**LibraryAdmin**

**(** fejleszői dokumentáció **)**

2021.05.

**Készítő: Frantal Attila**

Contents

[A projekt indításának a célja: 3](#_Toc72615383)

[A szoftver licence: 3](#_Toc72615384)

[A szoftver kialakításának és céljának fő irányelvei: 3](#_Toc72615385)

[Stage RUP 1 ( Követelmények ) 4](#_Toc72615386)

[Követelményfeltárás: 4](#_Toc72615387)

[Nem funkcionális követelmények: 4](#_Toc72615388)

[Funkcionális követelmények: 4](#_Toc72615389)

[Megszorítások: 4](#_Toc72615390)

[stage RUP 2 ( Tervezés ) 5](#_Toc72615391)

[UML tervek: 5](#_Toc72615392)

[MVC model: 5](#_Toc72615393)

[Control csomag: 5](#_Toc72615394)

[Model csomag: 6](#_Toc72615395)

[Adatbázis: 6](#_Toc72615396)

[View csomag: 7](#_Toc72615397)

[GUI terv: 7](#_Toc72615398)

[stage RUP 3 ( Implementálás ) 8](#_Toc72615399)

[ConnectDatabase osztály 8](#_Toc72615400)

[LoginFrame osztály (extends javax.swing.JFrame) 9](#_Toc72615401)

[MainFrame osztály (extends javax.swing.JFrame) 9](#_Toc72615402)

[SqlRunner osztály 10](#_Toc72615403)

[Client osztály 12](#_Toc72615404)

[Book osztály 12](#_Toc72615405)

[Tran osztály 13](#_Toc72615406)

[DatabaseControl osztály 13](#_Toc72615407)

[GuiControl 15](#_Toc72615408)

[MainControl oszály 16](#_Toc72615409)

[Stage RUP4 16](#_Toc72615410)

[Stage RUP5 16](#_Toc72615411)

[Stage RUP6 17](#_Toc72615412)

# A projekt indításának a célja:

Egy könyvtárban a könyvek kölcsönzésének (és visszavételének) folyamatát hivatott modellezni a program. Nyilvántartjuk a könyvtártagok adatait és könyv tartozásait, a könyvek adatait és a könyvtárban lévő mennyiségüket. Tároljuk a kölcsönzés és visszavétel tényét időbélyeggel, és a tranzakcióhoz tartozó tagi és könyv adatokkal. Külön táblában megjelenítjük a könyvtartozással rendelkezőket.

# A szoftver licence:

- A program kizárólag a készítő tulajdona, tovább értékesítés nem engedélyezett.

- A program nem megvásárolható.

- Kizárólag a készítő iskolai órán elsajátított tudásának ellenőrzésére szolgál.

- Próbaverzió, javaslat esetén a készítő módosíthatja.

# A szoftver kialakításának és céljának fő irányelvei:

1. A szoftver kizárólag szabad felhasználású proramokkal készült. (JAVA JDK, NetBeans, XAMPP, MariaDB, DIA)
2. A program futtatható a következő operációs rendszereken: Windows 7,8,XP,10, MAC IOS, Linux.
3. A program JAVA 14.0.2 verzióval készült.
4. Az adatbázis kezelő MariaDB 2.7.2.
5. A program RUP fejlesztési módszer szerint készült, a fejlesztés iteratív és inkrementális.
6. Alapvető funkciók:
7. Könyvtártagok adatainak nyilvántartása (tagi azonosító, név, telefon, e-mail, könyvtartozás).
8. Új tag felvétele, illetve meglévő tag adatainak módosítása.
9. Könyvek adatainak nyilvántartása (könyv azonosító, szerző, cím, készlet).
10. Új könyv felvétele, illetve meglévő könyv adatainak módosítása.
11. Tranzakciók (kölcsönzés/visszavétel) rögzítése időbélyeggel, tag és könyv adatokkal.
12. A könyvtartozással rendelkező tagok külön megjelenítése.

