

# Pokladna Dokumentace

Šišma Vojtěch  
5.4.2021

## Obsah

1.	Zadání .....	1
2.	Použité zkratky .....	2
3.	Program .....	3
3.1.	Databáze.....	3
3.2.	Server.....	3
3.2.1.	Komunikace s databází .....	3
3.2.2.	Provádění SQL příkazů .....	3
3.2.3.	Objednavka.....	4
3.2.4.	Uloziste .....	4
3.3.	Shared.....	4
3.3.1.	Třídy Polozka a Pridavek.....	4
3.3.2.	Rozhraní Polozky, Uloziste a Objednavka.....	4
3.4.	Pokladna .....	5
3.4.1.	Hlavní Menu .....	5
3.4.2.	Vytvoření objednávky.....	5
3.4.3.	Vypsání objednávky.....	6
3.5.	Admin .....	8
3.5.1.	Hlavní Menu .....	8
3.5.2.	Přidání nové položky/přídavku, editace .....	8
3.5.3.	Mazání položek/přídavků .....	9

## 1. Zadání

### **Pokladna klient-server**

*Uprav projekt Pokladna #2 (opakování 2. ročníku) do podoby klient-server řešení. Server bude komunikovat s databází (produkty a objednávky) a tato data poskytovat klientům.*

*Klienty budou dva:*

*Admin – pro úpravy produktů (zadávání nových, úprava stávajících, zrušení nabídky produktu) a pro prohlížení.*

*Pokladna – pro zadávání objednávek a zpětné prohlížení.*

### **Poznámky k realizaci**

*Aplikace klientů budou mít grafické rozhraní.*

*Klienty uchovávají pouze nezbytné minimum dat v operační paměti.*

*Server zajistí konkurenční přístup k datům uchovávaným v operační paměti.*

*Server každou hodinu uvolní z operační paměti objednávky starší než 10 minut.*

*Databázi si navrhni podle potřeb tvého návrhu při dodržení pravidel návrhu DB.*

### **Dokumentace**

*V dokumentaci v přiměřeném rozsahu popiš postup tvorby / úpravy aplikace, zvolená řešení a zdůvodni jejich volbu.*

*Součástí dokumentace bude E-R diagram databáze a přílohou skript pro její import (ideálně s výchozím naplněním daty).*

Viz MS Teams

## 2. Použité zkratky

GUI    grafické uživatelské rozhraní

SQL    Structured Query Language

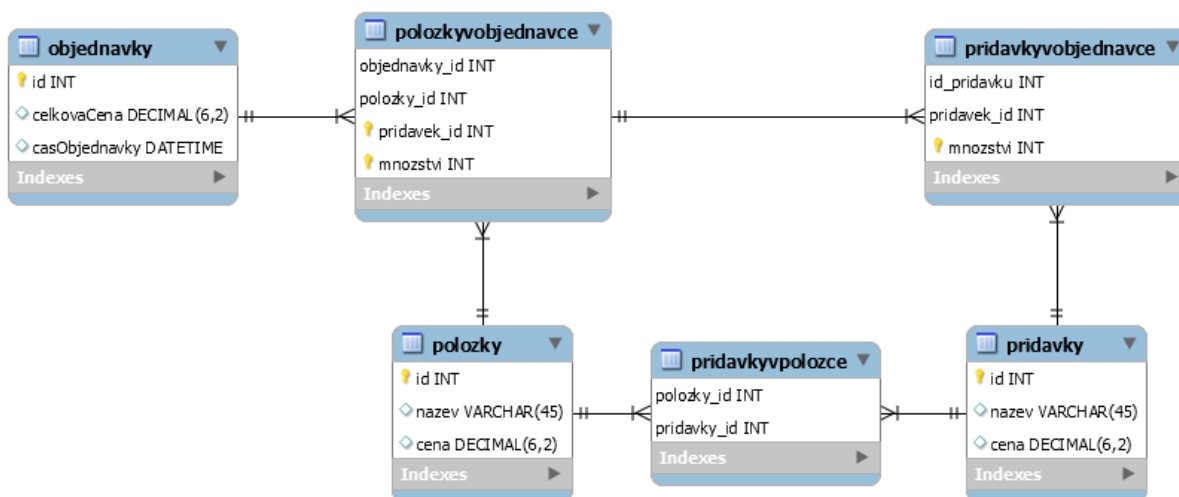
RMI    remote method invocation

### 3. Program

Program se skládá ze tří nezávislých částí (balíčků) – pokladna, admin a server, a sdílené části – shared. Rozhraní serveru využívá pouze konzole/terminálu, admin a pokladna mají vlastní GUI. Server komunikuje s MySQL databází a klientům umožňuje vzdáleně využívat metody jeho objektů pomocí RMI.

