

Gymnázium, Praha 6, Arabská 14
předmět programování, vyučující Mgr. Jan Lána



Nákupní lístek

Ročníkový projekt

Michal Bogdanov, 1.E

duben 2022

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, Arabská¹⁴ 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 2022

podpis:

Michal Bogdanov

Anotace:

Cílem projektu je umožnit uživatelům efektivní nakupování s ohledem na jejich potřeby. Tento projekt umožňuje vést nákupní seznam elektronicky. Program je napsán s pomocí grafického rozhraní JavaFX, které je součástí programovacího jazyka Java SE 17.

Abstract:

The aim of the project is to enable users to shop efficiently with their needs in mind. This project allows to keep a shopping list electronically. The program is written using the JavaFX graphical interface, which is part of the Java SE 17 programming language.

Zadání ročníkového projektu:

Nákupní Lístek

Nákupní lístek je program, kde si uživatel napíše nákupní seznam, ve kterém následně může vyškrtávat položky, které už má. Program mu následně vypíše položky, které jsou uvedeny v seznamu a uživateli chybí. Seznam si bude moct uživatel uložit do souboru, a poté ho znovu využít. Data v seznamu se budou ukládat do databáze (pojem databáze v tomto ročníkovém projektu je chápán jako ukládání do pole).

Obsah

1	Úvod	5
2	Vývojové prostředí	6
2.1	Java	6
2.2	IntelliJ IDEA	6
3	Kód	7
3.1	O Kódu	7
3.2	Problémy	7
4	Aplikace	9
4.1	Příručka	9
4.2	Vzhled	9
5	Závěr	11
	Seznam obrázků	12
	Seznam ukázek kódu	12
	Seznam zdrojů	13

Úvod

Na úvod bych chtěl uvést, proč jsem si vybral téma "Nákupní lístek" a následně popsat, co naleznete v dalších kapitolách.

Téma jsem zvolil kvůli tomu, že už se mi několikrát stalo, že jsem zapomněl, co mám nakoupit. Bylo to především proto, že se mi několikrát podařilo ztratit nákupní seznam a skutečnost, že mobil jen tak neztratím, mě navedla k tomu, udělat tento program. Další využití tohoto programu by mělo být, aby šel používat i k jiným účelům, než je nakupování např. jako seznam věcí, které plánuji udělat. Mojí představou bylo také, aby bylo přehledně znázorněno, co jsem splnil a co je potřeba splnit.

V následujících kapitolách naleznete, v jakém vývojovém prostředí a jazyce jsem program psal, následně úryvky kódu a obrázky aplikace. Popisuji zde, na jaké problémy jsem narazil při psaní kódu a jak jsem je poté vyřešil, také se zmíním o možných problémech této aplikace.

Vývojové prostředí

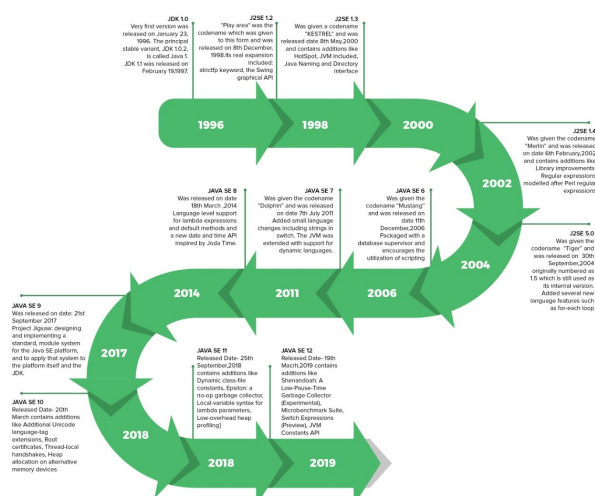
2.1 Java

"Java je programovací jazyk vyvinut firmou Sun Microsystems"[10], který byl představen v roce 1995. Následně se stala vlastníkem Javy SE firma Oracle. Java SE je platforma, kterou využívají aplikace pro počítače. [10]. V následujících letech se Java rozvíjí a přidávají se klíčová slova jako je private a protected, to znamená omezený přístup proměnných v jedné metodě k dalším metodám, poté vzniká Java pro práci s grafickým rozhraním jako je knihovna AWT, Swing a JavaFX.