# Stage RUP 1 ( Követelmények )

## Követelményfeltárás:

### Nem funkcionális követelmények:

A programnak Windows operációs rendszer alatt futnia kell.

Gépigény:

Windows 10

4 GB RAM

2 GHz processzor

Háttérprogramok:

Java futtatókörnyezet, MariaDB adatbáziskezelő.

A fejlesztés NetBeans IDE 12.2 használatával történik, Java 14.0.2 és MariaDB 2.7.2 verziószámú komponensekkel.

A program Java nyelven készült.

### Funkcionális követelmények:

A program használatához autentikáció szükséges.

Az adatokat SQL adatbázis tárolja.

Az adatokat grafikus felületen kell megjeleníteni.

Adatok nyilvántartása:

-könyvtártagok azonosítója

-könyvtártagok neve

-telefonszáma

-E-mail címe

-egyenleg (könyvtartozás)

-könyvek adatainak tárolása (kód, szerző, cím, készlet)

-tranzakciók nyilvántartása

új adatok felvétele

meglévő adatok módosítása

## Megszorítások:

Autentikáció → adatbázisból, jelszó nem olvasható formátumban tárolva

A fő ablak vezérlő gombjai inaktívak míg az autentikáció nem sikeres.

Felvett tagot és könyvet nem törlünk az adatbázisból.

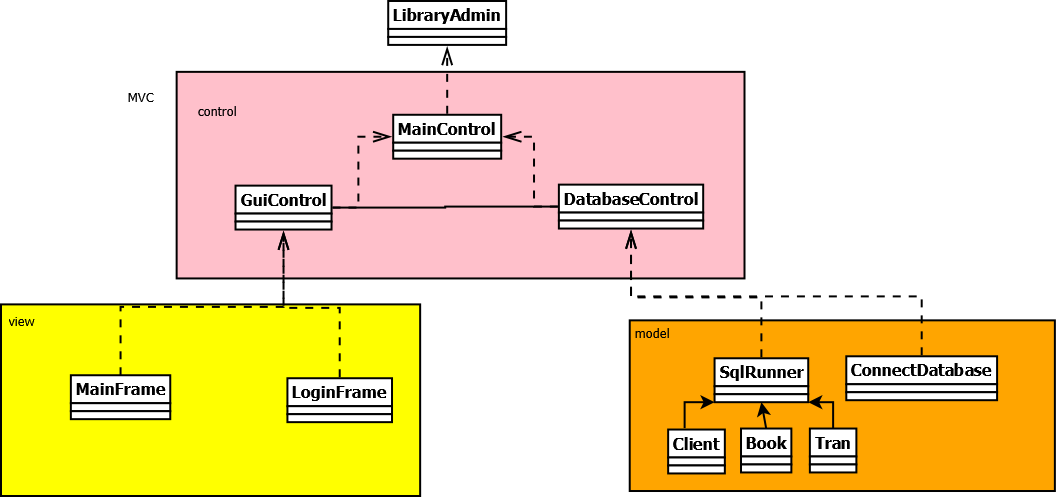
Adatbázis neve: library.

Autentikációs jelszót titkosítva kell tárolni.