#### 3.1. Databáze

Je použita databáze MySQL složená ze šesti tabulek. V tabulce objednávky je uloženo id objednávky celková cena objednávky a čas uskutečnění objednávky. V tabulce položkyvobjednavce jsou cizí klíče objednávky\_id z tabulky objednávky a položky\_id z tabulky položky, dále je zde uveden počet výskytů v objednávce(množství) a pridavek\_id. V tabulce položky je uloženo id, název a cena dostupných položek. V tabulce pridavkyvobjednavce je uveden cizí klíč id\_pridavku z tabulky pridavky a pridavek\_id z tabulky položkyvobjednavce. V tabulce pridavkyvpolocce je cizí klíč položky\_id z tabulky položky a pridavky\_id z tabulky pridavky. V tabulce pridavky je uvedeno id název a cena dostupných přidavků.



Obrázek 1 – ER diagram

#### 3.2. Server

##### 3.2.1. Komunikace s databází

Připojení k Mysql databázi zajišťuje třída Db tvořená návrhovým typem Singleton za pomoci knihovny MySQL Connector. Metoda `getConnection()` s návratovým typem `Connection` zajistí spojení s databází specifikovanou v proměnné `cs` za použití uživatele root a prázdného hesla.

##### 3.2.2. Provádění SQL příkazů

Provádění SQL příkazů zajišťuje třída Položky. Ve všech metodách je nejdříve získána instance třídy Db a poté je zavolána metoda `getConnection()`. Poté je vykonán SQL statement pomocí metody `executeQuery()`, popřípadě, pokud potřebujeme do příkazu vložit proměnné, je vytvořen prepared statement, kterému jsou nastaveny parametry a poté je vykonán pomocí metody `executeUpdate()`. Výsledky SQL příkazů jsou pak dostupné pomocí metod `getString()`, `getInt()` atd. třídy `ResultSet`. Vyjímky jsou ošetřeny pomocí chráněných bloků.

Vypsání dostupných položek a přidavků zajišťují metody `getPoložky()`, která navrací LinkedList se všemi položkami a `getPridavky(int id)`, která vrací LinkedList se všemi přidavky specifikované položky pomocí id.

Zápis objednávky provádí metoda `writeObjednavka(Objednavka objednavka)`, pokud se zápis provedl, vypíše id zapsané objednávky do konzole a vrátí true.

Id předchozí objednávky lze získat pomocí metody `getPrevId()`.

Zapsání nových položek/přídavků zajišťují metody `writePolozka(String nazev, int cena)` a `writePridavek(String nazev, int cena)`, pokud se zápis provedl, vrátí se true.

Získání všech dostupných položek je zajištěno pomocí metody `getPridavkyAll()`, která vrátí ArrayList se všemi přídavky.

Pomocí metod `getPolozka(int id)` a `getPridavek(int id)` je možné získat položku/přídavek na základě zadaného id.

Úpravu položek/přídavků zajišťují metody `updatePolozka(Polozka polozka)` a `updatePridavek(Pridavek pridavek)`.

Mazání položek/přídavků zajišťují metody `deletePolozka(int id)` a `deletePridavek(int id)`.

### 3.2.3. Objednavka

Třída `Objednavka` rozšiřuje `UnicastRemoteObject` a implementuje rozhraní `objednavka` z balíčku `shared`. `Objednavka` obsahuje id, čas objednávky, cenu, celkovou cenu a `LinkedList` s položkami a k těmto proměnným příslušné `get` a `set` metody. V metodě `getCelkovaCena()` se za použití `for` cyklu vypočítá celková cena objednávky. Konstruktor je rozšířen o inicializaci proměnné `casObjednavky`, která je nastavena na aktuální čas (čas vytvoření objednávky). Vzhledem k použití RMI `objednavka` obsahuje metodu `obnov`, která zajistí obnovení všech atributů na defaultní hodnoty.

### 3.2.4. Uloziste

Třída `Uloziste` zajišťuje ukládání provedených objednávek na serveru. Každou hodinu jsou objednávky starší než 10 minut smazány z paměti, což je zajištěno pomocí tříd `Timer` a `TimerTask`, samotné mazání zajišťuje metoda `odeber(int id)`. Třída dále obsahuje seznam objednávek implementovaný pomocí `HashMap`, kde je jako klíč použito id objednávky a metody `getObjednavka(int index)` pro získání dané objednávky, `pridej(Objednavka objednavka)` a `getPocet()` pro získání počtu uložených objednávek.

