

Gymnázium, Praha 6, Arabská 14

Arabská 14 Praha 6, 160 00



Maloobchodní kasa

Ročníková práce

Autor: David Mikolášek

Třída: 2.E

Předmět: Programování

Školní rok: 2022/23

Vyučující: Mgr. Jan Lána

Třídní učitel: Mgr. Blanka Hniličková

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

.....
David Mikolášek

Anotace

Cílem projektu bylo vytvořit program, který bude napomáhat chodu stánkového či maloobchodního prodeje a ulehčovat práci prodejcům. Hlavní úkol programu je sčítat celkovou cenu nákupu a vypisovat historii nákupů. Program je napsaný pomocí programovacího jazyka Java za použití grafického rozhraní JavaFX.

Annotation (English)

The goal of this project was to create a program, that will help the process of stall sales or retail sales and make the work of sellers easier. The main task of the program is to add up the total of the purchase and list the history of purchases. The program is written in Java using the graphic interface JavaFX.

Zadání ročníkové práce

Aplikace pro prodejce při stánkovém prodeji, kde si prodejce může upravit seznam produktů a jejich cenu, sečíst cenu nákupu, případně kolik má zákazníkovi vrátit, aplikace vytiskne účtenku, nebo udělá výčet prodaných produktů za den

Podrobnější popis projektu

Aplikace je určena pro maloobchod nebo stánkový prodej, tedy pro situace, kdy nemá prodejce přístup ke kase, musí si ceny produktů pamatovat, nebo sčítat na kalkulačce. V aplikaci si uživatel vytvoří seznam produktů a každému zadá svou cenu. Aplikace přiřadí každému produktu své tlačítko, které produkt a jeho cenu přidá do aktuálního nákupu. Aplikace průběžně sčítá celkovou cenu nákupu a po stisknutí tlačítka hotovo (*done*) přidá hotový nákup do přehledu (*overview*) pro lehčí dohledávání historie prodeje, nebo výdajů.

Obsah

Úvod	6
1 Uživatelské rozhraní	7
1.1 Hlavní menu	7
1.2 Funkční obrazovka	8
1.2.1 Ovládací tlačítka	8
1.3 Okno přidání nového produktu	9
1.4 Obrazovka přehledu	9
2 Fungování programu	10
2.1 Třída product	10
2.2 Soubor productlist.dat	10
2.3 Soubor donelist.txt	10
2.4 Třída HelloController.java	11
2.4.1 Metoda onStart	11
2.4.2 Metoda onAdd	12
2.5 Třída PopupController	12
2.5.1 Metoda onCheck	12
Závěr	13

Úvod

Cílem projektu bylo ulehčit prodejci ve stánkovém nebo maloobchodním prodeji sčítání ceny nákupů a zvýšit celkovou efektivitu a rychlost prodeje.

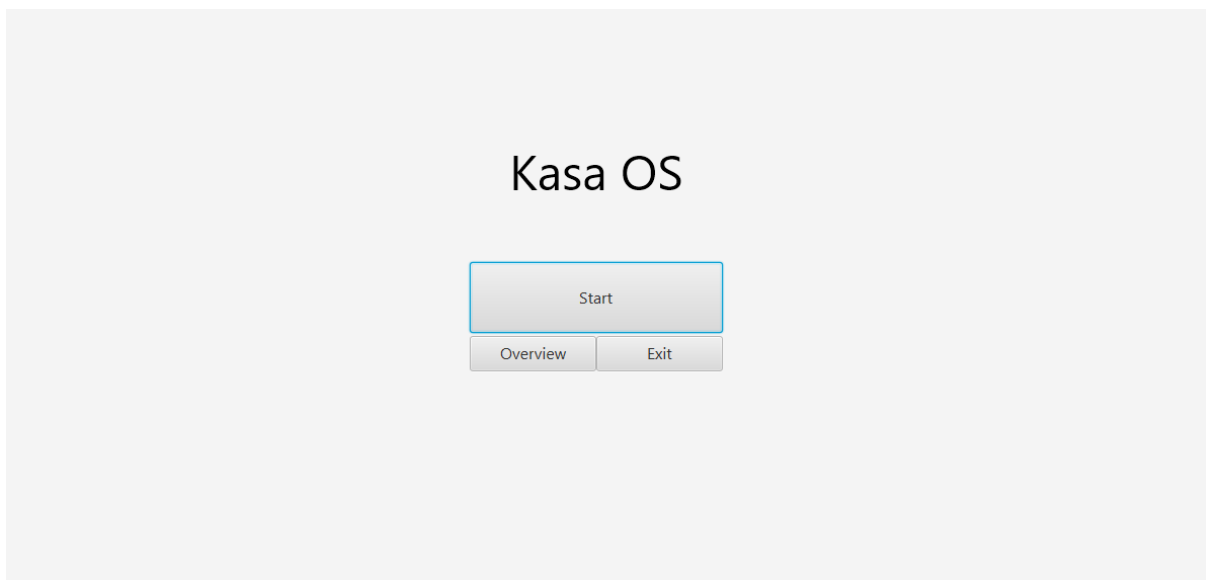
Tohle téma jsem si vybral, protože jsem se sám v takové situaci nacházel a napadlo mě, že by se tímhle způsobem mohl jednoduše zrychlit provoz a zvýšit efektivita a funkčnost prodeje.

