

Semestrální práce z předmětu

KIV/TI

Pavel Shevnin A21B0269P

20.06.2023

1. Zadání

1.1 Z21 Automatový model prodejního automatu

Automat bude prodávat 5 druhů zboží za různé ceny. Bude přijímat mince do hodnoty 20 Kč

Rámcová představa o hardware, který bude modelován:

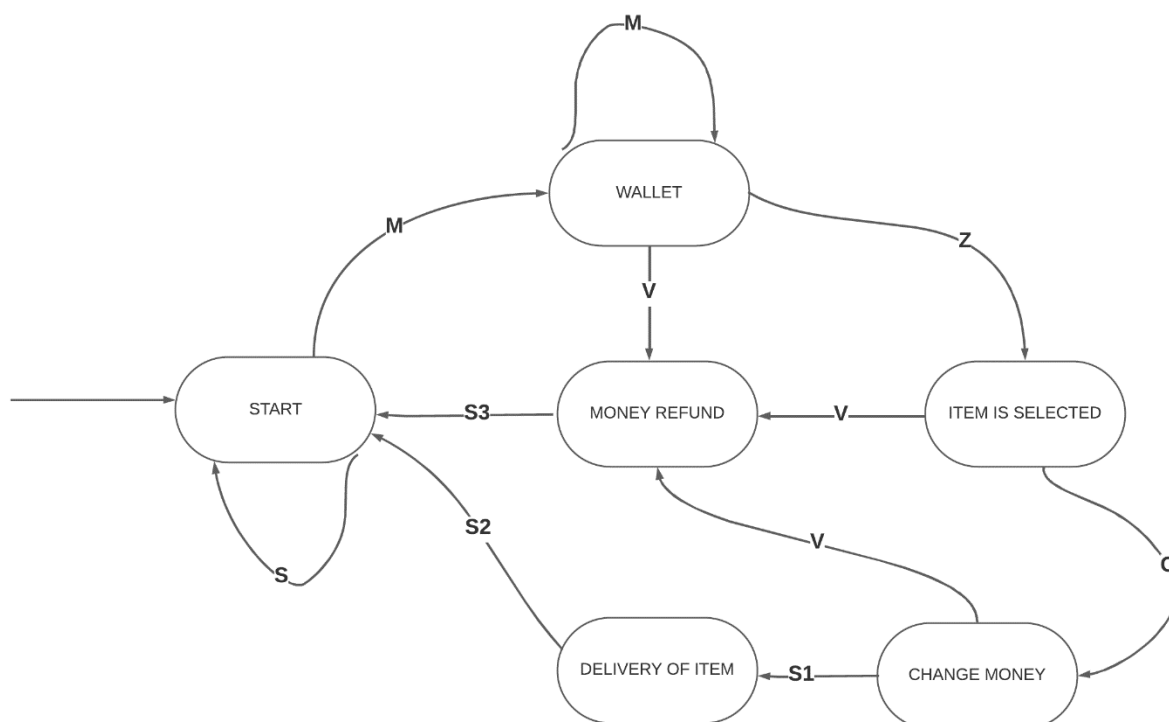
- vstupní zásobník mince (do něho padají mince a jsou v něm až do ukončení transakce)
- hlavní zásobník na mince (do něho po ukončení transakce bez třídění spadnou vhozené mince)
- výstupní zásobníky na mince (pro každou hodnotu mince samostatný zásobník, z nějž jsou vyplaceny vrácené drobné).
- STORNO tlačítko umožní přerušit rozpracovanou transakci (peníze vhozené do vstupního zásobníku vypadnou zpět zákazníkovi) výdejní zásobníky na jednotlivé druhy zboží

Definujte, co vše bude zahrnuto pod pojmem stav

Definujte počáteční stav (=stav po naplnění obsluhou).

Na základě tohoto modelu naprogramujte v libovolném programovacím jazyku program, který bude simulovat činnost takového prodejního automatu včetně jeho plnění obsluhou.

