

Gymnázium, Praha 6, Arabská 14

předmět Programování, vyučující Mgr. Jan Lána



**Gymnázium
Arabská**

Ročníková práce

Známkování

Krystof Maxa 2.E

Duben 2023

Čestné prohlášení

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne

Anotace

Cílem programu je nahrazení nefunkčních aplikací, které slouží k zapisování známek do školního systému. Program se skládá z přehledného a jednoduchého prostředí, které je funkční a plní to, co plnit má tzn. veškeré funkce, které učitel potřebuje k zapsání známek a to je přidání známek, upravení známek nebo jejich odebrání. Program disponuje přihlašovacím systémem nebo také možností změnění uživatelského hesla.

Abstract

The goal of the program is to replace non-functional applications that are used for recording grades into the school system. The program consists of a clear and simple interface that is functional and fulfills what it is supposed to do, i.e. all the functions that a teacher needs to record grades, such as adding grades, modifying grades, or deleting them. The program has a login system or the possibility of changing the user's password.

Zadání

Jednodušší bakaláři. Učitel bude moci přidávat k různým studentům známky/body. Bude tam mít i kolonku na průměrnou známku, která jim z daného předmětu vychází. Učitel si u dané třídy vytvoří předměty, které je učí. Učitel má možnost mít více tříd.

Obsah

1	Úvod	6
2	SQL-databáze	7
2.1	Struktura databáze	7
2.1.1	Tabulka tříd	7
2.1.2	Tabulka předmětů	7
2.1.3	Tabulka učitelů	7
2.1.4	Tabulka studentů	7
2.1.5	Tabulka trida-predmet-ucitel	7
2.1.6	Tabulka zaznamyznamek	8
3	Přihlášení	10
3.1	Použití	10
3.2	Funkce přihlášení	10
3.3	Funkce změnit heslo	12
3.3.1	Změna hesla	13
4	Známkování - ZnamkovaniController	15
4.1	Použití	15
4.2	Vybrání třídy a předmětu	15
4.3	Načtení studentů	16
4.3.1	Tabulka pro záznamy známek	16
4.3.2	Tabulka pro průměrnou známku	18
5	Přidání známky	21
5.1	Použití	21
5.2	PridatZnamkuController	22
6	Změna známky	24
6.1	Použití	24
6.2	ZmenitZnamkuController	25
7	Odebrání známky	26
7.1	Použití	26
7.2	OdebratZnamkuController	26
8	Závěr	27
9	Citace	28

1 Úvod

Tento program usnadní práci učitelům při známkování svých studentů. Při spuštění programu si učitel vybere své jméno. Poté se heslem může přihlásit do samotného systému známkování, nebo si heslo změnit. Po přihlášení na učitele vyskočí volba "jeho" tříd a k nim předměty, které danou třídu učí. Poté klikne na tlačítko Aktualizovat a program učiteli zobrazí tabulku známek a druhou tabulku, kde uvidí průměrnou známku u každého studenta. Učiteli je umožněno vykonat hlavní tři funkce a to je přidání známky, k tomu slouží tlačítko Přidat, úprava známky, k tomu slouží tlačítko Upravit, odebrání známky a k tomu má učitel k dispozici tlačítko Odebrat. Díky těmto funkcím má učitel možnost jakkoliv bez problému pracovat se známky.

2 SQL-databáze

Pro práci a ukládání dat, jakožto záznamy známek, jména a hesla učitelů, jména žáků, tříd, předmětů a dalších dat, jsem používal SQL databázi. K práci s SQL databází jsem používal **MySQL**¹. S prostředím MySQL jsem se naučil pracovat díky tomuto videu² a díky mému tátovi. Obrázek schéma databáze jsem vytvořil na webové stránce DB designer³. Jak propojit MySQL databázi s vývojovým prostředím IntelliJ IDEA jsem se naučil díky tomuto videu⁴

2.1 Struktura databáze

2.1.1 Tabulka tříd

Tabulka tříd slouží k uchování jmen jednotlivých tříd (1.A, 2.A atd.).

2.1.2 Tabulka předmětů

Tabulka předmětů slouží k uchování jmen jednotlivých předmětů (Matematika, Biologie atd.).

2.1.3 Tabulka učitelů

Tabulka učitelů slouží k uchování dat jednotlivých učitelů. Každý učitel má křestní jméno, příjmení, heslo ("password") a své unikátní uživatelské jméno ("username").

2.1.4 Tabulka studentů

Tabulka studentů slouží k uchování dat jednotlivých studentů, konkrétně do jaké třídy chodí("trida_id"), křestní jméno a příjmení.

2.1.5 Tabulka trida-predmet-ucitel

Tabulka trida-predmet-ucitel slouží k tomu, aby se ke konkrétnímu učiteli přiřadila třída a jaký předmět třídu vyučuje.

