

# Aplikace pro analýzu datového souboru

(Table Viewer)

**Tým TOK**

Pavel Třeštík

Tomáš Ott

Jan Kohout

17. května 2019

# Obsah

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Základní popis</b>                      | <b>2</b>  |
| <b>2</b> | <b>Uživatelská dokumentace</b>             | <b>3</b>  |
| 2.1      | Popis ovládacích prvků aplikace . . . . .  | 3         |
| 2.2      | Očekávané užívání aplikace . . . . .       | 13        |
| <b>3</b> | <b>Programátorská dokumentace</b>          | <b>14</b> |
| 3.1      | Package core . . . . .                     | 14        |
| 3.1.1    | FileLoader.java . . . . .                  | 14        |
| 3.1.2    | JSONLoader.java . . . . .                  | 14        |
| 3.1.3    | LoadedFile.java . . . . .                  | 14        |
| 3.1.4    | Table.java . . . . .                       | 15        |
| 3.2      | Package dependencyTree . . . . .           | 15        |
| 3.3      | Package dialogs . . . . .                  | 15        |
| 3.4      | Package gui . . . . .                      | 15        |
| 3.4.1    | GUI.java . . . . .                         | 16        |
| 3.4.2    | Controller.java . . . . .                  | 16        |
| 3.4.3    | View.java . . . . .                        | 17        |
| 3.5      | Package weekPattern . . . . .              | 19        |
| 3.5.1    | WeekPatternView.java . . . . .             | 19        |
| 3.5.2    | WeekPatternChart.java . . . . .            | 19        |
| 3.5.3    | WeekPatternTask.java . . . . .             | 20        |
| 3.5.4    | WeekPatternRenderer.java . . . . .         | 20        |
| 3.5.5    | WeekPatternToolTipGenerator.java . . . . . | 20        |
| <b>4</b> | <b>Formát JSON config souborů</b>          | <b>21</b> |

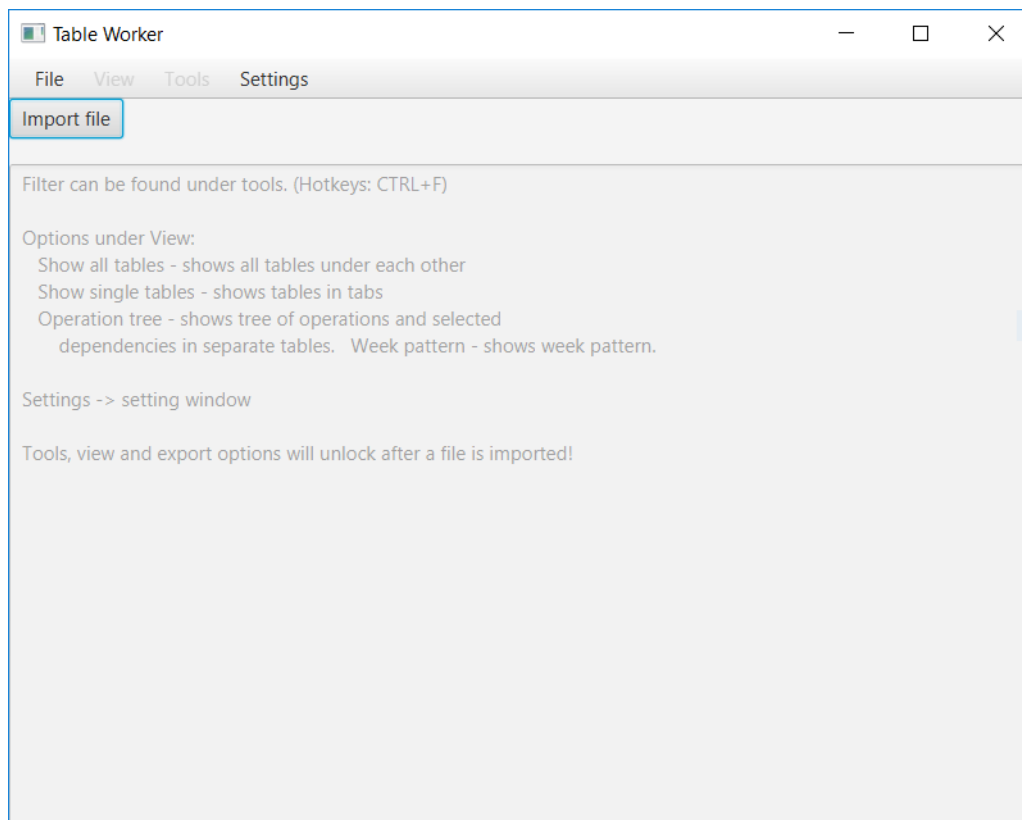
# 1 Základní popis

Tato aplikace byla napsána jako semestrální práce k předmětu KIV/ZSWI, pro externího zadavatele. Externím zadavatelem je firma Certicon a tato aplikace má sloužit jako tabulkový manager pro určité typy a formáty souborů.