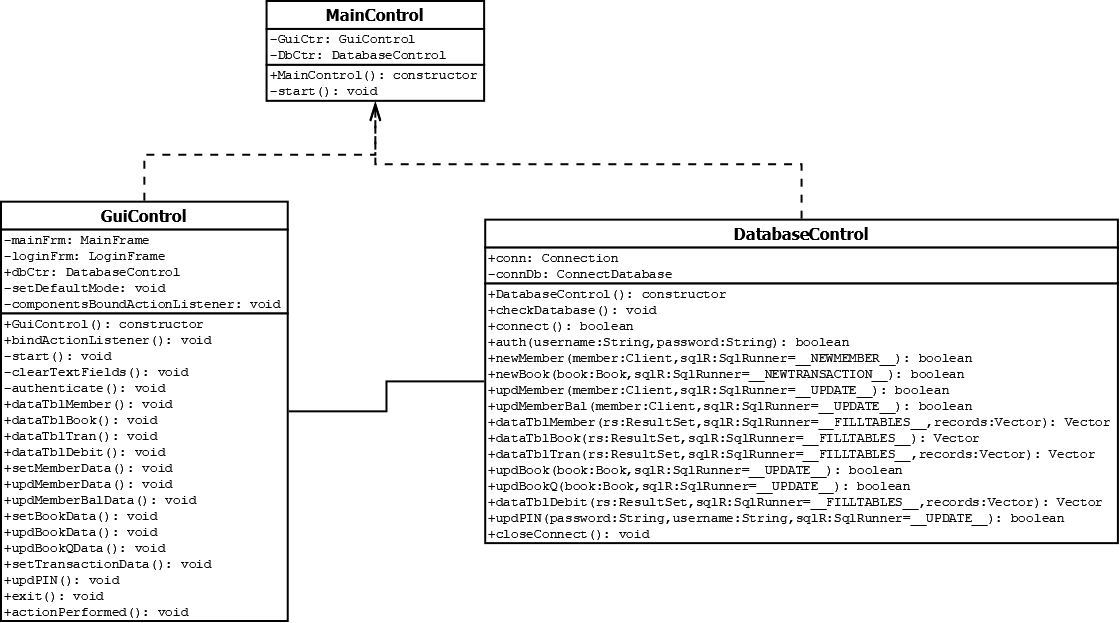
# stage RUP 2 ( Tervezés )

## UML tervek:

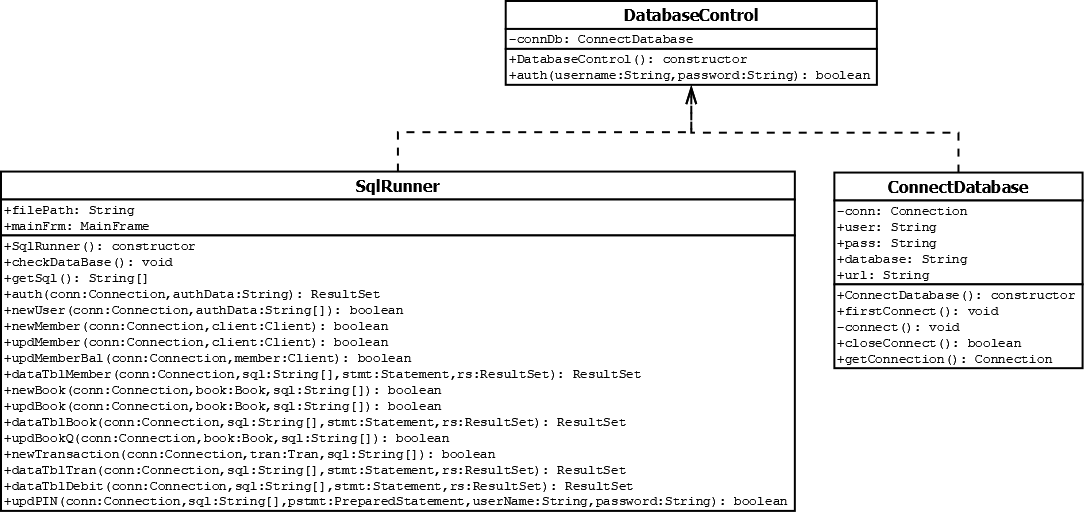
### MVC model:



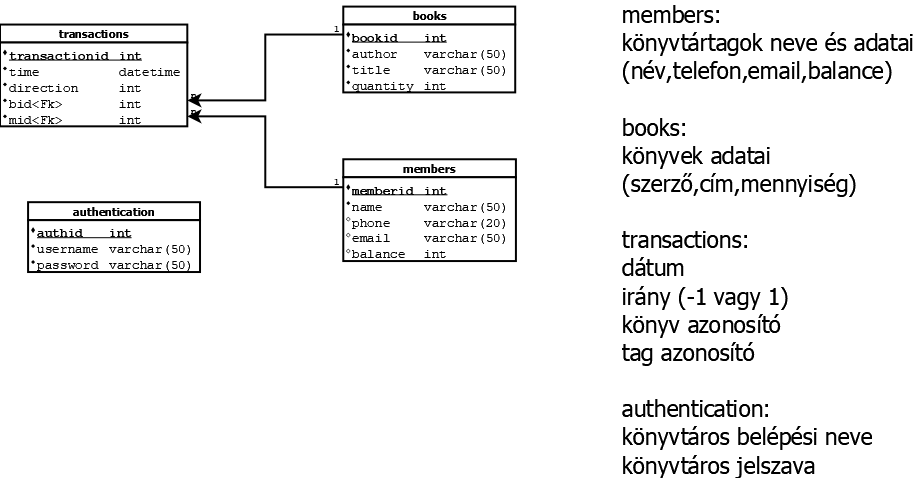
### Control csomag:



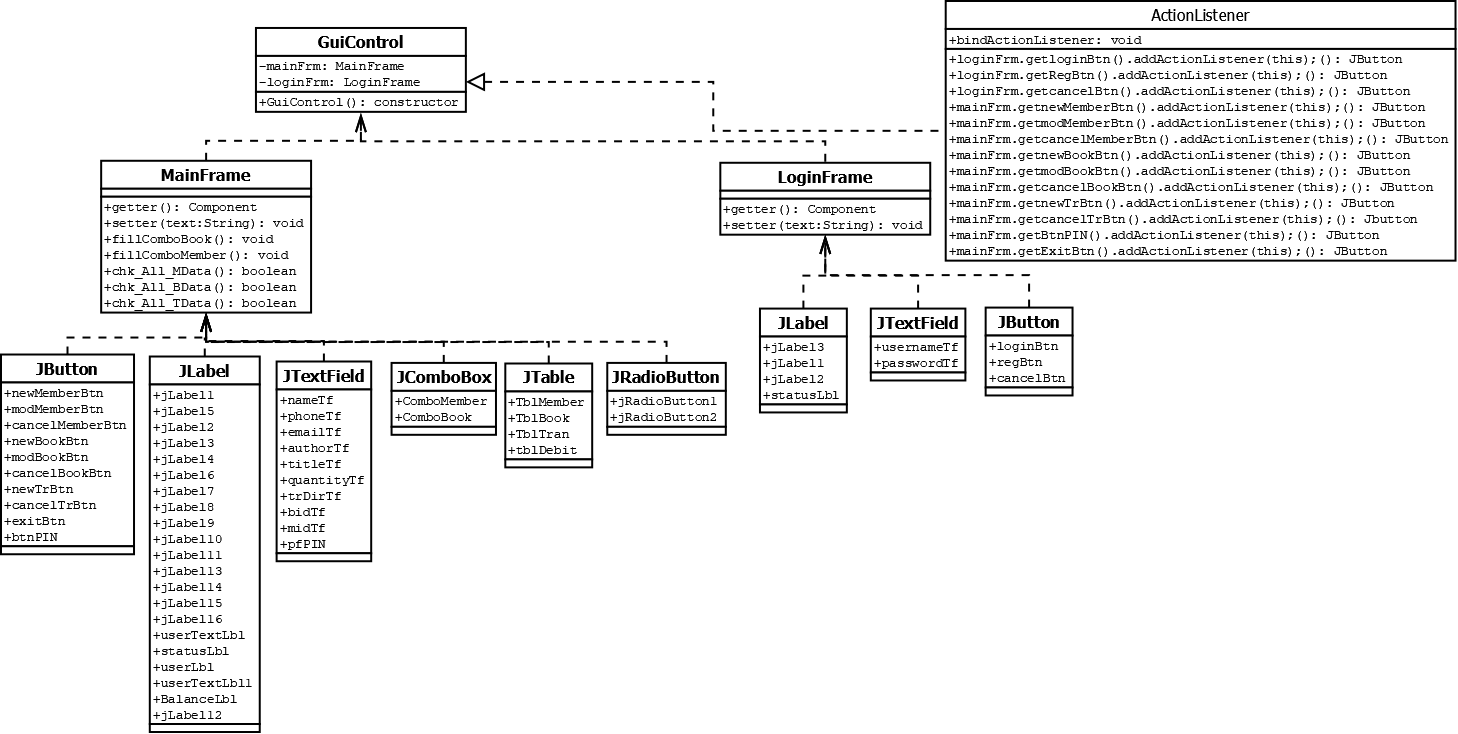
### Model csomag:



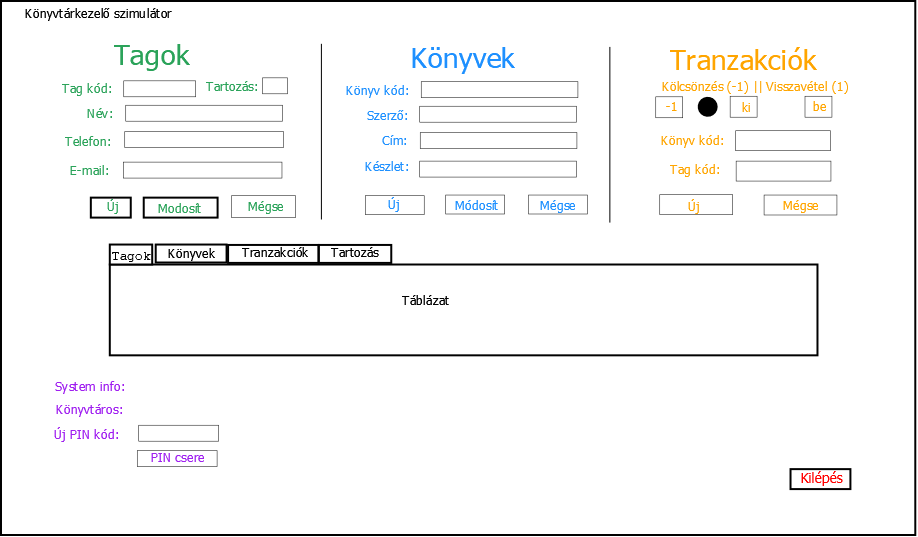
### Adatbázis:



### View csomag:



## GUI terv:



# stage RUP 3 ( Implementálás )

## ConnectDatabase osztály

Importált csomagok:

java.sql.Connection, java.sql.DriverManager, java.sql.SQLException

Osztályváltozók:

private Connection conn ( Adatbázis kapcsolat )

private String user ( Felhasználónév az adatbázis kapcsolathoz )

private String pass ( Jelszó az adatbázis kapcsolathoz )

private String database ( Adatbázis neve a kapcsolódáshoz )

private String url ( Adatbázis kapcsolat url -je )

Metódusok:

ConnectDatabase() (konstruktor)

Bejövő paraméter: nincs

Beállítja az osztályváltozók alapértelmezett értékét.

firstConnect()

Attributum: public

Bejövő paraméter: nincs

Felépít egy általános kapcsolatot az adatbázis szerverrel az url, az user és a pass változók értéke szerint. Kivételkezelés: SQLException, catch ágon java hibaüzenet kiíratása debug módban.

Visszatérési érték: nincs

connect()

Attributum: public

Bejövő paraméter: nincs

Felépít egy adatbázishoz kötődő kapcsolatot az url, a database, az user, és a pass változók értéke szerint.

A kapcsolatot a conn változóban tárolja. Kivételkezelés: SQLException, catch ágon java hibaüzenet kiíratása debug módban.

Visszatérési érték: nincs

closeConnect()

Attributum: public

Bejövő paraméter: nincs

Lezárja az aktuálisan felépített és tárolt adatbázis kapcsolatot. Kivételkezelés: SQLException, catch ágon java hibaüzenet kiíratása debug módban.

Visszatérési érték: boolean

getConnection()

Attributum: public

Bejövő paraméter: nincs

Visszaadja a conn változó értékét.