¹<https://www.mysql.com/>

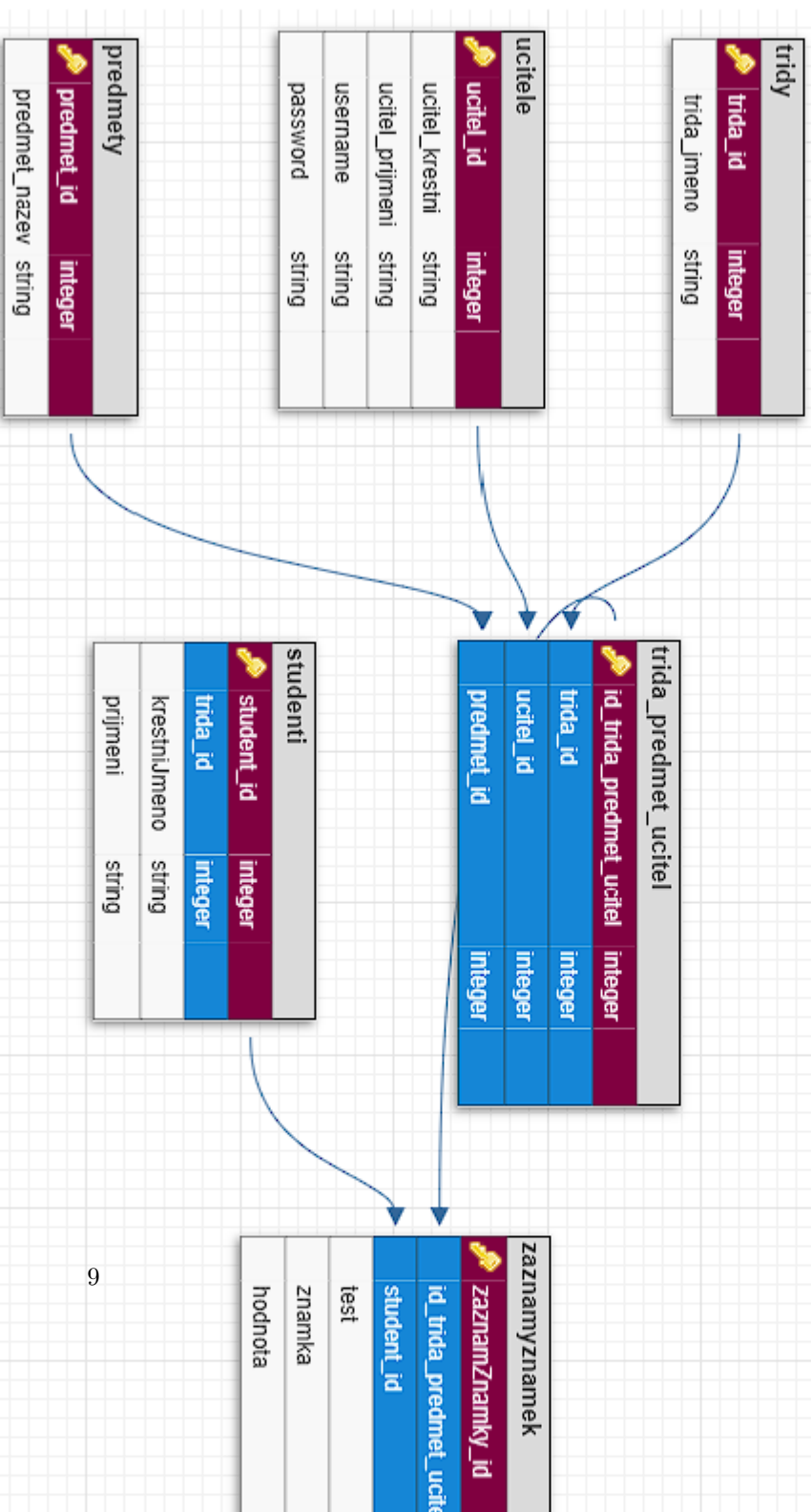
²https://www.youtube.com/watch?v=7S_tz1z5bAt = 9907s

³<https://www.dbdesigner.net/>

⁴<https://www.youtube.com/watch?v=8WJ5p3T9Iss>

2.1.6 Tabulka zaznamynamek

Tabulka zaznamynamek slouží ke shromažďování záznamů známek. Každý záznam známky má id-trida-predmet-ucitel, které slouží k přiřazení známky ke konkrétnímu učiteli, třídě a předmětu. Dále má student-id, které záznam známky přiřadí ke konkrétnímu studentovi. Kolonka test slouží k pojmenování daného testu (záznamu známky), kolonka znamka k přiřazení známky a kolonka hodnota k určení hodnoty (váhy) známky.



3 Přihlášení

3.1 Použití

Scéna přihlášení slouží k tomu, aby se učitel přihlásil, nebo si změnil heslo. V prostředí scény přihlášení je combobox, ve kterém si učitel vybere, pod jakým učitelem se chce přihlásit tzn. zvolí své jméno. Učitel zvolí své jméno a napíše do textového pole své heslo. Poté klikne na tlačítko "Přihlásit se". Program ověří zda je heslo správné a učitele přesune do scény, kde může pracovat se známkami. Dále je zde tlačítko "Změnit heslo", které přesune učitele do scény, kde si heslo může změnit. Logo jsem vytvořil přes webovou stránku Free Logo Design⁵