## 2 Uživatelská dokumentace

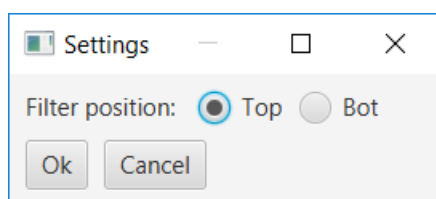
### 2.1 Popis ovládacích prvků aplikace

Po spuštění aplikace se zobrazí úvodní obrazovka, na které se nachází nápověda k základní funkcionalitě aplikace.



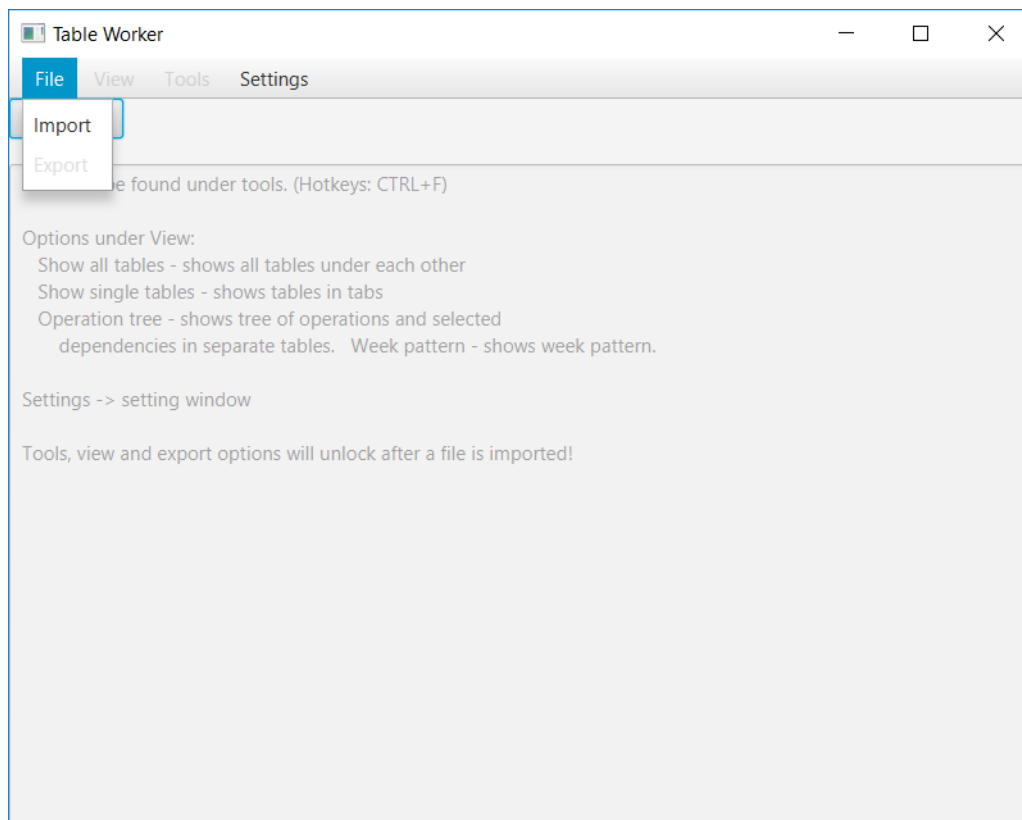
Obrázek 1: Úvodní obrazovka aplikace

Volbou **Settings window** z menu **Settings** se objeví okno s možným nastavením aplikace. V nastavní lze určit pozici filtru v hlavním okně.



Obrázek 2: Okno nastavení

Volbou **Import** z menu **File** lze vybrat soubor pro import.



Obrázek 3: Menu File před importem

Po načtení importovaného souboru se zobrazí pohled na všechny tabulky jako na následujícím obrázku.