JavaFX je stále rozvíjeným grafickým prostředím v programovacím jazyku Java, samotnou knihovnou se stává od Java 11 a nahrazuje Swing[11].

2.2 IntelliJ IDEA

"IntelliJ IDEA je vývojové prostředí pro programování v jazycích Java, XML a dalších." [9] Společnost JetBrains, která vytvořila toto vývojové prostředí, dostala za IntelliJ ocenění za nejlepší vývojové prostředí pro Javu [9]. IntelliJ IDEA Community Edition lze stáhnout zdarma nebo jako placenou verzi tedy Ultimate Edition. IntelliJ IDEA stáhnete na <https://www.jetbrains.com/idea/>.



Obrázek 2.1: Časová osa Javy do roku 2019 [2]

Kód

Tato kapitola vysvětluje strukturu programu a následně popisuje problémy v kódu. K vysvětlování struktury programu doporučuji si můj program otevřít. Problematiku sestavování kódu vysvětluji tak, jak jsem jí vyřešil já. Jinak na to mohou existovat jednodušší či více vylepšená řešení.

3.1 O Kódu

Před tím než začnu uvádět problémy, které jsem v rámci projektu řešil, tak se pokusím ve zkratce popsat svůj kód. Na začátku kódu si vytvářím proměnné označované jako „*member*” neboli „*člen*”. Tento člen třídy je určen k tomu, aby byl vidět i ve více metodách, které jsou ve třídě. Ale je potřeba určit, zda bude `public`, `private` nebo `protected`, jelikož když by byla proměnná pojmenována stejně a byl by použit například jiný datový typ, tak by to nemuselo fungovat podle představ, také je to samozřejmě i pro lepší čitelnost kódu.

```
public FileChooser fileDialog = new FileChooser();
```

Listing 3.1: příklad člena třídy

Dále v kódu vytvářím 9 metod/funkcí: 2x `pridat`, `odebrat`, `potvrdit`, `odskrtnout`, `ulozit`, `nacist`, `ovladaciPrvky`, `start`. Metoda `pridat` je dvakrát, protože jedou ji použijeme pokud bychom data načítali ze souboru. Takto se nám vytvoří stejný styl a stav, ve kterém jsme ho uložili. A podruhé, pokud bychom si seznam vytvářeli úplně nový. Metoda `potvrdit` slouží k tomu abychom jednotlivé položky (které potřebujeme koupit) načetli do databáze. Tato metoda se spustí za pomoci tlačítka „*potvrdit*”, po spuštění uloží položku do databáze a bude vidět v panelu(záložce) chybí. Funkce `ulozit` a `nacist` ukládá do souboru nebo načítá ze souboru typu CSV s tím, že v metodě `ulozit` se přepisuje středník na čárku. Tato funkce je zde s ohledem na to, že středník odděluje v CSV souborech sloupce. Metoda `ovladaciPrvky` uděluje specifikované akce tlačítkům a připravuje parametry, které poté budou vidět v aplikaci. Tato metoda se, poté zavolá ve funkci `start`, kde se vytvoří scéna.

3.2 Problémy

Jeden z prvních problémů tohoto programu byl, že po spuštění aplikace, byl vidět pouze prázdný panel ve všech záložkách. Na odstranění problému jsem použil metodu `.setContent()` [1], která vkládá obsah „něčeho” do záložek. Následně, co byl problém vyřešen, přišel další, a to, že záložky bylo možné zavřít. Toto bylo nežádoucí, proto jsem problém vyřešil takto:

```
panelNeeds.setOnCloseRequest(e -> e.consume());
```

Listing 3.2: zobrazení řešení problému zavírání záložek [8]