3.2 Funkce přihlášení

Pro přihlášení program používá třídu PrihlaseniController. S pomocí sql databáze program načte všechny existující učitele z databáze do ArrayListu objektů typu třídy Ucitel (*uciteleArr*), které mají parametry: username, password, krestni, prijmeni. Po vybrání svého jména z ComboBoxu *cbox-ucitele*, se do proměnné *ucitelUsernamePrihlaseni* nastaví username vybraného učitele.

```
try {
    ConnectionClass connectionClass = new ConnectionClass();
    Connection connection = connectionClass.getConnection();
    Statement statement = connection.createStatement();
    ResultSet resultSet = statement.executeQuery("SELECT
ucitel_krestni , ucitel_prijmeni , username , password FROM ucitele ");

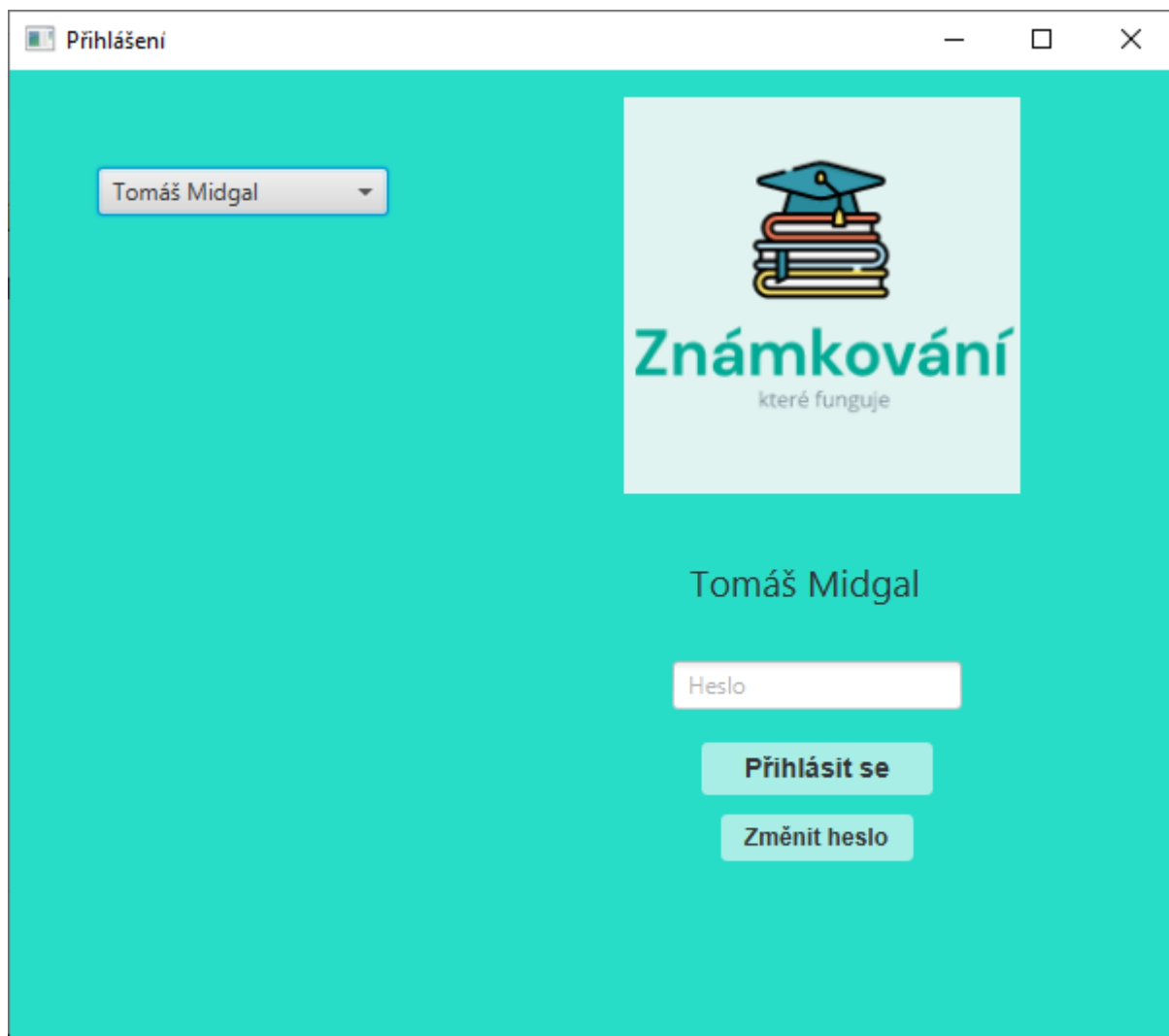
    while (resultSet.next()) {
        String krestni = resultSet.getString(" ucitel_krestni");
        String prijmeni = resultSet.getString(" ucitel_prijmeni");
        String username = resultSet.getString(" username");
        String password = resultSet.getString(" password");

        //ObservableList který slouží jen pro vložení ucitelu do ComboBoxu cbox_ucitele
        ucitele.add(new Ucitel(krestni, prijmeni, username, password));
        //ArrayListu ucitelu, podle kterého overovat přihlasovací údaje
        uciteleArr.add(new Ucitel(krestni, prijmeni, username, password));
    }
} catch (SQLException e) {
    e.printStackTrace();
}
```

⁵<https://www.freelogodesign.org/>


Poté učitel napíše své heslo do `TextFieldu` *tf – password* a klikne na tlačítko Přihlásit se. Po stisknutí se spustí ověření hesla s příslušným uživatelským jménem učitele. Program projede celý `ArrayList` učitelů a hledá takového učitele, kterému se shoduje celé jméno (křestní + " " + příjmení) s jménem, vybraným z *cbox – ucitele*. Poté porovná *password* nalezeného učitele s textem, který učitel napísal do *tf – password*. Jestliže se texty shodují, program zavolá metodu `changeScene` třídy `DBUtils`, která učitele přesune do jiné scény, konkrétně do scény, kde učitel známkuje. Jestliže se texty neshodují, zobrazí se text: Chybné heslo

```
for (int i = 0; i < uciteleArr.size(); i++) {
    if (ucitel.equals(uciteleArr.get(i).krestni + " " + uciteleArr.get(i).prijmeni)) {
        if (uciteleArr.get(i).password.equals(tf_password.getText())) {
            ucitelUsernamePrihlaseni = String.valueOf(uciteleArr.get(i).username);
            DBUtils.changeScene(event, "znamkovani.fxml", "Znamkovani");
        } else {
            label_spatneHeslo.setText("Chybné heslo");
        }
    }
}
```