[illegible]

Obrázek 4: Stav aplikace po importu

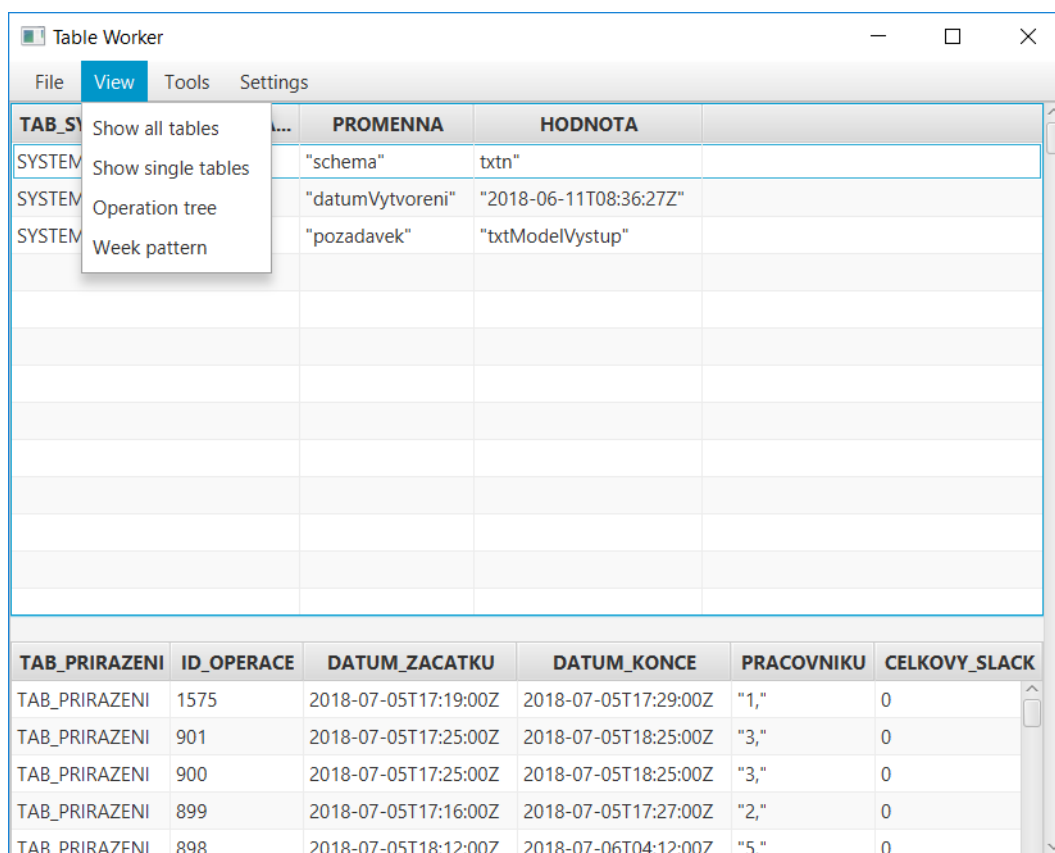
V menu **Tools** lze zvolit **Filter** bar pro zobrazení/skrytí pole pro filtrování. Pro tuto volbu lze využít též klávesovou zkratku **ctrl + F**.

[illegible]