Visszatérési érték: Connection

## LoginFrame osztály (extends javax.swing.JFrame)

Importált csomagok:

import javax.swing.JButton;

import javax.swing.JLabel;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

Konstruktor: initComponents()

A bejelentkezési ablakot/űrlapot jeleníti meg.

Metódusok:

Az űrlap elemeinek getterei-setterei

## MainFrame osztály (extends javax.swing.JFrame)

Importált csomagok:

import javax.swing.JButton;

import javax.swing.JLabel;

import javax.swing.JTextField;

import java.sql.ResultSet;

import java.sql.PreparedStatement;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import javax.swing.JPasswordField;

import javax.swing.JTable;

Konstrunktor:

initComponents();

fillComboBook();

fillComboMember();

A program fő ablaka/űrlapja.

Metódusok:

Az űrlap elemeinek getterei-setterei

fillComboMember()

Attribútum: public void

Bemenő paraméterek: connection, sql

Visszatérési érték: combobox feltöltése memberid-kkal

SQLException, catch ágon java hibaüzenet kiíratása debug módban

fillComboBook()

Attribútum: public void

Bemenő paraméterek: connection, sql

Visszatérési érték: combobox feltöltése bookid-kkal

SQLException, catch ágon java hibaüzenet kiíratása debug módban

## SqlRunner osztály

Importált csomagok:

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.sql.ResultSet;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.sql.Connection;

import java.sql.Statement;

import java.sql.DatabaseMetaData;

Osztályváltozók:

private String filePath; (futtatandó SQL file-ok elérési útvonala)

Konstruktor: this.filePath = "sql/" + filename + ".sql";

Bejövő paraméter: filename (az SQL file neve)

Visszatérési érték: az SQL file az elérési útvonalával együtt

Metódusok:

checkDataBase(Connection conn)

Bejövő paraméter: Connection, a ConnectDatabase osztály getConnection() metódusából

Ellenőrzi az adatbázis létezését a metaadatokból, hogy létezik-e a books tábla, ha nem létezik, akkor létrehozza az egész adatbázist.

Kivételkezelés: SQLException, catch ágon java hibaüzenet kiíratása debug módban.

Visszatérési érték: nincs

getSql()

Attributum: public

Bejövő paraméter: nincs

SQL file-t ;-kel tagolva feldarabolja

Visszatérési érték: a darabolás után az i-dik indexen lévő SQL utasítás

IOException, catch ágon java hibaüzenet kiíratása debug módban, és hibaüzenet a felhasználónak „Hiba a beolvasás során.”

auth(Connection conn, String[] authData)

Attributum: public ResultSet

Bejövő paraméter: adatbázis kapcsolat, azonosítási adatok

Azonosítási adatokat kéri le (felhasználónév, jelszó)

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Visszatérési érték: ResultSet (felhasználónév, jelszó)

newUser(Connection conn, String[] authData)

Attributum: public boolean

Bejövő paraméter: adatbázis kapcsolat, azonosítási adatok

Az új felhasználó bejelentkezési adatait kéri le (felhasználónév, jelszó)

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Visszatérési érték: boolean

newMember(Connection conn, Client client)

Attributum: public boolean

Bejövő paraméter: adatbáziskapcsolat, Client típusú objektum

Új könyvtártag adait szúrja be a members táblába

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Attributum: public boolean

updMember(Connection conn, Client client)

Attributum: public boolean

Bejövő paraméter: adatbáziskapcsolat, Client típusú objektum

Már létező könyvtártag adait módosítja a members táblában

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Attributum: public boolean

updMemberBal(Connection conn, Client member)

Attributum: public boolean

Bejövő paraméter: adatbáziskapcsolat, Client típusú objektum

Már létező könyvtártag tartozását módosítja a rögzített tranzakció alapján a members táblában

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Attributum: public boolean

dataTblMember (Connection conn)

Attributum: public ResultSet

Bejövő paraméter: adatbázis kapcsolat

A könyvtártag adatbázis táblát jeleníti meg az űrlapon

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Visszatérési érték: ResultSet (members tábla adatai)

newBook(Connection conn, Book book)