Přihlášení

Tomáš Midgal ▼


Známkování
které funguje

Tomáš Midgal

Heslo

Přihlásit se

Změnit heslo

3.3 Funkce změnit heslo

Pro možnost změny hesla je zde tlačítko *Změnit heslo*. Prvně by měl učitel zvolit své jméno. Když učitel klikne na tlačítko *Změnit heslo* bez zvolení svého jména, tak ho program přesune do scény, kde si heslo může změnit, ale rovnou bude informován, že se má vrátit a zvolit své jméno. Bez zvolení jména program nemůže poznat, o jakého učitele se jedná. Prvně musí učitel napsat své nynější

heslo. Poté do `TextFieldu tf – noveHeslo` zadá své nové heslo a do `TextFieldu tf – noveHesloOvereni` ho zadá znovu. Po stisknutí tlačítka *Změnit heslo* program ověří nynější heslo a poté zda se nová hesla shodují. Když je nynější heslo správné a zároveň se shodují se sebou hesla nová, tak program změní staré heslo na nové.

3.3.1 Změna hesla

Pro změnu hesla program používá třídu `ZmenaHeslaController`. Program jako první načte do proměnné `username` username zvoleného učitele ze třídy `PrihlaseniController`.

```
username = PrihlaseniController.ucitelUsernamePrihlaseni;
```

Poté zkontroluje zda `username` není null. Pokud `username` null není, načte k němu z databáze jeho password.

```
try{
    ConnectionClass connectionClass = new ConnectionClass();
    Connection connection = connectionClass.getConnection();
    Statement statement = connection.createStatement();
    ResultSet resultSet = statement.executeQuery("SELECT password
FROM ucitele\n" + "WHERE username = \""+username+"\"");

    while(resultSet.next()) {
        passwordAktualni = resultSet.getString("password");
    }
}catch (Exception e){
    e.printStackTrace();
}
```

Následně proběhne ověření nynějšího hesla, když je správné, tak proběhne ověření, zda jsou nová hesla stejná, jinak program zobrazí nápis: Nesprávné heslo. Jestliže se shodují i nová hesla, program aktualizuje v databázi heslo pro daného učitele na heslo nové a zobrazí se text: Heslo bylo úspěšně změněno. Při neshodě nových hesel se zobrazí text: Hesla se neshodují.

```
String aktualniHeslo = String.valueOf(tf_stareHeslo.getText());
passwordNove = String.valueOf(tf_noveHeslo.getText());
passwordNoveOvereni = String.valueOf(tf_noveHesloOvereni.getText());
if (aktualniHeslo.equals(passwordAktualni)) {
    if (passwordNove.equals(passwordNoveOvereni)) {
        try {
            ConnectionClass connectionClass = new ConnectionClass();
            Connection connection = connectionClass.getConnection();
            Statement statement = connection.createStatement();
            statement.execute("UPDATE ucitele SET password = \"" +
passwordNove + "\"\n" + "WHERE username = \"" + username + "\"");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

    }
    label_chybaNoveHeslo.setText(" Heslo bylo      spn      z m n n o ");
} else {
    label_chybaStareHeslo.setText("");
    label_chybaNoveHeslo.setText(" Hesla se neshoduj ");
}
} else {
    label_chybaNoveHeslo.setText("");
    label_chybaStareHeslo.setText(" Nespr v n heslo ");
}
}

```

Po změně hesla stiskne učitel tlačítko Přihlásit se, které ho přesune do scény pro přihlášení. A nastaví username zpátky na null.

The screenshot shows a Java Swing window titled "Změna hesla-tomasmidgal". The window has a standard title bar with minimize, maximize, and close buttons. The main content area has a light gray background. In the center, the name "Tomáš Midgal" is displayed in a bold, black, sans-serif font. Below the name, there are four input fields arranged vertically. The first field is empty and has a blue border. The second and third fields contain the placeholder text "Nové heslo" in a light gray font. The fourth field is empty. At the bottom of the window, there are two buttons: "Změnit heslo" and "Přihlásit se", both in a light gray font and with a slight 3D effect.

4 Známkování - ZnamkovaniController

4.1 Použití

Pro práci se známky program používá hlavní třídu ZnamkovaniController a k ní třídy PridatZnamkuController, ZmenitZnamkuController, OdebratZnamkuController. Ve scéně známkování si učitel z ComboBoxu `cmbox_trida` vybere třídu, do které chce známky zapsat. Na možnost se mu zobrazí jen ty třídy, které učitel vyučuje. Poté si z ComboBoxu `cmbox_predmet` vybere k jakému předmětu chce známky připsat, opět se mu jako možnost zobrazí jen ty předměty, které třídu vyučuje. Poté klikne na tlačítko Aktualizovat, díky kterému se do tabulky pro záznamy známek načtou příslušní studenti s jejich známkami a do tabulky pro průměrnou známku se načtou studenti s jejich průměrnou známkou. Dále má učitel k dispozici tlačítka Přidat, Upravit, Odebrat. Tlačítkem Přidat učitel k danému studentovi známku přidá, tlačítkem Upravit mu konkrétní známku upraví a tlačítkem Odebrat konkrétní známku odebere. Jak přesunout data mezi jednotlivými třídami jsem se naučil díky tomuto videu⁶. Jak pracovat s TableView jsem se naučil díky sérii těchto tří videí^{7 8 9} a ORACLE dokumentace¹⁰