Obrázek 5: Menu Tools







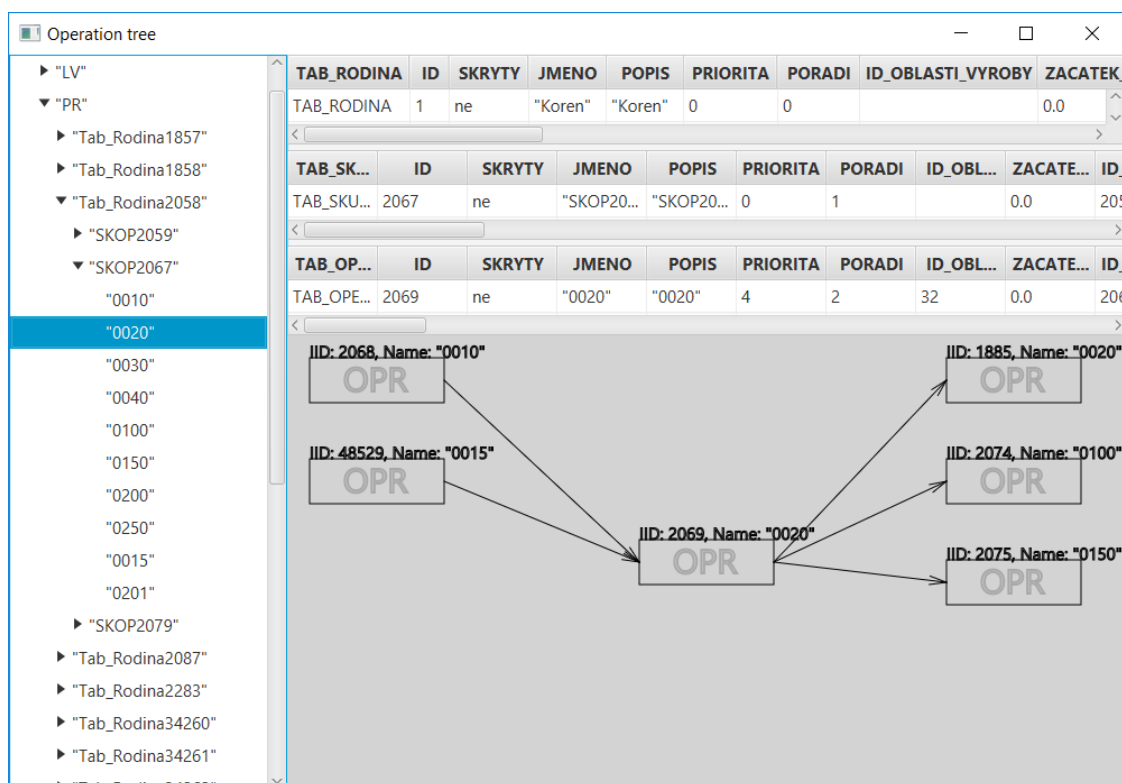
Obrázek 7: Menu View

Volbou **Show all tables** z menu **View** se změní zobrazení tabulek do jednotlivých záložek.

[illegible]

Obrázek 8: Zobrazení jednotlivých tabulek v záložkách

Volbou **Operation tree** z menu **View** se zobrazí nové okno reprezentujúci strom operácií.



Obrázek 9: Strom operací

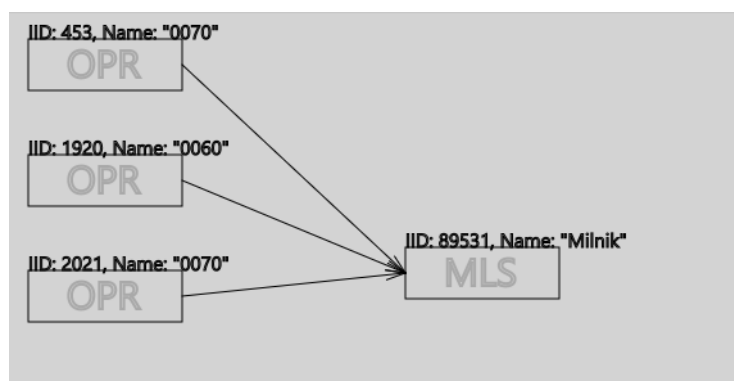
V okně Stromu operací se zobrazují závislosti mezi milníky, rodinami, skupinami a operacemi (viz obrázky 10 - 12).