Attributum: public boolean

Bejövő paraméter: adatbáziskapcsolat, Book típusú objektum

Új könyv adait szúrja be a books táblába

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Attributum: public boolean

updBook(Connection conn, Book book)

Attributum: public boolean

Bejövő paraméter: adatbáziskapcsolat, Book típusú objektum

Már létező könyv adatait módosítja a books táblában

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Attributum: public boolean

dataTblBook (Connection conn)

Attributum: public ResultSet

Bejövő paraméter: adatbázis kapcsolat

A books adatbázis táblát jeleníti meg az űrlapon

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Visszatérési érték: ResultSet (books tábla adatai)

updBookQ(Connection conn, Book book)

Attributum: public boolean

Bejövő paraméter: adatbáziskapcsolat, Book típusú objektum

Már létező könyv mennyiségi adatát módosítja a books táblában

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Attributum: public boolean

newTransaction(Connection conn, Tran tran)

Attributum: public boolean

Bejövő paraméter: adatbáziskapcsolat, Tran típusú objektum

Egy könyv kiadását illetve visszvételét rögzíti mint tranzakciót

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Attributum: public boolean

dataTblTran (Connection conn)

Attributum: public ResultSet

Bejövő paraméter: adatbázis kapcsolat

A tranzakciók táblát jeleníti meg az űrlapon

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Visszatérési érték: ResultSet (transactions tábla adatai)

dataTblDebit (Connection conn)

Attributum: public ResultSet

Bejövő paraméter: adatbáziskapcsolat

A könyvtártagok tartozásait jeleníti meg táblázatban az űrlapon

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Visszatérési érték: ResultSet (tartozások adatai)

updPIN(Connection conn, String[] authData)

Attributum: public boolean

Bejövő paraméter: adatbáziskapcsolat, azonosítási adatok

A bejelentkezett könyvtáros jelszavát változtatja meg

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Attributum: public boolean

## Client osztály

Osztályváltozók:

private Integer memberid;

private String name;

private String phone;

private String email;

private Integer balance;

Metódusok:

Az osztályváltozók getterei és setterei

Attributum: public

## Book osztály

Osztályváltozók:

private Integer bookid;

private String author;

private String title;

private int quantity;

Metódusok:

Az osztályváltozók getterei és setterei

Attributum: public

## Tran osztály

Osztályváltozók:

private int direction;

private int bid;

private int mid;

Metódusok:

Az osztályváltozók getterei és setterei

Attributum: public

## DatabaseControl osztály

Importált csomagok:

import java.sql.Connection;

import model.ConnectDatabase;

import java.sql.ResultSet;

import model.SqlRunner;

import java.sql.SQLException;

import java.util.Vector;

import model.Book;

import model.Client;

import model.Tran;

osztályváltozók:

private Connection conn;

private ConnectDatabase connDb;

konstruktor: checkDatabase()

metódusok:

checkDatabase()

Attributum: public, void

Bejövő paraméter: nincs

Meghívja az SqlRunner osztály checkDataBase() metódusát

Visszatérési érték: nincs

connect()

Attributum: protected boolean

Bejövő paraméter: nincs

Meghívja a ConnectDatabase osztály getConnection() metódusát

Visszatérési érték: boolean

auth(String[] authData)

Attributum: protected int

Bejövő paraméter: felhasználónév, jelszó

Meghívja az SqlRunner osztály auth() metódusát

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Visszatérési érték: egész szám

newUser(String[] authData)

Attributum: protected boolean

Meghívja az SqlRunner osztály newUser() metódusát

Bejövő paraméter: felhasználónév, jelszó

Visszatérési érték: boolean

newMember(Client member)

Attributum: protected boolean

Meghívja az SqlRunner osztály newMember() metódusát

Bejövő paraméter: Client típusú objektum

Visszatérési érték: boolean

newBook(Book book)

Attributum: protected boolean

Meghívja az SqlRunner osztály newBook() metódusát

Bejövő paraméter: Book típusú objektum

Visszatérési érték: boolean

newTransaction(Tran tran)