4.2 Vybrání třídy a předmětu

Podobně jako ve třídě ZmenaHeslaController program načte do proměnné `usernameUcitel` username zvoleného učitele ze třídy PrihlaseniController.

```
//prepise usernameUcitel
usernameUcitel = PrihlaseniController.ucitelUsernamePrihlaseni;
```

Poté program načte z databáze všechny třídy, které tento učitel učí, a nahraje je do ArrayListu `tridy`. Následně vloží ArrayList do ComboBoxu `cmbox_trida`.

```
//nacte existujici tridy pro tento username a nacte je do arraylistu tridy
try {
```

⁶<https://www.youtube.com/watch?v=dFLTha7jH7o>

⁷1. <https://www.youtube.com/watch?v=vego72w5kPUlist=PL3bGLnkkGnuUQemxQ3ojPcuMt5d8B5pyindex> =

21 <https://www.youtube.com/watch?v=Mkp20qrtLwlist=PL3bGLnkkGnuUQemxQ3ojPcuMt5d8B5pyindex> =

22 <https://www.youtube.com/watch?v=b-0QMxCB2zIlist=PL3bGLnkkGnuUQemxQ3ojPcuMt5d8B5pyindex> =

23 ¹⁰<https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/table-view.htmCJAGAAEE>

```

ConnectionClass connectionClass = new ConnectionClass();
Connection connection = connectionClass.getConnection();
Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery("select trida_jmeno
from trida_predmet_ucitel tpu " +
"JOIN tridy t ON (tpu.trida_id=t.trida_id)" +
"JOIN ucitele u ON(tpu.ucitel_id=u.ucitel_id) " +
"WHERE username = \" + usernameUcitel + "\"");

while(resultSet.next()) {
    String tridaJmeno = resultSet.getString("trida_jmeno");
    tridy.add(tridaJmeno);
}
}catch (Exception e){
    e.printStackTrace();
}

//array tridy nacte do comboboxu tridy, ze ktereho se vybere-
//ke ktere tride to chceme nacist
cmbobox_trida.getItems().addAll(tridy);

```

Dále program úplně stejným principem načte z databáze všechny předměty, které v této třídě učitel vyučuje. Toto načtení se spustí poté, co učitel vybere z `cmbobox_trida` třídu. Předměty nahraje do `ArrayListu` `predmety`. Následně vloží `ArrayList` do `ComboBoxu` `cmbobox_predmet` a do proměnné `tridaComboBox` vloží vybranou třídu.

```
tridaComboBox = cmbobox_trida.getValue();
```

Po vybrání předmětu z `cmbobox_predmet` se do proměnné `predmetComboBox` vloží vybraný předmět.

4.3 Načtení studentů

4.3.1 Tabulka pro záznamy známek

Tato funkce se spustí po stisknutí tlačítka Aktualizace. Funkce má za úkol nahrát všechny studenty z vybrané třídy, pro které existuje známka k vybranému předmětu. Prvně z tabulky vymaže všechny kolonky a data, jelikož program vždy načte celou tabulku znovu.

```
table_users.setColumns().clear();
table_users.getItems().clear();
```

Poté vytvoří kolonky: `tableColumnId`(id známky, tuto kolonku učitel neuvidí), `tableColumnPrijmeni`(příjmení studenta), `tableColumnKrestni`(křestní studenta), `tableColumnTest`(název testu), `tableColumnZnamka`(známka), `tableColumnHodnota`(hodnota známky).

Ukázka vytvoření kolonky `tableColumnId` a `tableColumnPrijmeni`:


```

TableColumn tableColumnId = new TableColumn("id");
tableColumnId.setCellValueFactory(new PropertyValueFactory<Student, Integer>("id"));
tableColumnId.setVisible(false);

```

```

TableColumn tableColumnPrijmeni = new TableColumn("Prijmeni");
tableColumnPrijmeni.setCellValueFactory(new
PropertyValueFactory<Student, String>("prijmeni"));

```

Poté si vytvoří `ArrayList studentiTabulka` do kterého bude ukládat objekty typu třídy `StudentZnamkaZaznam`, který má parametry: `int id`; `String krestni`; `String prijmeni`; `String test`; `int znamka`; `int hodnota`. Z databáze načte všechny záznamy známek, které existují k vybrané třídě a předmětu. Jako poslední pomocí `ObservableList students1` vloží záznamy do tabulky.

```

ArrayList<StudentZnamkaZaznam> studentiTabulka = new ArrayList<>();