Obrázek 10: Značení rodiny v závislostech

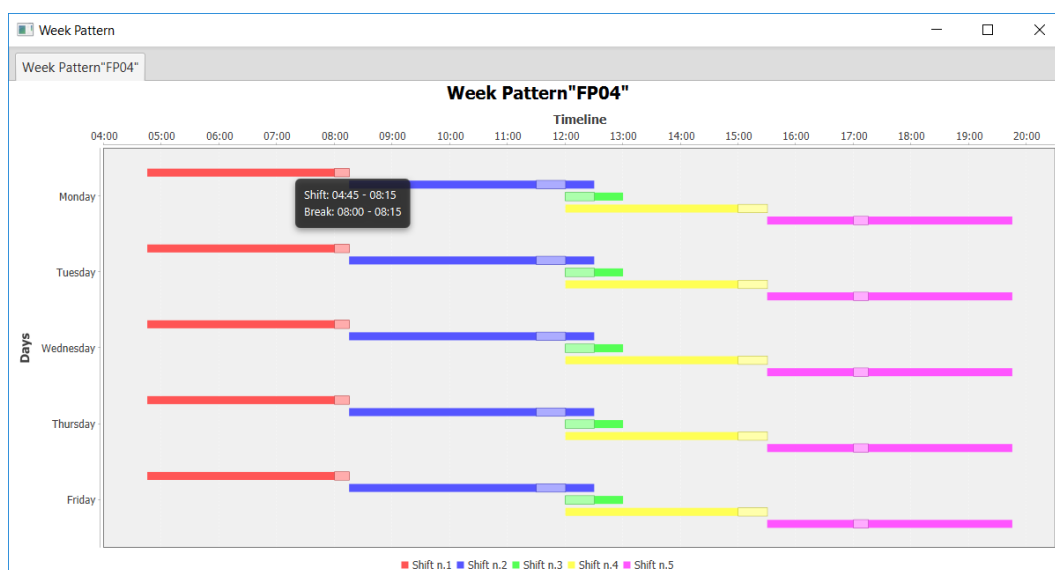


Obrázek 11: Značení skupiny v závislostech



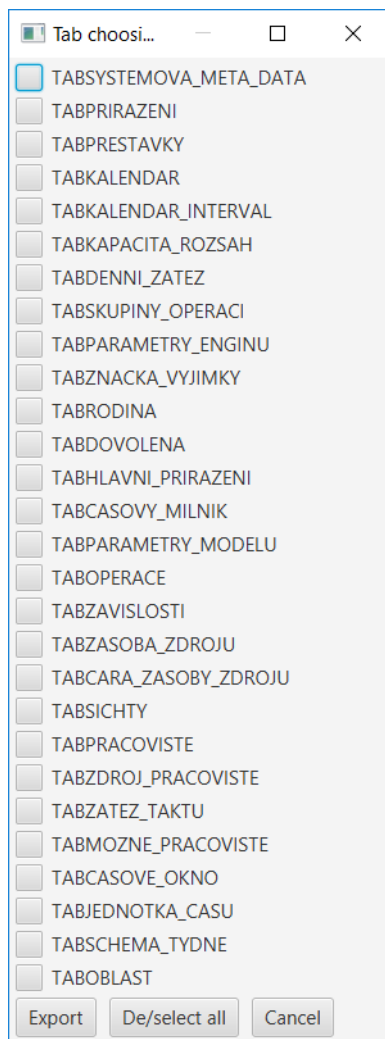
Obrázek 12: Značení milníku v závislostech

Volbou **Week pattern** z menu **View** se zobrazí nové okno schéma týdne. Pokud v souboru existuje více kalendářů či více schémat týdne, pak je každý graf vykreslen na samostatnou záložku.

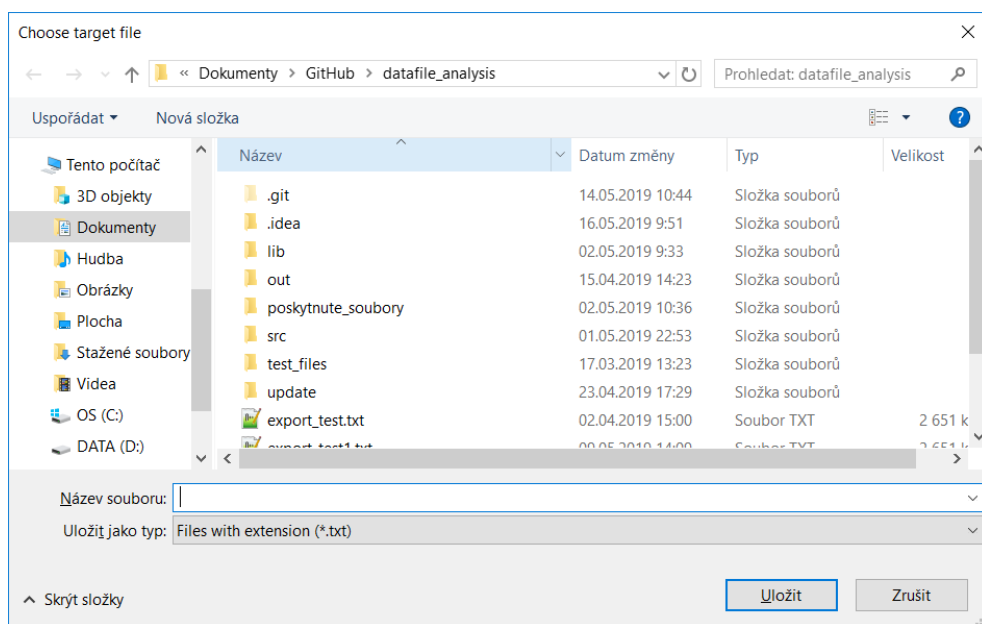


Obrázek 13: Week pattern

Volbou **Export** z menu **File** lze vybrat, které tabulky budou exportovány. Po zvolení požadovaných tabulek a potvrzení tlačítkem **Export**, je zapotřebí už jen vybrat lokaci a název pro exportovaný soubor.



Obrázek 14: Vybírání tabulek k exportu



Obrázek 15: Okno výběru lokace exportu

## 2.2 Očekávané užívání aplikace

## 3 Programátorská dokumentace

### 3.1 Package core

Nachází se zde třídy obstarávající hlavní funkce aplikace.

#### 3.1.1 FileLoader.java

Třída obstarávající načtení a zapsání do souboru. Obsahuje dvě statické metody.

**loadFile(File f)** Metoda načte soubor předaný parametrem. Tento soubor je načítán pomocí `BufferedReader`. Během načítání souboru metoda vytváří instance třídy `Table` (popsána dále). Metoda očekává korektní formát vstupního souboru a žádným způsobem ho nekontroluje. Při chybě načítání souboru metoda vypíše defaultní `printStackTrace` pro `IOException`. Vyjímky při načítání by bylo vhodné lépe ošetřit.

**exportFile(File f, List<Table> tables)** Metoda zapisuje do souboru předaného parametrem. Zapisuje tabulky také předané parametrem. K zápisu využívá `BufferedWriter`. Po každé tabulce se do souboru vypíše jedna prázdná řádka. Stejně ošetření vyjímek jako u načtení.