Na programování jsem použil IntelliJ IDEA community edition IDE, protože ho používám na hodinách programování a jsem na něj zvyklý.

1 Uživatelské rozhraní

1.1 Hlavní menu

Při spuštění aplikace se zobrazí hlavní menu, které obsahuje název programu a tlačítka *Start*, *Quit* a *Overview*.

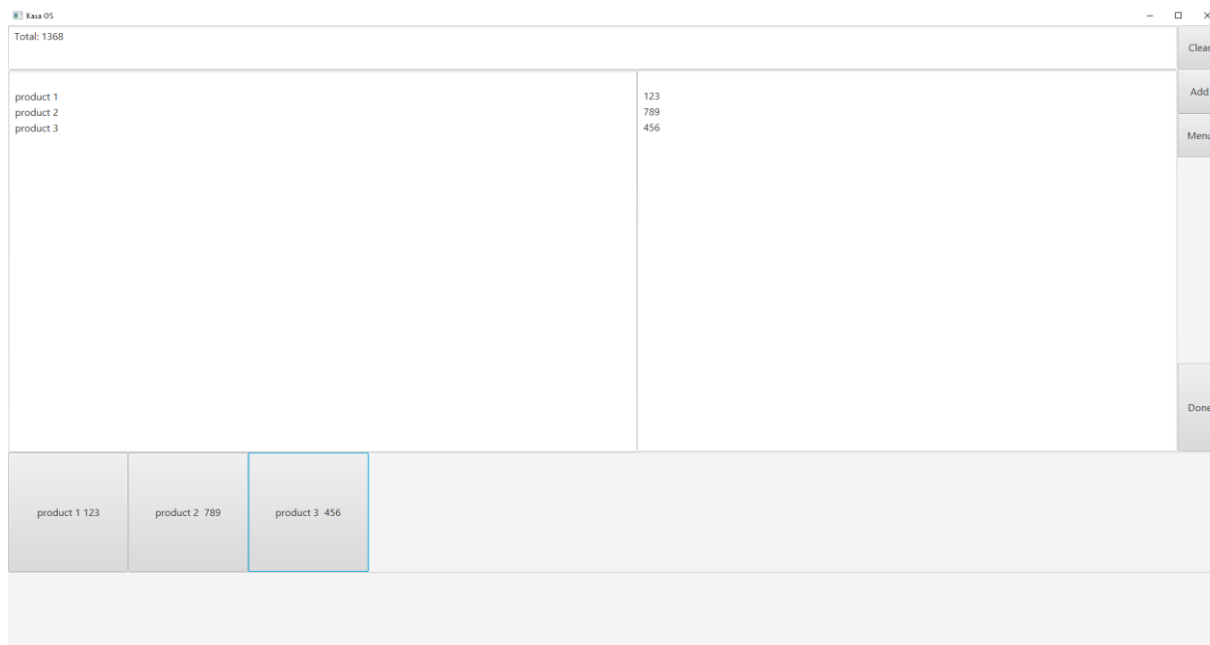


Obrázek 1 – Hlavní menu

Tlačítko *Start* přesune uživatele do funkční obrazovky programu, tlačítko *Exit* ukončí program a tlačítko *Overview* přesune uživatele do obrazovky shrnutí hotových nákupů.