```

```

try {
    ConnectionClass connectionClass = new ConnectionClass();
    Connection connection = connectionClass.getConnection();
    Statement statement = connection.createStatement();

    ResultSet resultSet = statement.executeQuery("
SELECT zaznamZnamky_id, krestniJmeno, prijmeni, test, znamka, hodnota
FROM zaznamyznamek zz\n" +
"JOIN studenti s ON(zz.student_id=s.student_id)\n" +
"JOIN trida_predmet_ucitel tpu ON
(zz.id_trida_predmet_ucitel=tpu.id_trida_predmet_ucitel)\n" +
"JOIN tridy t ON (tpu.trida_id = t.trida_id)\n" +
"JOIN predmety p ON (tpu.predmet_id = p.predmet_id)\n" +
"JOIN ucitele u ON (tpu.ucitel_id = u.ucitel_id)\n" +
"WHERE trida_jmeno = \""+ tridaComboBox +"\"
AND predmet_nazev = \""+ predmetComboBox +"\"
AND username = \""+usernameUcitel+"\"");

    while (resultSet.next()) {
        int id = resultSet.getInt("zaznamZnamky_id");
        String surName = resultSet.getString("prijmeni");
        String firstName = resultSet.getString("krestniJmeno");
        String test = resultSet.getString("test");
        int znamka = resultSet.getInt("znamka");
        int hodnota = resultSet.getInt("hodnota");

        studentiTabulka.add(new StudentZnamkaZaznam
(id,surName,firstName,test, znamka,hodnota));
    }
} catch (Exception e) {
    e.printStackTrace();
}

```

4.3.2 Tabulka pro průměrnou známku

Tato funkce se také spustí po stisknutí tlačítka Aktualizace. Proběhne po načtení záznamů do tabulky pro záznamy známek. Jejím úkolem je vypočítat ke každému studentovi jeho průměrnou známku a nahrát tyto informace do tabulky pro průměrnou známku. Prvně z tabulky vymaže všechny kolonky a data, jelikož program vždy načte celou tabulku znovu.

```
tablePrumernaZnamka.setColumns().clear();
tablePrumernaZnamka.getItems().clear();
```

Poté si vytvoří kolonky pro příjmení studenta, křestní jméno a jeho průměrnou známku. Kolonky přidá do tabulky. Poté si vytvoří ArrayList `studentiPrumer` do kterého bude ukládat objekty typu třídy `PrumerZnamka`. Objekt typu `PrumerZnamka` má parametry: `String prijmeni`; `String krestni`; `double znamkaPrumer`. Program v databázi najde a vypočte průměrnou známku každého studenta, ve zvolené třídě a k ní zvoleném předmětu, a následně data nahraje do `studentiPrumer`. Jako poslední pomocí `ObservableList students2` vloží načtená data do tabulky.

```
try {
    ConnectionClass connectionClass = new ConnectionClass();
    Connection connection = connectionClass.getConnection();
    Statement statement = connection.createStatement();
    SQL_statement = "SELECT s.prijmeni , s.krestniJmeno ,
    ROUND(sum(hodnota*znamka)/sum(hodnota),2) AS znamka " +
    "FROM zaznamyznamek zn " +
    "JOIN studenti s ON (s.student_id=zn.student_id)" +
    "JOIN trida_predmet_ucitel tpu ON
    (zn.id_trida_predmet_ucitel=tpu.id_trida_predmet_ucitel)" +
    "JOIN tridy t ON (tpu.trida_id = t.trida_id)" +
    "JOIN predmety p ON (tpu.predmet_id = p.predmet_id)" +
    "JOIN ucitele u ON (tpu.ucitel_id = u.ucitel_id)" +
    "WHERE t.trida_jmeno = '\"'+tridaComboBox+'\"
    AND p.predmet_nazev = '\"'+predmetComboBox+'\"
    AND u.username = '\"'+usernameUcitel+'\"" +
    " group by s.prijmeni , s.krestniJmeno " +
    "ORDER BY s.prijmeni , s.krestniJmeno";

    ResultSet resultSet = statement.executeQuery(SQL_statement);

    while (resultSet.next()) {
        String prijmeni = resultSet.getString("prijmeni");
        String krestni = resultSet.getString("krestniJmeno");
        double znamka = resultSet.getDouble("znamka");

        studentiPrumer.add(new PrumernaZnamka(prijmeni , krestni , znamka));
    }
} catch (Exception e) {
    e.printStackTrace();
}
```

}

Odhlásit se

Odebrat

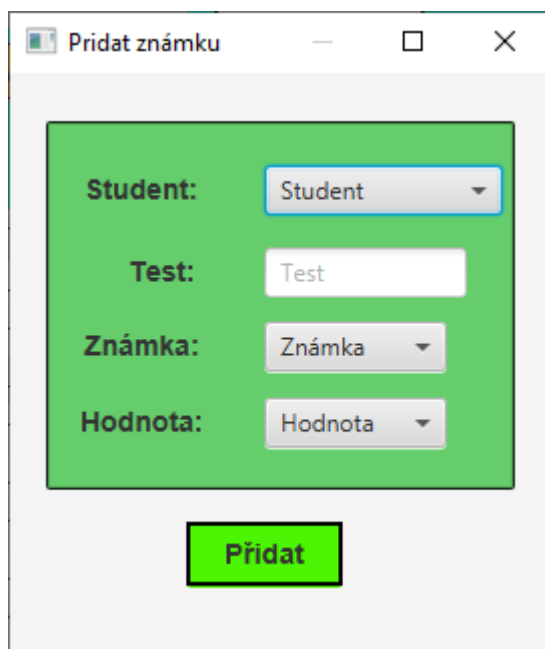
Průměr

[illegible]

5 Přidání známky

5.1 Použití

Pro přidání známky slouží třída `PridatZnamkuController`. Po stisknutí tlačítka Přidat známku se objeví okénko pro přidání známky. Prostředí se skládá z `ComboBoxu cmbox_student`, ve kterém si učitel vybere, ke kterému studentovi známku přiřadí. Poté je zde `TextField tf_test`, do kterého učitel napíše název testu. V `ComboBoxu cmbox_znamka` učitel vybere známku, kterou student dostane a v `ComboBoxu cmbox_hodnota` vybere váhu, kterou bude známka mít. A v poslední řadě je zde tlačítko `button_pridat`, které přidá vytvořený záznam známky do databáze.