#### 3.1.2 JSONLoader.java

Tato třída slouží pro práci se souborem "configuration.json". Pokud nebyl tento soubor nalezen, nebo je zapsán ve špatném formátu, tak se použijí přednastavené názvy.

**loadJSONFile()** Metoda slouží pro přečtení obsahu konfiguračního souboru. Tento obsah pak rozdělí na objekty ("weekPattern" a "operationTree"). Ty dále předá metodě `readFromObject(...)`. V případě, že při čtení je uhozena výjimka `JSONException` (Data byla zapsána ve špatném formátu), metoda se ukončí a jsou použity již přednastavené názvy.

**readFile()** metoda je použita pro načtení souboru "configuration.json". Načtené řádky souboru spojí jako jeden `String` objekt a vrátí ho volající metodě.

**readFromObject(String[] tableNames, String[] jsonTableNames, String[] columnNames, JSONObject subObject)** Metoda projde všechny názvy tabulek z "jsonTableNames". Při nalezení přiřadí obsah z jeho atributu "tableName" na příslušný index pole "tableNames". Také použije atribut "attributeNames" a přiřadí jeho obsah do pole "columnNames".

### 3.1.3 LoadedFile.java

**Atributy** Třída má dva atributy. Atribut **fileTables** - list tabulek (Table) načtených ze souboru a **tableCount** - počet načtených tabulek.

**Loadedfile()** Jednoduchý konstruktor inicializující atribut *fileTables*.

**addTable(Table table)** Metoda přidá tabulku do listu tabulek a zvýší počet tabulek o 1.

**getFileTables()** Getter pro atribut *fileTables*.

**getTableCount()** Getter pro atribut *tableCount*.

### 3.1.4 Table.java

**Atributy** Třída obsahuje 3 atributy. String **tableName** - název tabulky. String[] **header** - header tabulky (názvy sloupců). ObservableList<String[]>**data** - data tabulky.

**Table()** Jednoduchý konstruktor inicializující *data*.

**addRow(String[] row)** Přidá záznam do tabulky (do *data*).

**setHeader(String[] header)** Nastaví header tabulky.

**createGUITable(Controller controller)** Metoda vrací instanci TableView z JavaFX. Metoda vytvoří grafickou reprezentaci dat z *header* a *data*. Metoda také nastaví možnost editaci údajů v tabulce. Pokud údaj v tabulce měl na alespoň na začátku nebo na konci " ", tak edit nastavený v této metodě automaticky doplní uvozovky při dokončení editace. Mimo editaci tato metoda také kontroluje korektnost zadaného id v případě jeho editace, ale pouze pokud je editováno id ze souboru, který je částí načtené konfigurace.



**getTableName()** Getter pro *tableName*.

**setTableName(String tableName)** Setter pro *tableName*.

**getHeader()** Getter pro *header*.

**getData()** Getter pro *data*.

**toString()** Metoda vrací tabulku v textové podobě, která by měla být ve stejném formátu jako v načítaném souboru.

## 3.2 Package dependencyTree

## 3.3 Package dialogs

## 3.4 Package gui

Package obsahuje tři hlavní třídy pro grafické rozhraní. Původně třídy měly tvořit MVC návrh, ale tato struktura není velmi dobře dodržena.

### 3.4.1 GUI.java

Hlavní třída spouštící aplikaci. Obsahuje metodu **start(Stage primaryStage)** - jedná se metodu vytvářející hlavní okno aplikace, které je pak spuštěno ze třídy **main**.

### 3.4.2 Controller.java

Třída, která obstarává akce GUI.

**Atributy** Obsahuje atributy: Parent **parent** - scéna rodičovského okna, LoadedFile **loadedFile** - instance třídy načteného souboru, která obsahuje instance tabulek a String **fileExtension**, **fileName** - atributy obsahující příponu a jméno načteného souboru. Používaná pouze **fileExtension** při exportu pro předvyplnění přípony souboru.

**Controller(Parent parent)** Konstruktor inicializující atribut *parent*.

**serviceImport()** Metoda obsluhuje akci načtení souboru. Vrací true/ false podle toho, zda byl import úspěšný.

**serviceExport()** Metoda obsluhuje akci exportu do souboru. Metoda nejdříve vytvoří okno se seznamem načtených tabulek a umožní uživateli zvolit tabulky k exportu. Akce které toto okno nabízí jsou také obslouženy v této metodě.

**exportChooser(Stage stage)** Metoda je pomocná k metodě předešlé. Vytváří dialog pro výběr adresáře, do kterého se vytvoří exportovaný soubor.