## 3.3. Shared

### 3.3.1. Třídy Polozka a Pridavek

Třídy `polozka` a `pridavek` obsahují id, název, cenu, množství a celkovou cenu, `polozka` má navíc `LinkedList` s `přídavky` a `přídavek` má id nadřazeného produktu (`parent`). Třídy obsahují `get` a `set` metody pro nastavení a získání hodnot těchto proměnných.

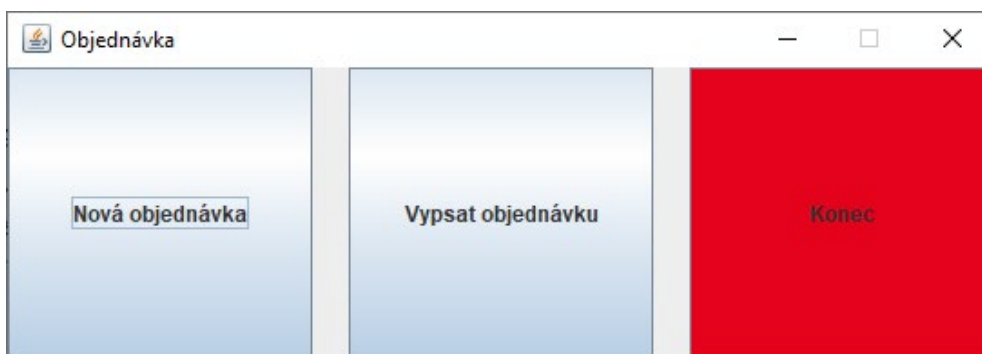
### 3.3.2. Rozhraní Polozky, Uloziste a Objednavka

Rozhraní `Polozky`, `Uloziste` a `Objednavka` rozšiřují rozhraní `Remote` (k určení, že metody lze vyvolat ze vzdálené aplikace) a obsahují deklaraci metod daných tříd.

### 3.4. Pokladna

#### 3.4.1. Hlavní Menu

Po spuštění pokladny se zobrazí nabídka s výběrem akcí.

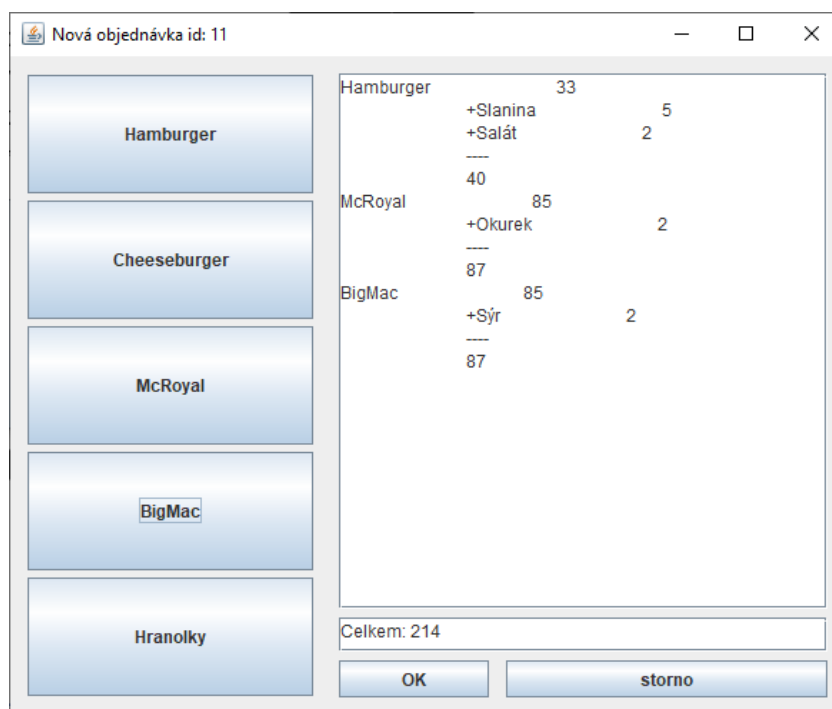


Obrázek 2 - hlavní menu

Pro zobrazení hlavního menu aplikace je použita třída MainFrame. Jednoduché rozložení tří tlačítek je docíleno použitím grid layoutu. Text tlačítek je nastaven pomocí metody `.setText(String)`. Každému tlačítku je přidán ActionListener pomocí metody `.addActionListener(ActionListener)`. Po kliknutí na tlačítko *Nová objednávka* se zobrazí okno pro [vytvoření objednávky](#), po kliknutí na *Vypsát objednávku* se zobrazí okno na [vypsání objednávky](#) a po kliknutí na *Konec* se program ukončí pomocí metody `.exit(int)`.