Attributum: protected boolean

Meghívja az SqlRunner osztály newTransaction() metódusát

Bejövő paraméter: Tran típusú objektum

Visszatérési érték: boolean

updMember(Client member)

Attributum: protected boolean

Meghívja az SqlRunner osztály updMember() metódusát

Bejövő paraméter: Client típusú objektum

Visszatérési érték: boolean

updMemberBal(Client member)

Attributum: protected boolean

Meghívja az SqlRunner osztály updMemberBal() metódusát

Bejövő paraméter: Client típusú objektum

Visszatérési érték: boolean

updBook(Book book)

Attributum: protected boolean

Meghívja az SqlRunner osztály updBook() metódusát

Bejövő paraméter: Client típusú objektum

Visszatérési érték: boolean

updBookQ(Book book)

Attributum: protected boolean

Meghívja az SqlRunner osztály updBookQ() metódusát

Bejövő paraméter: Client típusú objektum

Visszatérési érték: boolean

dataTblMember()

Attributum: protected Vector<Vector<String>>

Meghívja az SqlRunner dataTblMember() metódusát

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Visszatérési érték: Vector<Vector<String>>

dataTblBook()

Attributum: protected Vector<Vector<String>>

Meghívja az SqlRunner dataTblBook() metódusát

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Visszatérési érték: Vector<Vector<String>>

dataTblTran()

Attributum: protected Vector<Vector<String>>

Meghívja az SqlRunner dataTblDebit() metódusát

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Visszatérési érték: Vector<Vector<String>>

dataTblDebit()

Attributum: protected Vector<Vector<String>>

Meghívja az SqlRunner dataTblTran() metódusát

SQLException, catch ágon java hibaüzenet kiíratása debug módban

Visszatérési érték: Vector<Vector<String>>

updPIN(String[] authData)

Attributum: protected boolean

Meghívja az SqlRunner osztály updPIN() metódusát

Bejövő paraméter: Client típusú objektum

Visszatérési érték: boolean

closeConnect()

Attributum: protected void

Bejövő paraméter: nincs

Visszatérési érték: nincs

Bazárja az élő adatbázis kapcsolatot

## GuiControl

Implementálja az ActionListener interfészt, az eseménykezelés helyben történik.

Mezők:

- private mainFrm ( Példányosított MainFrame objektum )

- private loginFrm ( Példányosított LoginFrame objektum )

- private dbCtr ( Példányosított DatabaseControl objektum )

Változók:

MainFrame mainFrm – a fő panel/ablak osztályának példányosítása

LoginFrame loginFrm – a bejelentkező ablak osztályának példányosítása

Konstruktor:

Bejövő paraméter: DatabaseControl dbCtr tárolva a dbCtr mezőben.

ComponentsBoundActionListener() metódus hívása.

ComponentsBoundActionListener:

Nincs bejövő paraméter.

A formokon elhelyezett komponensek kötése az ActionListener interfészhez.

A komponensekhez getter és setter metódusokkal lehet hozzáférni.

SetDefaultMode:

Nincs bejövő paraméter.

Példányosítja a MainFrame osztályt és láthatóvá teszi. Példányosítja a LoginFrame osztályt és

láthatóvá teszi a MainFrame felett.

Elindítja az adatbázis kapcsolat felépítését a DatabaseControl connect() metódusának meghívásával.

A connect() metódus visszatér egy boolean típussal melyet kiértékel és az eredmény függvényében

a MainForm felületén a statusLbl komponensen megjeleníti a kapcsolódás eredményét.

A newBtn, modifyBtn, cancelBtn, pinBtn gombok inaktívvá tétele.

## MainControl oszály

Osztályváltozók:

private DatabaseControl dbCtr;

private GuiControl guiCtr;

konstruktor: start()

metódusok:

start()

Attributum: private void

Nincs bejövő paraméter.

Példányosítja a DatabaseControl és a GuiControl osztályt.

# Stage RUP4

Tesztelés

# Stage RUP5

Átadás

# Stage RUP6

Karbantartás