The image shows a Java Swing window titled "Pridat známku". Inside the window, there is a green rectangular panel containing four labels and their corresponding input fields:

- Student:** A dropdown menu with "Student" selected.
- Test:** A text input field containing the text "Test".
- Známka:** A dropdown menu with "Známka" selected.
- Hodnota:** A dropdown menu with "Hodnota" selected.

Below the green panel, centered, is a red button with the text "Přidat".

5.2 PridatZnamkuController

Z databáze načte studenty z vybrané třídy. Ty následně vloží do ArrayListu objektů typu třídy StudentJmeno, který má parametry: String krestni; String prijmeni. Tento ArrayList vloží do ComboBoxu `cmbox_student`.

```
try{
    ConnectionClass connectionClass = new ConnectionClass();
    Connection connection = connectionClass.getConnection();
    Statement statement = connection.createStatement();

    ResultSet resultSet = statement.executeQuery("
SELECT krestniJmeno, prijmeni FROM studenti s \n" +
"JOIN tridy t ON(s.trida_id=t.trida_id)\n" +
"WHERE trida_jmeno = \""+trida+"\"");

    while(resultSet.next()) {
        String krestni = resultSet.getString("krestniJmeno");
        String prijmeni = resultSet.getString("prijmeni");
        studenti.add(new StudentJmeno(krestni, prijmeni));
    }
}catch (Exception e){
    e.printStackTrace();
}

cmbox_student.getItems().addAll(studenti);
```

Pomocí ComboBoxů a TextFieldu si program nahraje do pomocných proměnných informace, které využije při přidávání záznamu do databáze (pomocné proměnné: `krestni`, `prijmeni`, `test`, `znamka`, `hodnota`). Příklad nahrání informací do proměnné `znamka` a `hodnota`:

```
cmbox_znamka.setOnAction(actionEvent->{
    znamka = cmbox_znamka.getValue();
});
cmbox_hodnota.setOnAction(actionEvent->{
    hodnota = cmbox_hodnota.getValue();
});
```

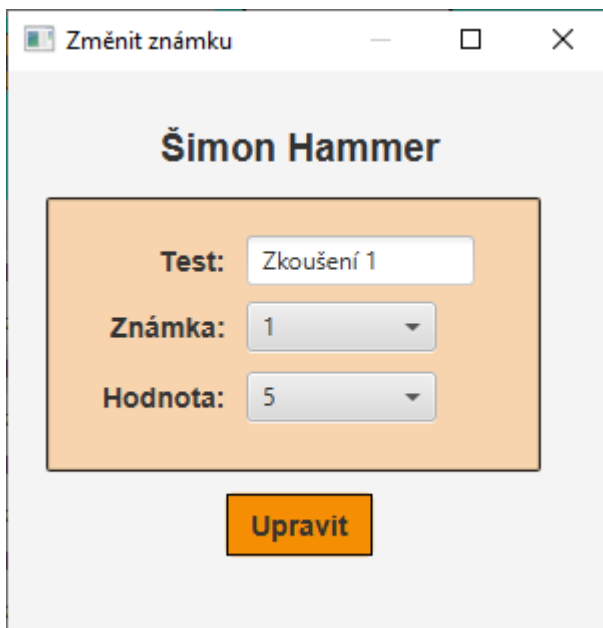
Po stisknutí tlačítka Přidat, se spustí kód pro přidání záznamu známky do databáze.

```
try{
//prida do databaze novou znamku
    ConnectionClass connectionClass = new ConnectionClass();
    Connection connection = connectionClass.getConnection();
    Statement statement = connection.createStatement();
    statement.execute("INSERT INTO zaznamyznamek
(student_id,id_trida_predmet_ucitel,test,znamka,hodnota)\n" +
"SELECT s.student_id,tpu.id_trida_predmet_ucitel,
\""+test+"\", \""+znamka+"\", \""+hodnota+"\" \n" +
"FROM studenti s\n" +
"JOIN trida_predmet_ucitel tpu\n" +
"JOIN predmety p ON (tpu.predmet_id=p.predmet_id)\n" +
"JOIN tridy t ON (tpu.trida_id=t.trida_id)\n" +
"JOIN ucitele u ON (tpu.ucitel_id=u.ucitel_id)\n" +
"WHERE s.krestniJmeno = \""+krestni+"\"
AND s.prijmeni = \""+prijmeni+"\" \n" +
"AND p.predmet_nazev = \""+predmet+"\" \n" +
"AND t.trida_jmeno = \""+trida+"\" \n" +
"AND u.username = \""+username+"\" ");
}catch (Exception e){
    e.printStackTrace();
}
```

6 Změna známky

6.1 Použití

Pro přidání známky slouží třída `ZmenitZnamkuController`. Učitel může změnit jméno testu, známku, její hodnotu nebo vše najednou. Po stisknutí tlačítka Upravit se objeví okénko pro upravení známky. Prostředí se skládá z nápisu o jakého studenta se jedná, TextFieldu `tf_test`, ve kterém učitel může změnit jméno testu, poté ComboBox `cmbox_znamka`, ve kterém může vybrat novou známku, ComboBox `cmbox_hodnota`, ve kterém může zvolit novou váhu známky a tlačítko Upravit, které upraví konkrétní záznam známky tak, jak si učitel navolí.



