TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Evidence skladu

Semestrální práce MTI/ALG2

vypracoval Martin Motejlek

LS 2019/2020

1. Zadání práce

Zapište program pro vedení obsahu skladu a evidenci transakcí. Program bude umožňovat přidávání položek do skladu. Každá položka bude mít svůj název a údaj o počtu kusů. Dále bude program umožňovat zvyšování a snižování počtu kusů položky (transakci), které se bude evidovat. Evidence bude obsahovat čas transakce, název položky a změnu v počtu kusů.

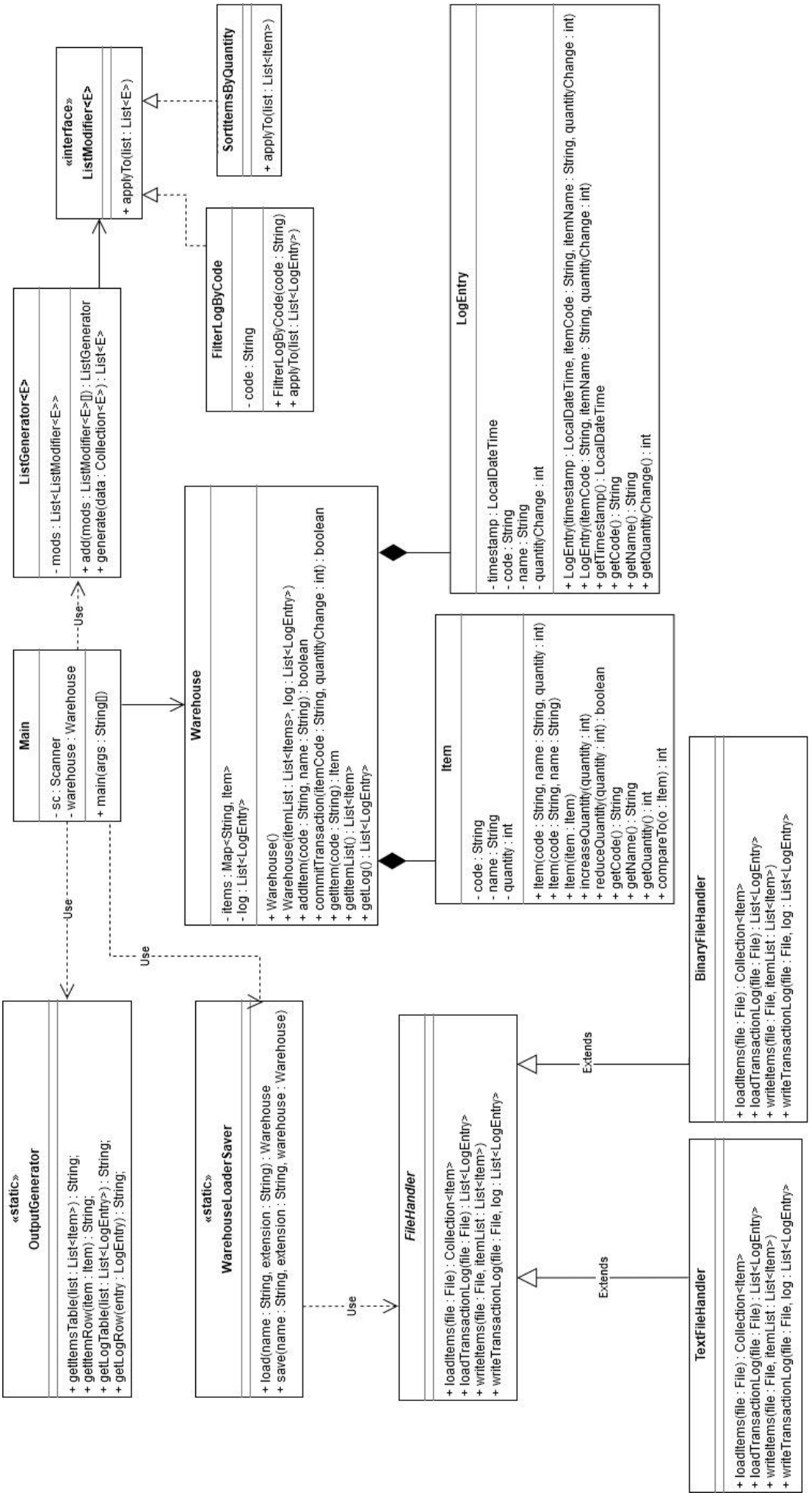
Pro komunikaci s uživatelem bude použita příkazová řádka. Vstupy od uživatele budou ošetřeny tak, aby nedošlo k pádu programu. Program bude nabízet přehledný výpis položek ve skladu a transakcí a možnosti filtrace nebo třídění výpisů. Data bude umožněno zapisovat do textových a binárních souborů a načítat z nich.