Jelikož nebylo přesně definováno, jak uložit text a stav do databáze, zároveň kód programu byl tak trochu *”toaletní papír”*, i přestože program fungoval, bylo nutné ho změnit. Proto jsem ho přepsal a dal jsem vše do polí (*”Pole je datová struktura, kterou si můžeme představit jako sadu proměnných stejného typu poskládaných v řadě za sebou.”*[6]) a použil výše specifikované metody, což zkrátilo kód. Zároveň vytvoření polí vyřešilo problém ukládání dat do databáze. Skutečnost, že se jednalo o pole statické, tedy s omezenou velikostí, byl další problém. Uživatel nebude chtít být omezen v rozsahu zadávaných údajů v kódu. A jelikož tento problém způsobil další problém a to mezi židli a klávesnicí, bylo nezbytné celý kód přepsat. Tentokrát jsem si nejprve vytvořil vedlejší program, který byl podobný mému ročníkovému projektu. Nyní jsem na místo polí statických, vytvořil dvě pole dynamické, tedy se stále se rozšiřující velikostí. Tyto pole slouží k uložení názvu produktu a jeho stavu, o tom zda je splněn nebo chybí. Takto byl vyřešen zásadní problém. Následně chybělo uložit a načíst údaje seznamu ze souborů, k tomu je potřeba *FileChooser*. Pro metodu uložit jsem použil

```
File newFile = fileDialog.showSaveDialog(null);
if (newFile != null) {
    try {
        OutputStreamWriter writer = new OutputStreamWriter(
            new FileOutputStream(newFile));

        BufferedWriter bufWriter = new BufferedWriter(writer);

        for (int index = 0; index < data.size(); index++) {
            String name = data.get(index);
            name = name.replace(delimiter, ',');
            bufWriter.write(name);
            bufWriter.write(delimiter);
            bufWriter.write(String.valueOf(done.get(index)));
            bufWriter.newLine();
        }
        bufWriter.close();
    }
    catch (IOException e) {
        e.printStackTrace();
    }
}
```

Listing 3.3: ukázka funkce uložit [4]

s tím že se údaje uloží do CSV souboru s využitím kódování UTF-8. Pro metodu nacíst je to podobné, ale je kód pozměněn např. *.showOpenDialog()* a *.BufferedReader()* a další[3]. Jelikož se jedná o zobrazení pro čtení CSV souboru, vede to k dalším komplikacím v případě použití znaku *”středník”*. Soubory CSV rozdělují sloupce právě středníkem, přičemž aplikace potřebuje sloupce pro rozdělení stavu a zápisu seznamu. Jestliže by uživatel napsal do seznamu středník, tak by zobrazení seznamu, při znovu vložení neodpovídalo požadované struktuře. Pro odstranění problému jsem použil metodu *.replace()*, která středník nahradí čárkou. Tedy pokud by uživatel napsal středník do seznamu, uloží se tento symbol jako čárka, a pokud se bude seznam znovu vkládat, tak se vypíše místo středníku čárka.

Poslední komplikace, která nastala, byla s ikonou aplikace, která je přímo s programem, a která nešla zobrazit. Chyba nebyla úplně v kódu, ale ve špatném uložení ikony. Ikona, se totiž musí uložit do *”com.example.název projektu”*, což je pod sou-

borem *"resources"*, pak stačí zadat:

```
java.net.URL imgURL = getClass().getResource(  
    "/com/example/nazev_projektu/nazev_obrazku");
```

Listing 3.4: vkládní obrázku[7]

java.net.URL čte lokaci pro *resources* a java.net poskytuje třídy pro implementaci síťových aplikací[5].

Aplikace

V této kapitole popisuji, jak používat aplikaci Nákupní seznam a popisuji vzhled aplikace. Příručku jsem popsal tak, aby uživateli bylo zřejmé, jaké položky má vyplňovat, jakým způsobem má ukládat Nákupní seznam a jaké klávesové zkratky může využít. V podkapitole vzhled vysvětluji print screen aplikace.