**serviceFilter(tableView, TextField flt)** Metoda vytvoří filtrování tabulky získané prametem **tableView** a filtruje pomocí textového pole **flt**. Metoda nastavuje filtr tak, aby se filtrovalo po každé změně ve vstupním textovém poli.

**setTableHeight(List<TableView>tables, int height)** Metoda má nastavovat výšku tabulek v zobrazení všech tabulek pod sebou. Metoda není využívána, protože tato funkce způsobovala problémy při nastavení výšky tabulek na větší hodnoty a proto byla tato funkce nepoužita, ačkoliv stále je v kódu a po úpravách by mohlo být možné tuto funkci a metodu použít.

**createRightClickAction(tableView<String[]>tv, DependencyView dv, View view)** Metoda vytváří a obstarává funkce kontextového menu, které se zobrazí při pravém kliknutí myši na záznam v tabulce. Funkce zobrazení v Dependencyview je vytvořena pouze u tabulek, které jsou načteny z konfigurace závislých tabulek.

**showItemInTree(tableView<String[]>tv, DependencyView dv, TableRow row, View view)** Metoda zobrazí zvolený záznam v zobrazení závislostí Dependencyview. Tato metoda je pomocná metoda k předešlé metodě **createRightClickAction(...)**.

**addCopyLine(tableView<String[]>tv, TableRow<String[]>row)** Metoda přidá kopii vybraného záznamu na konec tabulky. Pomocná metoda k metodě **createRightClickAction(...)**.

**isValidID(String id)** Metoda kontroluje jestli se zadané id (předané prametem) již nevyskytuje tabulce či tabulkách závislých.

**getUnavailableIDs()** Metoda vrací list nedostupných ID. Tento list je využíván předešlou metodou **isValidID(...)**.

**areAllBoxesSelected(CheckBox[] boxes)** Pomocná metoda pro vybrání/odvybrání všech checkboxů pomocí tlačítka při exportu.

**getLoadedFile()** Getter pro *loadedFile*.

### 3.4.3 View.java

Třída, která vytváří GUI prvky (ideálně při dodržení mvc, má minimální logiku k prvkům).

**Atributy** Vzhledem k tomu, že tato třída vytváří prvky GUI má značné množství atributů, které jsou celkem slušně popsány v javadoc. Nejvýznamnější jsou pravděpodobně VBox **pane** ve kterém jsou všechny prvky GUI, Controller **controller**, AnchorPane **shower**, DependencyView **dv**.

**View()** Konstruktor volající metodu, která dále rekurzivně volá další metody pro tvorbu GUI. Dále inicializuje *controller* a načítá tabulky z JSON konfiguračního souboru, které jsou použity pro zobrazení závislostí a week pattern.

**asParent()** Metoda vrací *pane* jako typ Parent.

**setupPane()** Metoda inicializuje *pane*, volá další metody, které vytvářejí další prvky GUI a přidává je do *pane*.

**setupWelcomePane()** Metoda vytvoří rozložení, které se zobrazí při zapnutí aplikace.

**showSettingPane()** Metoda zobrazí okno nastavení a dále obstarává jeho rozložení a logiku (ta by měla být v *controller*). Metoda má navíc "předpřipravený" přidání nastavení výšky tabulek v zobrazení podsebou.

**setupMenu()** Metoda vytvoří menu aplikace. V metodě se volají akce na jednotlivé položky menu a pro náročnější operace je vytvořené vlastní vlákno.

**setupFilter()** Metoda vytvoří rozložení pro filtr.

**addFilterToScene()** Metoda přidá filtr do scény.

**removeFilterFromScene()** Metoda odstraní filtr ze scény.

**setShowerContent(Node nd)** Metoda nastaví obsah atributu *shower*, který slouží k zobrazení všech pohledů v hlavním jevišti.

**setupTabs()** Metoda vytvoří a naplní rozložení pro jednotlivé tabulky přepínané pomocí záložek. Také obstarává nastavení filtru na zrovna zobrazenou tabulku.

**filterAllTables()** Metoda nastavuje filtr při zobrazení všech tabulek pod sebou.

**setupAllTablesView()** Metoda vytvoří a naplní rozložení všech tabulek podsebe. Všem tabulkám nastaví velikost 400px.

**getOperationTreeMenuItem()** Metoda vrací položku menu pro strom operací.

**getWeekPatternMenuItem()** Metoda vrací položku menu pro week pattern.

**getTableIndexByName(String tabName)** Metoda vrací index tabulky podle jména získaného parametrem.

**getTableIndexInTabs(String tabName)** Metoda vrací index záložky tabulky se jménem získaným z parametru (pouze u zobrazení jednotlivých tabulek). Jedná se o pomocnou metodu k předcházející metodě *getTableIndexByName(...)*

**getTableIndexInScrollPane(String tabName)** Metoda vrací index tabulky se jménem získaným z parametru (pouze u zobrazení všech tabulek podsebou). Jedná se o pomocnou metodu k metodě *getTableIndexByName(...)*

**getTableContentByIndex(int tabIndex)** Metoda vrací data tabulky s indexem získaným z parametru.

**getTableAttributeIndexes(int tableIndex, String[] columnNames)** Metoda vrací index/y sloupců, které obsahují určité jména, získané z parametru **columnNames**, v tabulce s indexem **tableIndex**.