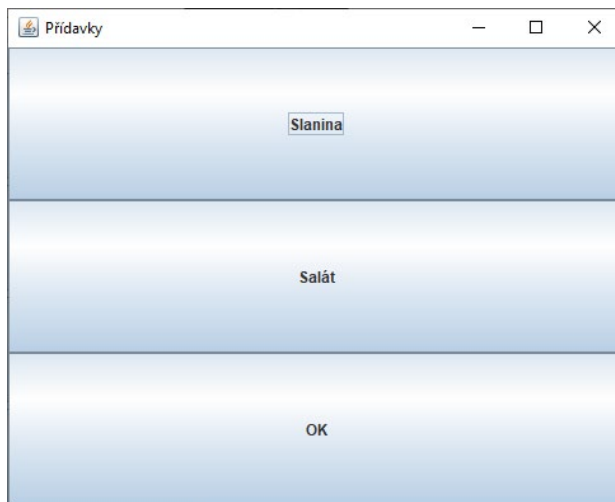
#### 3.4.2. Vytvoření objednávky

Okno vytvoření objednávky je vytvořeno ve třídě ObjednavkaFrame. Při vytvoření okna se pomocí metody `lookup()` třídy Naming získá ze serveru instance třídy Polozky a Objednavka, dále se pomocí metody `getPrevId()` získá id předchozí objednávky a navýší se o 1, čímž získáme id aktuální objednávky. Pomocí metody `getPolozky()` třídy Polozky se získají produkty z nabídky a pomocí foreach cyklu se vytvoří tlačítka pro výběr těchto produktů.



Obrázek 3 - zadávání objednávky

Položka má pomocí metody `.addActionListener(ActionListener)` přidán ActionListener, který vyvolá metodu `actionPerformed()`. V této metodě je vytvořeno nové okno s výběrem přídatků příslušících dané položce.

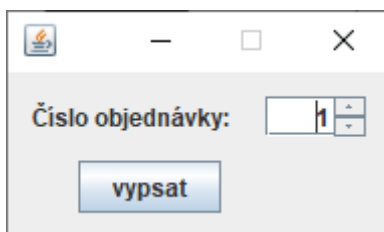


Obrázek 4 - výběr přídatků

Po vybrání přídatku se přídatek přidá do LinkedListu přídatků dané položky pomocí metody `pridej()` a po ukončení zadávání přídatků se zavolá metoda `zapis(Polozka p)`, pomocí které se položka i s přídatky zapíše do JTextArea pomocí metody `append(String)` v okně objednávky.

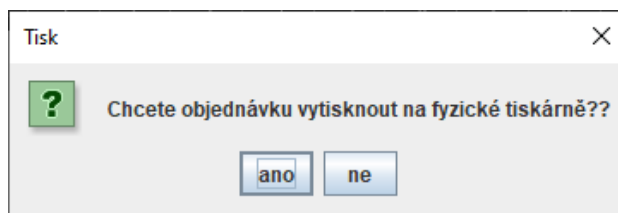
### 3.4.3. Vypsání objednávky

Výpis objednávek je zajištěn pomocí třídy VypisFrame. Nejdříve je vytvořeno okno s výběrem čísla objednávky, číslo objednávky se zadává do JSpinneru ze kterého se získá pomocí metody `getValue()`.



Obrázek 5 - výběr objednávky k vypsání

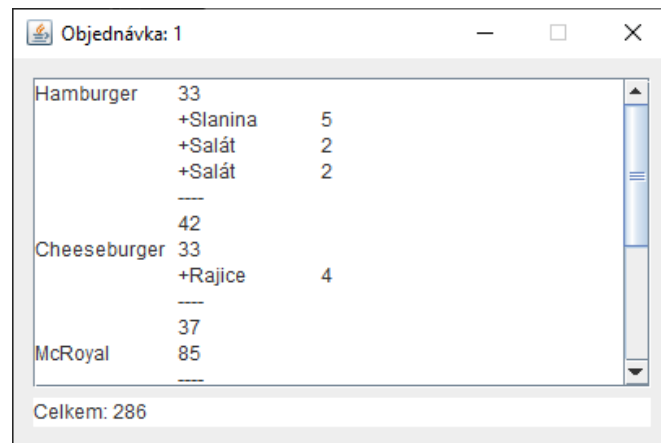
Po kliknutí na tlačítko ok se vyvolá metoda `vypis(int id)`, proběhne pokus o získání objednávky z paměti, pokud se objednávka v paměti nenachází, získá se z databáze pomocí metody třídy položky `getObjednava(int index)`.