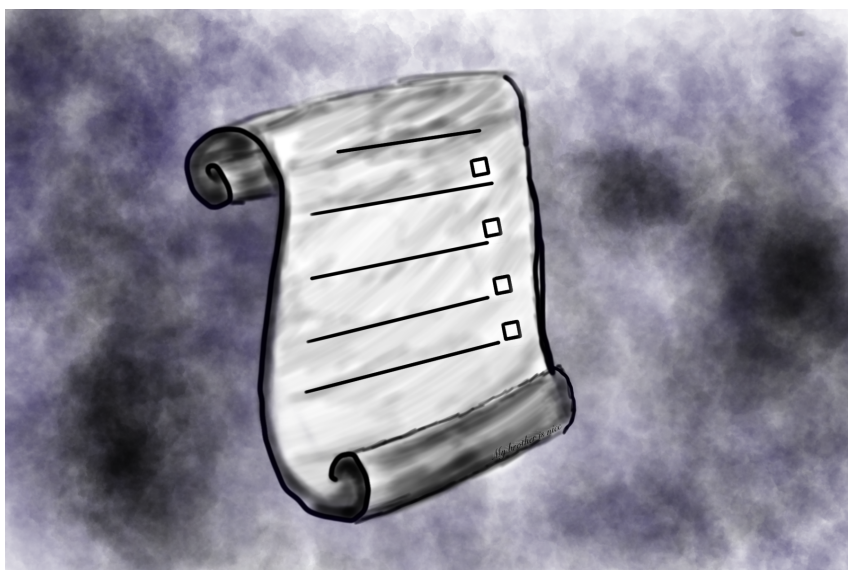
4.1 Příručka

Po spuštění aplikace je možné přidávat, pomocí tlačítka *přidat* nebo klávesové zkratky *Control(Ctrl) + Alt + P*, či odebírat za pomoci tlačítka *odebrat* nebo klávesy *Delete(Del)*, textové pole, do kterého lze zapsat položky do seznamu. Poté co je seznam napsaný, tak klikněte na všechny tlačítka *potvrdit* nebo zmáčkněte klávesu *Enter* v textovém poli, aby se data uložila do databáze. Doporučuji do textového pole nepsat středník, jelikož po uložení a znovu načtení se místo středníku napíše čárka. Potom co jste vše potvrdili můžete si to uložit do souboru pomocí tlačítka *uložit* nebo pomocí kláves *Control + Shift + S*.

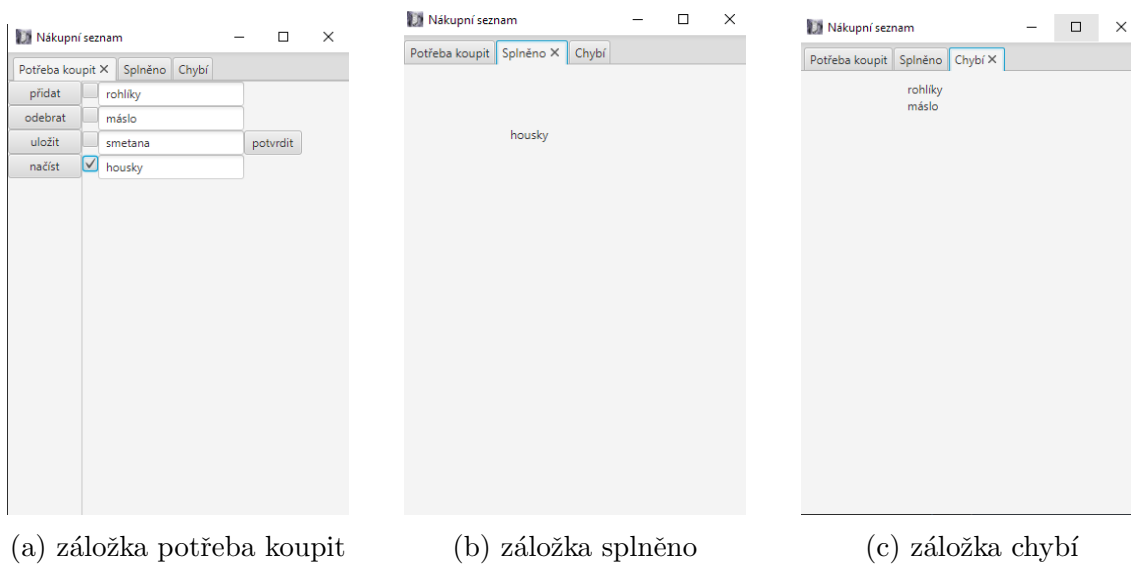
Pozor, soubor se uloží ve formátu CSV, ale pokud budete chtít přepsat stav z "chybí" na "splněno" nebo naopak, tak se soubor otevře v excelu a teprve nyní můžete ve sloupci přepsat údaj, kde se nachází true(splněno) nebo false(chybí). Pokud máte uloženo a potvrzeno můžete nakupovat a u toho odškrtnávat položky, které máte splněné. Položky, které máte odškrtnuty se zobrazí v záložce *splněno* a ty které, odškrtnuty nejsou budou zobrazeny v záložce *chybí*. V aplikaci lze stav změnit pouze z "chybí" na "splněno". Stav položek a položky v seznamu lze průběžně měnit a ukládat. Pro načtení souboru je tlačítko *načíst* nebo klávesová zkratka *Control + V*.