2. Návrh konečného automatu



Obr. 1 – Návrh konečného automatu

Legenda:

M – Vkládání mince do prodejního automatu.

S – Servisní tlačítko pro obdluhu prodejního automatu

V – Tlačítko STORNO, pro zrušení transakce a vracení peněz

Z – Vyber zboží

O – OK (potvrzení výběru zboží a prodloužení objednávky)

S1 – Peníze jsou úspěšně vracené

S2 – Zboží je úspěšně vydano

S3 – Peníze ze vstupního zásobníku jsou vracene a objednávka je zrušena

2.1 Slovní popis struktury konečného automatu

V počátečním stavu "START" uživatel může vložit mince nebo stisknout servisní tlačítko. Pokud vloží první minci, přejde do stavu "WALLET", ze kterého může pomocí tlačítka "V" (STORNO) vrátit peníze ze vstupního zásobníku. Uživatel může vkládat další mince nebo vybrat zboží tlačítkem "Z" a přejít do stavu "ITEM IS SELECTED". Z tohoto stavu také může vrátit peníze pomocí "V". Automat kontroluje, zda dané zboží je dostupné. Pokud ano, uživatel může potvrdit objednávku a přejít do stavu "CHANGE MONEY". Pokud automat nemá co vrátit v zásobníku pro vrácení, vrátí peníze ze vstupního zásobníku a přejde do stavu "START", pokud uživatel vložil dostatečnou částku a automat má co vrátit, budou vydány peníze a automat, pomocí "S1" přejde do stavu "DELIVERY OF ITEM", ve kterém v případě úspěšně vydaného zboží přejde zpět do stavu "START".

3. Implementace

Pro semestrální práci jsem si vybral programovací jazyk Java z toho důvodu, že je objektově orientovaným jazykem. Což bude během této semestrální práce velkou výhodou. Při realizaci programu jsem se snažil dodržovat architekturu MVC, která je sestavena z modulů Model, View a Controller. Model slouží pro komunikaci s daty, Controller slouží pro řízení celého programu a View slouží pro práci s obrazovkou (v našem případě pro vypis do konzoli).

3.1 Main

Tato třída je třídou která spouští celý program, a v nekonečném cyklu while čeká na uživatelské vstupy. Před spuštěním programu v této třídě se generují atributy tříd Model a Automat. Model je součástí bloku model, automat je součástí bloku controller z MVC.

3.2 Produkt

Tato třída reprezentuje zboží ktere bude automat prodavat. obsahuje id, name, count a price.

3.3 Model

Tato třída slouží pro náš program databáze. Zde se nacházejí všechny datové struktury, které budeme využívat. Máme zde tři uložště pro mince (vstupní zásobník, hlavní zásobník a výstupní zásobník), které jsou reprezentovány datovým typem `HashMap<Integer, Integer>`, kde klíč je hodnota mince a hodnota je počet mincí uvnitř uložště. V této třídě je ještě jeden `HashMap`, který je typu `<Integer, Product>`, kde klíč je index zboží a hodnota je atribut třídy `Product`, který obsahuje informace o zboží. V této třídě jsou metody pro komunikaci a manipulaci s daty.