The screenshot shows a window titled "Změnit známku" (Change grade). Inside the window, the student's name "Šimon Hammer" is displayed at the top. Below the name is a light orange rectangular box containing three input fields: "Test:" with a text field containing "Zkoušení 1", "Známka:" with a dropdown menu showing "1", and "Hodnota:" with a dropdown menu showing "5". Below this box is an orange button labeled "Upravit" (Edit).

6.2 ZmenitZnamkuController

Pro přehlednost program vytvoří pomocné proměnné:

```
String krestni = ZnamkovaniController.krestnizmenitZnamku;
String prijmeni = ZnamkovaniController.prijmenizmenitZnamku;
String test = ZnamkovaniController.testzmenitZnamku;
int znamka = ZnamkovaniController.znamkazmenitZnamku;
int hodnota = ZnamkovaniController.hodnotazmenitZnamku;
int znamkaID = ZnamkovaniController.znamkaIDzmenitZnamku;
```

Poté nahraje informace do ComboBoxů, TextFiledu a Labelu:

```
label_student.setText(krestni+" "+prijmeni);
cmbox_znamka.setValue(znamka);
cmbox_hodnota.setValue(hodnota);
tf_test.setText(test);
cmbox_znamka.getItems().addAll(znamky);
cmbox_hodnota.getItems().addAll(hodnoty);
```

Po stisknutí tlačítka Upravit, se spustí kód pro aktualizaci zvoleného záznamu známky v databázi.

```
//vysledna hodnota, znamka a jmeno testu
int znamkaV = cmbox_znamka.getValue();
int hodnotaV = cmbox_hodnota.getValue();
String testV = String.valueOf(tf_test.getText());

try {
    ConnectionClass connectionClass = new ConnectionClass();
    Connection connection = connectionClass.getConnection();
    Statement statement = connection.createStatement();
    statement.execute("update zaznamyznamek set test='"+testV+"'",
        znamka=" "+ znamkaV +", hodnota = "+hodnotaV+"
        WHERE zaznamZnamky_id=" + znamkaID);
} catch (SQLException e) {
    throw new RuntimeException(e);
}
```

7 Odebrání známky

7.1 Použití

Pro odebrání známky slouží třída `OdebratZnamkuController`. Učitel může odebrat jakýkoliv záznam známky z tabulky. Po stisknutí tlačítka `Odebrat` se objeví okénko, které se učitele zeptá zda chce známku opravdu odebrat. Prostředí se skládá z tlačítka `Ponechat` a `Odebrat`. Při zmáčknutí tlačítka `Odebrat` program odebere záznam známky z databáze. Při zmáčknutí tlačítka `Ponechat` program nové okénko zavře bez jakékoliv jiné operace.

7.2 `OdebratZnamkuController`

Jako první načte id vybraného záznamu známky z tabulky¹¹:

```
StudentZnamkaZaznam selectedItem = (StudentZnamkaZaznam)
mycontroller.getTable().getSelectionModel().getSelectedItem().get(0);
int znamkaID = selectedItem.id;
```

Poté z databáze smaže zvolený záznam známky:

```
try {
    ConnectionClass connectionClass = new ConnectionClass();
    Connection connection = connectionClass.getConnection();
    Statement statement = connection.createStatement();
    statement.execute("delete from zaznamyznamek where zaznamZnamky_id=" + znamkaID);
} catch (SQLException e) {
    throw new RuntimeException(e);
}
```

¹¹<https://stackoverflow.com/questions/44097537/javafx-tableview-get-selected-row-cell-values>

8 Závěr

Program funguje tak, jak fungovat má. Prostředí by mohlo být graficky zpracováno lépe a samotný kód by mohl být naspán jednodušším a přehlednějším způsobem. Při psaní této ročníkové práce jsem se naučil jak pracovat s SQL databází. Chtěl bych poděkovat mému tátovi, který mě naučil jak s SQL databází pracovat a poradil mi při návrhu tabulek. Psaní toho programu mě bavilo a myslím si, že výsledek práce by mohl být i užitečný.

9 Citace

1. MySQL databáze <https://www.mysql.com/>
2. Tutoriál jak pracovat v SQL <https://www.youtube.com/watch?v=7S-tz1z-5bAt=9907s>
3. Obrázek schémata SQL databáze <https://www.dbdesigner.net/>
4. Propojení MySQL databáze s vývojovým prostředím IntelliJ IDEA <https://www.youtube.com/watch?v=8WJ5p3T9Iss>
5. Vytvoření loga: <https://www.freelogodesign.org/>
6. Přesun dat mezi jednotlivými třídami <https://www.youtube.com/watch?v=dFLTha7jH7o>
7. Série tří videí o tom, jak pracovat s TableView 1. <https://www.youtube.com/watch?v=vego7d8B5pyindex=21>
8. 2. <https://www.youtube.com/watch?v=M-kp20qrtLwlist=PL3bGLnkkGnuUQemxQ3ojPcd8B5pyindex=22>
9. 3. <https://www.youtube.com/watch?v=b-0QMxCB2zIlist=PL3bGLnkkGnuUQemxQ3ojPcd8B5pyindex=23>
10. ORACLE dokumentace - TableView <https://docs.oracle.com/javase/8/javafx/user-interface-tutorial/table-view.htmCJAGAAEE>