1.2 Funkční obrazovka

Funkční obrazovka se skládá ze tří částí – *Scrollpane*, ve kterém jsou tlačítka se jmény a cenami produktů, textová pole *textfield* 1,2 a 3, která vypisují současný nákup a celkovou cenu nákupu a ovládací tlačítka *Clear*, *Add*, *Menu* a *Done*.



Obrázek 2 – Funkční obrazovka

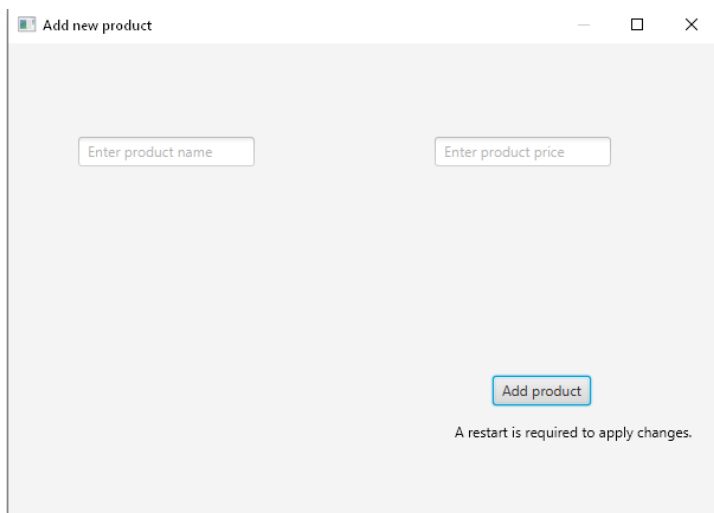
1.2.1 Ovládací tlačítka

Tato tlačítka se nachází na pravé straně programu a slouží k manipulaci se zvoleným nákupem, nebo navigaci mezi obrazovkami. Tlačítko *Clear* vymaže aktuální nákup, tlačítko *Add* otevře okno pro přidání nového produktu, Tlačítko *Menu* přesune uživatele do hlavního menu a *Done* potvrdí nákup, vymaže textová pole a přidá nákup do souboru *donelist.txt* s datem potvrzení nákupu.

1.3 Okno přidání nového produktu

Při stisknutí tlačítka *Add* se otevře okno *Add new product* s textovým polem *name* pro jméno produktu a *price* pro jeho cenu. Ve spodní části okna je tlačítko *Add product*, které zkontroluje, zda je správně zadané jméno a cena produktu.