4.2 Vzhled

Na prvním obrázku vidíte ikonu nebo spíše logo programu, které má znázorňovat nákupní seznam. Na níže uvedených třech obrázcích je možné vidět stav aplikace, kterou může vidět uživatel. Jedná se o záložky s různými stavy například u položky housky je vidět, že je odškrtnutá tedy není vidět v záložce "chybí" ale v záložce "splněno". Nebo je možné si všimnout, že smetana nebyla potvrzena a to znamená, že není vidět ani v záložce "splněno" tak ani v záložce "chybí". Položky mléko a rohlíky byly potvrzeny, proto tlačítko "potvrdit" se znovu nezobrazuje. Protože tyto položky nebyly v záložce "potřeba koupit" odškrtnuty, jsou zobrazeny pouze v záložce "chybí".



Obrázek 4.1: Ikona aplikace



(a) záložka potřeba koupit

(b) záložka splněno

(c) záložka chybí

Obrázek 4.2: obraz uživatele

Závěr

Tento ročníkový projekt mě naučil pracovat, jak s abstraktní částí programování, tak i s grafickou částí. Také jsem se naučil pracovat s metodami a vytvářet lepší proměnné. Pomůckou mi bylo, že jsem si nakreslil schéma programu, podle kterého jsem se snažil postupovat. Aplikace slouží pro splnění základního požadavku, nicméně by, dle mého názoru, bylo uživatelsky komfortnější, aby např. text se ukládal do pole už bez tlačítka *potvrdit* nebo vzhled tlačítek a pozadí v aplikaci měly odlišnou barvu, nebo např. byl vycentrován "Label" v záložkách "chybí" a "splněno". Domnívám se, že tato aplikace by mohla usnadnit uživateli sledovat a upravovat jednotlivé položky v nákupním seznamu.

Seznam obrázků

2.1	Časová osa Javy do roku 2019 [2]	6
4.1	Ikona aplikace	10
4.2	obraz uživatele	10

Seznam ukázek kódu

3.1	příklad člena třídy	7
3.2	zobrazení řešení problému zavírání záložek [8]	7
3.3	ukázka funkce uložit [4]	8
3.4	vkládní obrázku[7]	9

Seznam zdrojů

- [1] *EDUCBA*. Dub. 2021. URL: <https://www.educba.com/javafx-tabpane/>.
- [2] *GeeksforGeeks*. Dub. 2022. URL: <https://www.geeksforgeeks.org/the-complete-history-of-java-programming-language/>.
- [3] *How to Read and Write Text File in Java*. Čvc. 2019. URL: <https://www.codejava.net/java-se/file-io/how-to-read-and-write-text-file-in-java>.
- [4] *How to save files using a File Chooser in JavaFX?* neznámý neznámý. URL: <https://www.tutorialspoint.com/how-to-save-files-using-a-file-chooser-in-javafx>.
- [5] *java.net (Java Platform SE 7)*. Čvn. 2020. URL: <https://docs.oracle.com/javase/7/docs/api/java/net/package-summary.html>.
- [6] Jan Lána. *Testovadlo*. Zář. 2021. URL: <https://gyarab.ddns.net/programovani-pg1/09-stabni-kultura-pole-239/>.
- [7] *ow to Use Icons (The Java™ Tutorials ě Creating a GUI With Swing ě Using Swing Components)*. neznámý neznámý. URL: <https://docs.oracle.com/javase/tutorial/uiswing/components/icon.html>.
- [8] *Stack Overflow*. Dub. 2016. URL: <https://stackoverflow.com/questions/31531059/how-to-remove-close-button-from-tabs-in-javafx>.
- [9] *Wikipedia*. Pros. 2021. URL: https://cs.wikipedia.org/wiki/IntelliJ_IDEA.
- [10] *Wikipedia*. Led. 2022. URL: [https://cs.wikipedia.org/wiki/Java_\(programovac%C3%AD_jazyk\)](https://cs.wikipedia.org/wiki/Java_(programovac%C3%AD_jazyk)).
- [11] *Wikipedia*. Dub. 2022. URL: <https://cs.wikipedia.org/wiki/JavaFX>.