3.4 Automat

Tato třída je hlavní řídicí třídou celé aplikace. Obsahuje atribut třídy `Model` pro komunikaci s daty a atribut třídy `WindowView` pro výpis zpráv do konzole. V proměnné `"actualState"` uchovává informaci o aktuálním stavu konečného automatu, který řídí celý program. Po každém uživatelském příkazu v metodě `"main"` volá metoda `"processInput(String input)"`, která přijímá uživatelský příkaz a na základě aktuálního stavu programu volá odpovídající metody. Například, pokud je program ve stavu `"WALLET"`, volá metodu `"walletStateController(String input)"` pro zpracování uživatelského příkazu v tomto stavu. Pro různé stavy jsou definovány vlastní metody, které přijímají uživatelské příkazy, které jsou platné v daném stavu. Tyto metody rozpoznávají typ příkazu, například `"M"` pro vložení mince, a volají další metody pro provedení příkazu. Pro příkaz typu `"M"` je volána metoda `"sumHandlingController(String command)"`, která kontroluje formát příkazu a hodnotu mince. Pokud je vše v pořádku, mince je přidána do vstupního zásobníku, jinak je vypsáno upozornění pomocí metod třídy `WindowView`. Stejný postup je použit pro všechny uživatelské příkazy. Kromě toho jsou v těchto metodách prováděny kontroly uživatelského vstupu, takže program nespadne při zadání nesprávného vstupu. V této třídě je také mnoho technických metod, které provádějí různé potřebné výpočty. Tyto metody jsou okomentovány ve zdrojovém kódu.

3.5 WindowView

Tato třída obsahuje metody pro výpis zpráv do konzole pro uživatele. Například metoda `startStateView` se volá při přechodu automatu do stavu "START" a slouží k výpisu informací pro tento stav aplikace, jako jsou možné další příkazy.

4. Uživatelská dokumentace

Program je dostatečně jednoduchý k pochopení, protože je intuitivní a výpisy do konzole zobrazují všechny možné další postupy pro daný stav aplikace.

Po spuštění aplikace se objeví startové menu:

```
Aktualni stav: START
Dostupne aktivity:
A-M=N pro vložení mince, kde N je mena mince
A-Z=N pro vyber zboží, kde N je index zboží
A-I pro vypis info o dostupnem zboží
A-S Servisni tlačítko pro naplnění automatu zbožím a odber mincí
```

Pomocí příkazu "A-M=20" uživatel může vložit mince hodnotou 20Kč a tím přejít do stavu "WALLET"

```
Aktualni stav: WALLET
Jste vložil 20 Kč
Dostupne aktivity:
A-M=N pro vložení mince, kde N je mena mince
A-Z=N pro vyber zboží, kde N je index zboží
A-I pro vypis info o dostupnem zboží
A-V Tlačítko <STORNO>
```

Další lze vypsát informace o dostupném zboží pomocí příkazu "A-I"

```
Dostupne zboží:
0 Bageta Pocet: 10X, Cena:55Kc
1 Tycinka Snikers Pocet: 10X, Cena:25Kc
2 Tycinka Kitkat Pocet: 10X, Cena:25Kc
3 Tycinka Knopers Pocet: 10X, Cena:25Kc
4 Napoj Pepsi Pocet: 10X, Cena:20Kc
```

Pomocí příkazu "A-Z=4" lze koupit Nápoj Pepsi

```
A-Z=4
Je vybrano: Nápoj Pepsi
Aktualní stav: ITEM IS SELECTED
Dostupné aktivity:
A-O potvrdit vyber zboží
A-V Tlačítko <STORNO>
|
```

Ted' musíme potvrdit nákup stiskem "A-O"

```
Aktualní stav: CHANGE MONEY
Vracené mince:
Mince: 1 Kč 0X
Mince: 2 Kč 0X
Mince: 20 Kč 0X
Mince: 5 Kč 0X
Mince: 10 Kč 0X

Aktualní stav: DELIVERY OF ITEM
Vydáno zboží: Nápoj Pepsi

Aktualní stav: START
Dostupné aktivity:
A-M=N pro vložení mince, kde N je mena mince
A-Z=N pro vyber zboží, kde N je index zboží
A-I pro vypis info o dostupném zboží
A-S Servisní tlačítko pro naplnení automatu zbožím a odber minci
```

Dostáli jsme zakoupené zboží a vrátili do stavu "START"

5. Závěr

Výsledkem práce je funkční program který má MVC architekturu a je řízený konečným automatem. Samozřejmě má program i prostor pro zlepšení, například mohl bych jsem udělat grafické uživatelské rozhraní v Java FX nebo SWING. Přesto program splňuje zadání a funguje dobře.