2. Návrh řešení

2.1 Funkční specifikace

- Načíst informace o skladu ze souborů (textové, binární)
- Uložit informace o skladu do souborů (textové, binární)
- Přidat nový typ položky do skladu
- Provést transakci (přesunout určitý počet kusů dané položky z/do skladu)
- Vypsat informace o položce podle jejího identifikátoru (kódu)
- Vypsat položky ve skladu
    - Seřadit
        - Podle názvu položky (implicitně)
        - Podle počtu položek
- Vypsat evidenci transakcí
    - Filtrovat
        - Identifikátor (kód) položky

2.2 Objektový model



2.3 Popis struktury souborů

Program pracuje vždy se dvěma soubory zároveň, se souborem s daty o položkách ve skladu a se souborem s evidencí transakcí. Oba soubory mají textovou a binární variantu.

Názvy souborů jsou ve tvaru:

<název>.items.<přípona> – soubory s informacemi o položkách
<název>.log.<přípona> – soubory s evidencí transakcí

Dvojice souborů je načítána a ukládána pomocí názvu a přípony.

.items.txt
Každý řádek souboru obsahuje informace o jedné položce. Jednotlivé informace jsou odděleny libovolným počtem libovolných bílých znaků a jsou uloženy v tomto pořadí a formátu:

- množství (celé číslo)
- identifikační kód
- název (až do konce řádku včetně bílých znaků)

.log.txt
Každý řádek souboru obsahuje informace o jedné transakci. Jednotlivé informace jsou odděleny libovolným počtem libovolných bílých znaků a jsou uloženy v tomto pořadí a formátu:

- datum (yyyy-mm-dd)
- čas (hh:mm:ss)
- změna množství (celé číslo)
- identifikační kód
- název (až do konce řádku včetně bílých znaků)

.items.dat
Data o jednotlivých položkách jsou uložena za sebou v tomto pořadí a formátu:

- String zapsaný pomocí metody writeUTF třídy java.io.DataOutputStream – identifikační kód
- String zapsaný pomocí metody writeUTF třídy java.io.DataOutputStream – název
- 4 byte int –množství

.log.dat
Data o jednotlivých transakcích jsou uložena za sebou v tomto pořadí a formátu:

- 8 byte long – počet dní od Epoch
- 4 byte int – počet sekund od půlnoci
- String zapsaný pomocí metody writeUTF třídy java.io.DataOutputStream – identifikační kód
- String zapsaný pomocí metody writeUTF třídy java.io.DataOutputStream – název
- 4 byte int – změna množství

3. Testovací sada

Soubory se zkušebními daty se jmenují sklad.items.txt, sklad.log.txt, sklad.items.dat a sklad.log.dat. Varianty txt a dat obsahují shodná data. V adresáři se soubory se nachází také zálohy s přidanou koncovkou .backup.

