszlophoz kapcsolódik.

NEV (STRING TÍPUSÚ)

- Az operációs rendszer nevét tárolja, például "Windows 10", "Ubuntu 20.04", vagy "macOS Monterey".
- Az adat az adatbázis **nev** nevű oszlopához kapcsolódik.

AZ OSZTÁLY FUNKCIÓI:

Konstruktorok

Alapértelmezett konstruktor (**public OS()**):

Egy üres konstruktor, amely lehetővé teszi egy üres objektum létrehozását.

Paraméterezett konstruktor (**public OS(String nev)**):

Ez a konstruktor lehetővé teszi egy új operációs rendszer példány létrehozását egy név megadásával.

Getter és Setter metódusok:

getId() és **setId(int id)**:

Ezekkel az azonosító értéke lekérdezhető és módosítható.

getNev() és **setNev(String nev)**:

Ezek a metódusok az operációs rendszer nevének lekérdezésére és módosítására szolgálnak.

HIBERNATE.CFG.XML

KAPCSOLÓDÁS AZ SQLITE ADATBÁZISHOZ:

A Hibernate a feladat.db fájlhoz csatlakozik, ami a projekt főkönyvtárával egy helyen van, illetve ha .jar készült, a jar fájlal egy mappában.

ENTITÁSOK KEZELÉSÉNEK ENGEDÉLYEZÉSE:

Az alábbi entitások kerülnek regisztrálásra:

- **OS**: Az operációs rendszer entitás (oprendszer tábla).
- **Notebook**: A notebook entitás (gep tábla).
- **CPU**: A processzor entitás (processzor tábla).

MAPPINGEK BETÖLTÉSE:

A Hibernate az adatbázis táblákat az osztályokhoz (és annotációikhoz) igazítja, automatikusan kezelve az adatbázis-műveleteket, például beszúrást, törlést és frissítést.

FELADATMEGOLDÁSBAN HASZNÁLT GUI ELEMEL

```
@FXML public Label lb1,lb2,lb3,lb4,lb5, radioLabel, updateLabel, createLabel, deleteLabel; 1 usage
@FXML public VBox Read2, updateOS, createOS, deleteNotebook; 3 usages
@FXML public TextField searchField, modifiedOS, newOS; 3 usages
@FXML public ComboBox comboGyartok; 6 usages
@FXML public RadioButton radioButtonINTEL, radioButtonATI, radioButtonNVIDIA, radioButtonVIA; 8 usages
@FXML public CheckBox checkBox; 3 usages
@FXML public Button searchButton; 1 usage
@FXML public ComboBox comboOS, comboNotebook; 7 usages
@FXML public AnchorPane readNotebook; 1 usage

@FXML private TableView<Notebook> tv1, tv2, tv5;
@FXML private TableColumn<Notebook, Integer> IDCol; 6 usages
@FXML private TableColumn<Notebook, String> gyartoCol; 6 usages
@FXML private TableColumn<Notebook, String> tipusCol; 6 usages
@FXML private TableColumn<Notebook, Float> kijelzoCol; 6 usages
@FXML private TableColumn<Notebook, Integer> memoriaCol; 6 usages
@FXML private TableColumn<Notebook, Integer> merevlemezCol; 6 usages
@FXML private TableColumn<Notebook, String> videovezerloCol; 6 usages
@FXML private TableColumn<Notebook, Integer> arCol; 6 usages
@FXML private TableColumn<Notebook, Integer> processzorCol; 6 usages
@FXML private TableColumn<Notebook, Integer> oprendszerCol; 6 usages
@FXML private TableColumn<Notebook, Integer> dbCol; 6 usages

@FXML private TableView<OS> tv3,tv4;
@FXML private TableColumn<OS, Integer> idCol; 3 usages
@FXML private TableColumn<OS, String> nevCol; 3 usages
```