**getTableAttributeIndexesFromColumns(List<TableColumn>columns, String[] columnNames)** Metoda vrací index/y sloupců, obsahující jméno/a z parametru **columnNames**, z množiny sloupců získané parametrem **columns**.

**getTableCollums(int tableIndex)** Metoda vrací sloupce tabulky podle indexu získaného z parametru.

### 3.5 Package weekPattern

Zde se nachází třídy pro vizualizaci schémat týdne.

#### 3.5.1 WeekPatternView.java

Třída reprezentuje hlavní okno pro vizualizaci schémat týdne.

#### 3.5.2 WeekPatternChart.java

Třída reprezentuje jednotlivé diagramy schémat týdne.

#### 3.5.3 WeekPatternTask.java

Třída dědí od třídy **Task** a doplňuje její vlastnosti o délku trvání pauzy v šichtách.

#### 3.5.4 WeekPatternRenderer.java

Třída dědí od třídy **GanttRenderer** a přepisuje její metodu **drawTask()** pro přidání podpory pro vykreslování pauz do jednotlivých šicht v diagramu.

#### 3.5.5 WeekPatternToolTipGenerator.java

Třída dědí od třídy **IntervalCategoryToolTipGenerator** a přepisuje její metodu **crateItemArray()**, aby vytvářela popis jednotlivých šicht ve formátu jaký můžete vidět na obrázku 13.

## 4 Formát JSON config souborů

Pro zavedení vlastních názvů tabulek a jejich atributů je použit soubor "configuration.json". Ten obsahuje dva hlavní objekty ("operationTree" a "weekPattern") :

```
{ "operationTree": {  
.....  
},  
"weekPattern": {  
.....  
}  
}
```

Objekt "operationTree" dále obsahuje tyto komponenty:

```
"family": {  
"tableName": "TAB_RODINA",  
"attributeNames": ["ID", "ID_VLASTNIK_RODINA", "JMENO"]  
},  
"operationGroup": {  
"tableName": "TAB_SKUPINY_OPERACI",  
"attributeNames": ["ID", "ID_VLASTNIK_RODINA", "JMENO"]  
},  
"operation": {  
"tableName": "TAB_OPERACE",  
"attributeNames": ["ID", "ID_VLASTNIKA_SKOP", "JMENO"]  
},  
"milestone": {  
"tableName": "TAB_CASOVY_MILNIK",  
"attributeNames": ["IID", "ID_VLASTNIK_RODINA", "JMENO"]  
},  
"dependency": {  
"tableName": "TAB_ZAVISLOSTI",  
"attributeNames": ["ID", "ID_AKTIVITY_OD", "ID_AKTIVITY_DO"]  
}  
}
```

A objekt "weekPattern" obsahuje:

```
"calendar": {  
"tableName": "TAB_KALENDAR",  
"attributeNames": ["ID"]  
}
```

```

},
"calendarInterval": {
  "tableName": "TAB_KALENDAR_INTERVAL",
  "attributeNames": ["ID_KALENDARE", "ID_VZORCE_TYDNE"]
},
"weekPattern": {
  "tableName": "TAB_SCHEMA_TYDNE",
  "attributeNames": ["ID", "SEZNAM_PONDELNICH_SICHT",
    "SEZNAM_UTERNICH_SICHT",
    "SEZNAM_STREDECNICH_SICHT", "SEZNAM_CTVRTECNICH_SICHT",
    "SEZNAM_PATECNICH_SICHT", "SEZNAM_SOBOTNICH_SICHT",
    "SEZNAM_NEDELNICH_SICHT"]
},
"shift": {
  "tableName": "TAB_SICHTY",
  "attributeNames": ["SEZNAM_SICHT", "CAS_ZACATKU", "CAS_KONCE", "MA_PAUZU",
    "ZACATEK_PAUZY", "KONEC_PAUZY"]
}

```

Každá komponenta obsahuje dva atributy. První ("tableName") je použit pro název tabulky, která komponentu nějak popisuje. A druhý ("attributeNames") nese názvy všech potřebných atributů tabulky.

Je vřele doporučeno dodržovat názvy objektů, komponent a atributů! Při špatném zapsání bude configurační ignorován a budou použity přednastavené názvy.