

# Tabulka v JavaFX

## Připojení k databázi přes JDBC

Alena Buchalcevová

**katedra informačních technologií**  
**Vysoká škola ekonomická v Praze**

**buchalc@vse.cz**





# Model tabulky

- Při použití tabulky v JavaFX aplikaci je dobrou praktikou implementovat třídu, která definuje datový model a má metody a pole pro práci s tabulkou
- V našem příkladu je to třída Person
  - Můžeme ji implementovat:
    - jako statickou vnitřní třídu ve třídě s metodou start (TableJavaFXVPameti)
    - v příkladu je implementována jako veřejná vnější třída
- firstName, lastName a email jsou string property vytvořené proto, aby bylo možné se odkazovat na jednotlivé prvky
- dále jsou pro každou property vytvořeny get a set metody

# Třída PanelTabulky

- V příkladu je vytvořena třída PanelTabulky, která vytvoří tabulku a vloží ji do panelu – FlowPane
- Má metodu getPanel, který panel s tabulkou vrátí
- Data, která se mají v tabulce zobrazit, se drží v atributu **data**, který je typu ObservableList <Person>
- Vytvoří se ObservableArrayList a naplní se 5 záznamy

```
data = FXCollections.observableArrayList(  
    new Person("Jacob", "Smith", "jacob.smith@example.com"),  
    new Person("Isabella", "Johnson", "isabella.johnson@example.com"),  
    new Person("Ethan", "Williams", "ethan.williams@example.com"),  
    new Person("Emma", "Jones", "emma.jones@example.com"),  
    new Person("Michael", "Brown", "michael.brown@example.com")  
);
```

# Třída PanelTabulky/2

- Dále je třeba definovat, která data se budou zobrazovat ve sloupcích – to se dělá pomocí properties definovaných pro každý prvek

```
TableColumn firstNameCol = new TableColumn("First Name");  
firstNameCol.setMinWidth(100);  
firstNameCol.setCellValueFactory(  
    new PropertyValueFactory<>("firstName"));
```

```
TableColumn lastNameCol = new TableColumn("Last Name");  
lastNameCol.setMinWidth(100);  
lastNameCol.setCellValueFactory(  
    new PropertyValueFactory<>("lastName"));
```

```
TableColumn emailCol = new TableColumn("Email");  
emailCol.setMinWidth(200);  
emailCol.setCellValueFactory(  
    new PropertyValueFactory<>("email"));
```

metoda `setCellValueFactory` definuje cell factory pro každý sloupec  
cell factory je implementována pomocí třídy `PropertyValueFactory` class, která použije property `firstName` resp. `lastName` resp. `email` sloupce tabulky jako odkaz na odpovídající metody třídy `Person`

# Třída PanelTabulky/3

- Když je vytvořen datový model a data jsou asociována se sloupci tabulky, můžete přiřadit data do tabulky pomocí metody `setItems`  
`table.setItems(data)`
- Protože objekt `ObservableList` sleduje změny prvků, obsah `TableView` se automaticky aktualizuje

# Další k TableView

Tutoriál

[http://docs.oracle.com/javafx/2/ui\\_controls/table-view.htm#CJAGAAEE](http://docs.oracle.com/javafx/2/ui_controls/table-view.htm#CJAGAAEE)


# Uložení tabulky do databáze SQLite

- Ukázkový příklad TableJavaFXDB

Persisting TableView Data in DataBase

Address Book

First Name	Last Name	Email
Jacob	Smith	jacob.smith@example.com
Isabella	Johnson	isabella.johnson@example.com
Ethan	Williams	ethan.williams@example.com
Emma	Jones	emma.jones@example.com
Michael	Brown	michael.brown@example.com



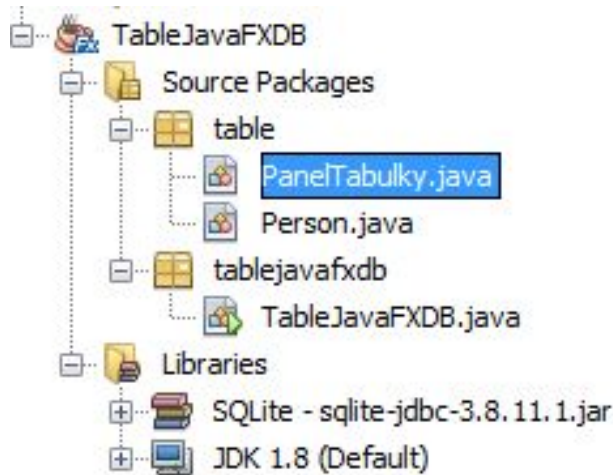
The diagram illustrates the process of persisting data. On the left, there is a stack of three database cylinders. Two green arrows point from the database towards the table, indicating data being written to the database. The table on the right contains the data shown in the table above.



# Připojení SQLite

- Vytvořit knihovnu a vložit do ní jar soubor, který se stáhne na

<https://bitbucket.org/xerial/sqlite-jdbc/downloads/sqlite-jdbc-3.8.11.1.jar>



# Připojení k DB a prvotní naplnění do třídy PanelTabulky

```
try {
    connection = DriverManager.getConnection("jdbc:sqlite:Pokus");
    statement = connection.createStatement();
    //stat.executeUpdate("drop table if exists person");
    statement.executeUpdate("create table if not exists osoby(FirstName varchar(50),LastName
varchar(50),Email varchar(50));");

    // pocatecni ulozeni
    data = observableArrayList(
        new Person("Jacob", "Smith", "jacob.smith@example.com"),
        new Person("Isabella", "Johnson", "isabella.johnson@example.com"),
        new Person("Ethan", "Williams", "ethan.williams@example.com"),
        new Person("Emma", "Jones", "emma.jones@example.com"),
        new Person("Michael", "Brown", "michael.brown@example.com")
    );
    saveData();
}
```

# Načtení dat z DB do observable listu data

```
data = FXCollections.observableArrayList();
ResultSet rs = connection.createStatement().executeQuery("select * from osoby");
    while (rs.next()) {
        data.add(new Person(rs.getString("FirstName"),
rs.getString("LastName"), rs.getString("Email")));
    }
```

# Další na

---

- <http://www.javafxapps.in/tutorial/Persisting-TableView-datas-in-Database.html>