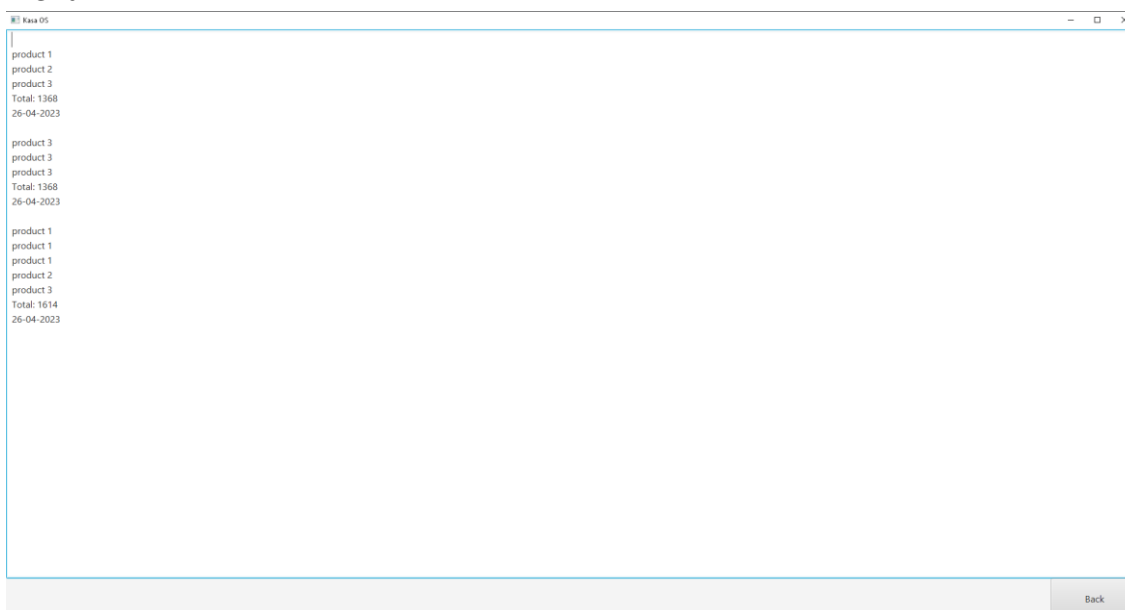
Pokud je všechno v pořádku, přidá produkt do souboru *productlist.dat* a zavře okno. Pokud není zadané jméno, cena, nebo jsou v políčku *price* písmena, vyzve uživatele, aby zadal platné jméno a cenu, nebo vymaže chybný obsah políčka *price*.



Obrázek 3 – Okno přidání nového produktu

1.4 Obrazovka přehledu

Tlačítko *Overview* v hlavním menu přesune uživatele do obrazovky přehledu. Tato obrazovka obsahuje neupravitelné textové pole *overviewarea*, ve kterém jsou vypsány potvrzené nákupy s daty jejich uskutečnění a tlačítko *Back*, které vrátí uživatele do hlavního menu.



Obrázek 4 – Obrazovka přehledu

2 Fungování programu

2.1 Třída product

Třída *product* je jednoduchá třída obsahující jméno a cenu produktu. Při přidání nového produktu pomocí tlačítka *Add* vytvoří nový produkt a přidá ho do souboru *productlist.dat*.

2.2 Soubor productlist.dat

V tomto souboru je uložen *Arraylist* se všemi přidávanými produkty. Z tohoto souboru se při spuštění funkční obrazovky vezme *Arraylist* s produkty pomocí třídy *ObjectInputStream* a vytvoří pro každý produkt jeho příslušné tlačítko a přidá ho do útvaru *ScrollPane* ve spodní části funkční obrazovky.

Při tvoření tlačítek se jim nastaví funkce pomocí *Lambda expression* – vypsání jména produktu do textového pole *textfield 1*, vypsání ceny produktu do *textfield 2* a přičtení ceny produktu do celkové ceny nákupu v *textfield 3*.

2.3 Soubor donelist.txt

Při stisknutí tlačítka *Done*, pokud nejsou textová pole prázdná, se jejich obsah vypíše do souboru *donelist.txt*. Pod ním se vypíše datum pomocí řetězce *date* s formátem *SimpleDateFormat* dd-MM-yyyy. Ze souboru *donelist.txt* se vypíše text do textového pole *overviewarea* pomocí třídy *ObjectInputStream* při stisknutí tlačítka *Overview* v hlavním menu.

2.4 Třída HelloController.java

Tato třída dává funkce všem tlačítkům v hlavním okně programu.

2.4.1 Metoda onStart

Metoda *onStart* se spustí po stisknutí tlačítka *Start* v hlavním menu. Přesune uživatele na funkční obrazovku programu a vytvoří stejně velká tlačítka pro všechny přidané produkty. Tlačítka jsou umístěna do útvaru *Gridpane*, který je vložený do útvaru *Scrollpane*.