Test vstupních dat

```
Zadejte příkaz "help" pro vypsání nápovědy.
> help
Seznam příkazů:
load <název> <koncovka>
    Načte soubory <název>.items.<koncovka> a <název>.log.<koncovka>.
save <název> <koncovka>
    Uloží soubory <název>.items.<koncovka> a <název>.log.<koncovka>.
item <kód>
    Vypíše informace o zvolené položce.
items [sort-by-quantity]
    Vypíše informace o všech položkách setříděné abecedně podle názvu.
    sort-by-quantity : setřídí vzestupně podle počtu kusů
log [filter-code <kód>]
    Vypíše log transakcí setříděný vzestupně podle času.
    filter-code : vypíše pouze transakce položky se zadaným kódem
add <kód> <název>
    Zaeviduje novou položku do skladu. Název může být víceslovný.
change <kód> <změna množství>
    Provede transakci. Záporné množství pro transakci směrem ze skladu,
    kladné množství pro transakci směrem do skladu.
help
    Vypíše tuto nápovědu.
> neexistuje
Neznámý příkaz.
> add jablko Červené jablko
Položka přidána.
> add jablko Zelené jablko
Položka s daným kódem se již ve skladu nachází.
> add hruska
Nedostatek parametrů.
> add hruska Hruška
Položka přidána.
> add
Nedostatek parametrů.
> item jablko
    0 jablko     Červené jablko
> item jablko něco
Příliš mnoho parametrů.
> item neexistuje
Kód "neexistuje" neodkazuje na žádnou položku.
> item
Nedostatek parametrů.
> change jablko 5
Transakce provedena.
> change hruska 50
Transakce provedena.
> change hruska -10
Transakce provedena.
> change jablko -6
Ve skladu není dostatečný počet kusů pro provedení transakce.
```

```
> change neexistuje 5
Kód "neexistuje" neodkazuje na žádnou položku.
> change jablko 4
Transakce provedena.
> change jablko 1.1
Změna počtu kusů má být celé číslo.
> change jablko 0
Změna množství musí být různá od 0.
> change jablko 100 neco
Špatný počet parametrů.
> change jablko
Špatný počet parametrů.
> change
Špatný počet parametrů.
> items
Seznam položek:
   40 hruska     Hruška
    9 jablko     Červené jablko
> items sort-by-quantity
Seznam položek:
    9 jablko     Červené jablko
   40 hruska     Hruška
> items neco
Neplatný parametr.
> items sort-by-quantity neco
Příliš mnoho parametrů.
> log
Log:
2020-05-29 02:23:35     +5 jablko     Červené jablko
2020-05-29 02:23:40    +50 hruska     Hruška
2020-05-29 02:23:49    -10 hruska     Hruška
2020-05-29 02:24:19     +4 jablko     Červené jablko
> log filter-code jablko
Log:
2020-05-29 02:23:35     +5 jablko     Červené jablko
2020-05-29 02:24:19     +4 jablko     Červené jablko
> log filter-code neexistuje
Log:

> log filtrer-code jablko neco
Neplatný parametr.
> log neco
Neplatný parametr.
> load neexistuje dat
Zadaný soubor neexistuje.
data\neexistuje.items.dat (Systém nemůže nalézt uvedený soubor)
> load sklad neexistuje
Přípona "neexistuje" není podporována.
> load
Špatný počet parametrů.
> load sklad
Špatný počet parametrů.
> load sklad txt neco
Špatný počet parametrů.
> save neco neexistuje
Nepodporovaná koncovka souboru.
```

```
> save
Špatný počet parametrů.
> save neco txt neco
Špatný počet parametrů.
> save example txt
> save example dat
> load sklad txt
> items
Seznam položek:
   15 ananas     Ananas
   30 banan      Banán
    0 broskev    Broskev
    0 celer      Celer
   13 cibule     Cibule
    0 hruska     Hruška
   32 jablko     Jablko
    0 kiwi       Kiwi
    1 kokos      Kokosový ořech
    0 kvetak     Květák
    0 lilek      Lilek
    0 mandarinka Mandarinka
   20 meloun     Meloun
    0 merunka    Meruňka
   60 mrkev      Mrkev
   37 okurka     Okurka
   10 paprika    Paprika
   21 pomeranc   Pomeranč
   15 rajce      Rajče
   20 zeli-hl    Zelí hlávkové
> log
Log:
2020-05-28 21:30:02    +20 ananas     Ananas
2020-05-28 21:30:07    +10 paprika    Paprika
2020-05-28 21:30:12    +15 rajce      Rajče
2020-05-28 21:30:24     +3 pomeranc   Pomeranč
2020-05-28 21:30:34     -5 ananas     Ananas
2020-05-28 21:30:40    +13 cibule     Cibule
2020-05-28 21:31:00   +100 mrkev      Mrkev
2020-05-28 21:31:08    +40 okurka     Okurka
2020-05-28 21:31:14    +20 zeli-hl    Zelí hlávkové
2020-05-28 21:31:23    +18 pomeranc   Pomeranč
2020-05-28 21:31:31    +42 jablko     Jablko
2020-05-28 21:31:38    -13 jablko     Jablko
2020-05-28 21:31:43     +1 kokos      Kokosový ořech
2020-05-28 21:31:50    +20 meloun     Meloun
2020-05-28 21:31:54     -6 mrkev      Mrkev
2020-05-28 21:32:00    -43 mrkev      Mrkev
2020-05-28 21:32:07    +28 mrkev      Mrkev
2020-05-28 21:32:17    +30 banan      Banán
2020-05-28 21:32:49     +5 okurka     Okurka
2020-05-28 21:32:54     -8 okurka     Okurka
2020-05-28 21:33:01     +3 jablko     Jablko
2020-05-28 21:33:15    -19 mrkev      Mrkev
> load sklad dat
```

```
> items
Seznam položek:
   15 ananas     Ananas
   30 banan      Banán
    0 broskev    Broskev
    0 celer      Celer
   13 cibule     Cibule
    0 hruska     Hruška
   32 jablko     Jablko
    0 kiwi       Kiwi
    1 kokos      Kokosový ořech
    0 kvetak     Květák
    0 lilek      Lilek
    0 mandarinka Mandarinka
   20 meloun     Meloun
    0 merunka    Meruňka
   60 mrkev      Mrkev
   37 okurka     Okurka
   10 paprika    Paprika
   21 pomeranc   Pomeranč
   15 rajce      Rajče
   20 zeli-hl    Zelí hlávkové
> log
Log:
2020-05-28 21:30:02    +20 ananas     Ananas
2020-05-28 21:30:07    +10 paprika    Paprika
2020-05-28 21:30:12    +15 rajce      Rajče
2020-05-28 21:30:24     +3 pomeranc   Pomeranč
2020-05-28 21:30:34     -5 ananas     Ananas
2020-05-28 21:30:40    +13 cibule     Cibule
2020-05-28 21:31:00   +100 mrkev      Mrkev
2020-05-28 21:31:08    +40 okurka     Okurka
2020-05-28 21:31:14    +20 zeli-hl    Zelí hlávkové
2020-05-28 21:31:23    +18 pomeranc   Pomeranč
2020-05-28 21:31:31    +42 jablko     Jablko
2020-05-28 21:31:38    -13 jablko     Jablko
2020-05-28 21:31:43     +1 kokos      Kokosový ořech
2020-05-28 21:31:50    +20 meloun     Meloun
2020-05-28 21:31:54     -6 mrkev      Mrkev
2020-05-28 21:32:00    -43 mrkev      Mrkev
2020-05-28 21:32:07    +28 mrkev      Mrkev
2020-05-28 21:32:17    +30 banan      Banán
2020-05-28 21:32:49     +5 okurka     Okurka
2020-05-28 21:32:54     -8 okurka     Okurka
2020-05-28 21:33:01     +3 jablko     Jablko
2020-05-28 21:33:15    -19 mrkev      Mrkev
> exit
```