Obrázek 6 - dotaz na tisk

Následně je obsah objednávky vypsán do JTextArea a do druhé JTextArea je vypsána celková cena objednávky.





Hamburger	33	
	+Slanina	5
	+Salát	2
	+Salát	2
	----	
	42	
Cheeseburger	33	
	+Rajice	4
	----	
	37	
McRoyal	85	
	----	
Celkem: 286		

Obrázek 7 - výpis objednávky

Zároveň je také pomocí metody `tiskni()` třídy `Printer` zapsána objednávka do textového souboru pomocí třídy `FileWriter`.

### 3.5. Admin

#### 3.5.1. Hlavní Menu

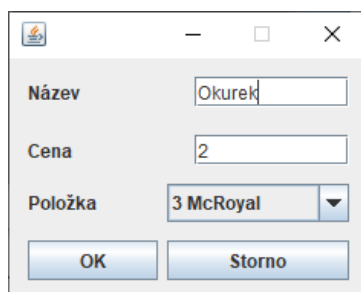
Po spuštění aplikace se zobrazí nabídka s výběrem akcí.



Obrázek 8 - hlavní menu (admin)

Pro zobrazení hlavního menu aplikace je použita třída `MainFrame`. Rozložení tlačítek je docíleno použitím `grid layout`. Text tlačítek je nastaven pomocí metody `setText(String)`. Každému tlačítku je přidán `ActionListener` pomocí metody `addActionListener(ActionListener)`. Po kliknutí na tlačítka *Nová položka*, *Nový přídavek*, *Upravit položku*, *Upravit přídavek* se zobrazí okno pro přidání nové položky/přídavku a editaci, po kliknutí na *Vypsat objednávku* se zobrazí okno na vypsání objednávky, které je shodné s vypsáním v pokladně a po kliknutí na *Konec* se program ukončí pomocí metody `exit(int)`.

#### 3.5.2. Přidání nové položky/přídavku, editace

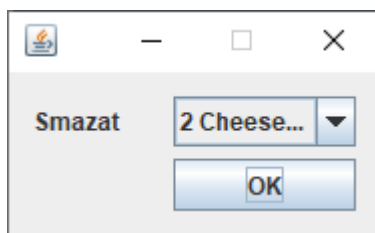


Obrázek 9 - úprava přídavku

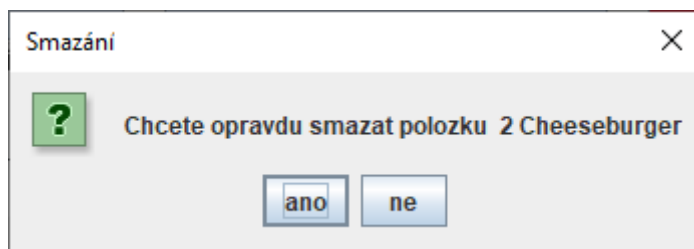
Přidávání nových položek zajišťuje třída `PolozkaFrame` a `PridavekFrame`. Vstupním parametrem konstruktoru třídy je `id` položky/přídavku, pokud je `id` nenulové, jedná se o editaci, jinak se jedná o vkládání nové položky. Výběr položky k editaci je zajištěn pomocí třídy `EditFrame`. Okno zadávání položky se skládá ze dvou textových polí a tlačítek, okno s přídavky má navíc `JComboBox` s výběrem příslušící položky. Vložení správných hodnot je ošetřeno pomocí regulárních výrazů. Po potvrzení je pomocí metody `writePolozka()/writePridavek()` vytvořena nová položka/přídavek, popřípadě pomocí `updatePolozka()/updatePridavek()` upravena.

### 3.5.3. Mazání položek/přídavků

Mazání zajišťuje třída `DeleteFrame`. Podle vstupního parametru konstruktoru je určeno zda se jedná o mazání položky či přídavku. Položky/přídavky se pak pomocí metod `getPolozky()`/`getPridavkyAll()` vypíší do `JComboBox`.



Obrázek 10 - mazání položky



Obrázek 11 – potvrzení mazání

Po kliknutí na ok je uživatel dotázán pomocí `JOptionPane`, zda chce položku opravdu smazat, poté proběhne mazání pomocí metod `deletePolozka()`/`deletePridavek()`.