Gridpane je použit, aby tlačítka zachovala strukturu mřížky s deseti sloupci. Útvar *Scrollpane* zajistí, aby se dalo kolečkem myši posouvat v mřížce s tlačítky. Tlačítka zaujmou správnou polohu v mřížce díky jednoduchému *for cyklu*, který zkontroluje kolikáté je tlačítko v seznamu a přidá ho do správného sloupce a řádku. Pokud je počet tlačítek větší než 10, 20 atd., *for cyklus* přidá mřížce řádky tak, aby se do ní vešla všechna tlačítka.

```
int radky = 0;
if (vbox.getChildren().isEmpty()) {FileInputStream in = new FileInputStream(fn);
    ObjectInputStream inn = new ObjectInputStream(in);
    ArrayList<product> temp = (ArrayList<com.example.rpkasa2.product>) inn.readObject();
    for (int i = 0; i < temp.size(); i++) {
        int sloupce = i;
        if (i > 9) {
            radky = (i/10);
            sloupce = (i%10);
        }
        Button b = new Button();
        b.setStyle("-fx-font-size: 16px;");
        product pp = temp.get(i);
        b.setText(pp.nazev + " " + pp.cena + "\n");
        b.setMinSize(190, 190);
        b.setOnAction((event) -> {
            String curtext = textarea.getText();
            textarea.setText(curtext + "\n" + pp.nazev);
            String curnumbers = textarea2.getText();
            total[0] += pp.cena;
            textarea3.setText("Total: " + total[0]);
            textarea2.setText(curnumbers + "\n" + pp.cena);
        });
        if (i % 9 == 0) {
            gridpane.addRow(radky);
        }
        gridpane.add(b, sloupce, radky);
    }
}
```

Obrázek 5 – Metoda onStart

2.4.2 Metoda onAdd

Tato metoda otevře nové okno s vlastním *fxml* souborem a vlastní třídou *PopupController* udělující funkci tlačítkům v okně.

2.5 Třída PopupController

2.5.1 Metoda onCheck

Metoda *onCheck* ve třídě *PopupController* při stisknutí tlačítka *Add* zkontroluje, zda jsou informace v textových polích *name* a *price* zadány zprávně. Pokud uživatel nechá jedno z polí prázdné, zobrazí se text *error*, který vyzve uživatele, aby správně vyplnil obě políčka. V případě, že uživatel chybně zadá cenu produktu, zobrazí se text *error* a obsah v poli *price* se smaže.

Když je jméno i cena zadaná uživatelem v pořádku, vytvoří se nová proměnná typu *product* se jménem a cenou produktu a přidá se do souboru *productlist.dat*. Pokud je soubor *productlist.dat* prázdný, vytvoří se nový *ArrayList* s novým produktem, který je následně přidán do souboru pomocí třídy *ObjectOutputStream*. Pro přidání nově zadaných produktů je vyžadováno restartování aplikace.

```
String name = namefield.getText();
if (name.isEmpty()) {
    error.setVisible(true);
}
if (price == 0) {
    error.setVisible(true);
}
if (!name.isEmpty() && price != 0) {
    System.out.println(name + " " + price);
    product pproduct = new product(name, price);
    File fn = new File("productlist.dat");
    if (fn.length() == 0) {
        ArrayList<product> list = new ArrayList<>();
        list.add(pproduct);
        FileOutputStream out = new FileOutputStream(fn);
        ObjectOutputStream oo = new ObjectOutputStream(out);
        oo.writeObject(list);
    } else {
        FileInputStream in = new FileInputStream(fn);
        ObjectInputStream inn = new ObjectInputStream(in);
        ArrayList<product> temp = (ArrayList<com.example.rpkasa2.product>) inn.readObject();
        temp.add(pproduct);
        in.close();
        FileOutputStream out = new FileOutputStream(fn);
        ObjectOutputStream oo = new ObjectOutputStream(out);
        oo.writeObject(temp);
        out.close();
    }
    Stage stage = (Stage) checkbtn.getScene().getWindow();
    stage.close();
}
```

Obrázek 6 – Metoda *onCheck*

Závěr

Myslím si, že se mi podařilo úspěšně splnit zadání, program funguje tak, jak jsem si představoval. Přišel jsem na to, jak otevřít nové okno v programu a na to, jak funguje scrollpane.

Pořád jde program vylepšit v některých ohledech, třeba mít u každého produktu v aktuálním nákupu tlačítko na odstranění, aby nemusel uživatel smazat celý nákup. Ale celkově bych považoval projekt za úspěšný.