4. Funkční a technické požadavky

1. Javadoc pro všechny metody a třídy (není potřeba pro zřejmé konstruktory, getry, setry a metody toString)
2. Menu s opakovaným výběrem funkcí aplikace a možností ukončit aplikaci
3. Zformátovaný výpis výsledků do konzole (String.format() a StringBuilder nebo ekvivalentní)
4. Načítání vstupních dat ze souboru
5. Zápis výstupních dat do souboru
6. Možnost práce s textovými i binárními soubory
7. Všechny soubory musí být umístěné v adresáři data
8. Tři balíčky:
    a. ui – třída Main s hlavním programem
    b. app – třídy tvořící logiku aplikace – modely, kontroléry
    c. utils – pomocné třídy, např. vlastní výjimky a rozhraní
9. Vlastní interface
10. Použití java.time API pro práci s časem
11. Použití kontejnerové třídy jazyka Java (ArrayList, LinkedList, HashMap,…)
12. Alespoň dvě možnosti třídění s využitím rozhraní Comparable a Comparator
13. Použití regulárního výrazu
14. Ošetření vstupů od uživatele, aby nezpůsobily ukončení programu – pomocí existujících nebo vlastních výjimek
15. Vhodné ošetření povinně ošetřovaných výjimek
16. Použití vybrané externí knihovny (v této práci vybrán JUnit)