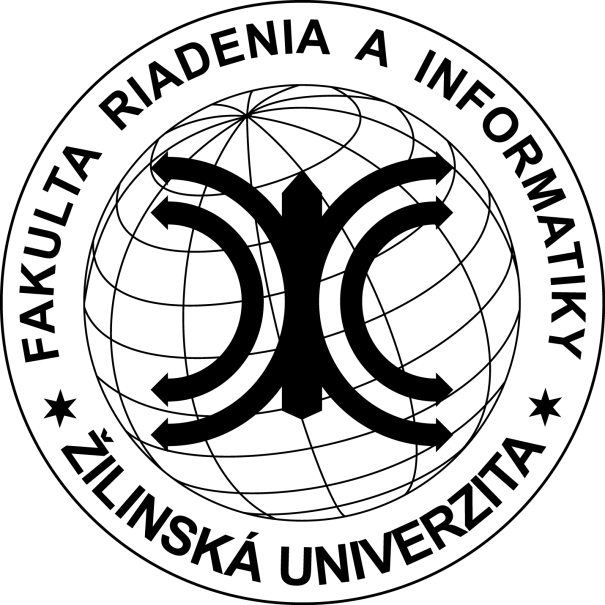
Fakulta riadenia a informatiky

Informačné systémy – Aplikovaná informatika

Údajové štruktúry 2 - Semestrálna práca 1

# Červeno – čierny strom (RB Tree)

Michal Banik

št. skupina: 5ZSA21

os. číslo: 554200

cvičenie: pondelok, blok: 2 - 3

**Zadanie:**

Zadaním práce bolo vytvorenie demonštračnej verzie systému pre správu evidencie stavu dodávok, skladových zásob a odberateľov distribučnej firmy.

**Požiadavky:**

* centralizovaná databáza (v demonštračnej verzii nie je potrebné zabezpečiť vzdialený prístup, je ale potrebný výpis jednotlivých požiadaviek ktoré sa budú realizovať vzdialene)
* všetky dáta sa budú nachádzať v operačnej pamäti
* predbežné hodnoty:
  + veľkosklady: 900
  + odberatelia: >1 000 000
* evidovanie jednotlivých kusov tovarov
* každý veľkosklad pozná svojich odberateľov
* každý odberateľ pozná svoj veľkosklad
* možnosť vytvárania expedícií
  + každý expedovaný tovar vie odkiaľ a kam je prevážaný
* celá databáza sa musí dať vyexportovať do CSV formátu
* implementovať generátor pre naplnenie databázy pre testovanie funkcií

Pre každý veľkosklad evidujte aspoň nasledovné údaje:

* názov
* jednoznačný identifikátor v numerickom tvare (celočíselný identifikátor)
* adresa
* zoznam tovarov, ktoré sú na sklade, prípadne sa expedujú
* zoznam tovarov, ktoré odobral (zoznam všetkých ukončených dodávok tovaru z iných veľkoskladov)

Pre každého lokálneho odberateľa evidujte aspoň nasledovné údaje:

* názov (meno firmy, osoby)
* jednoznačný identifikátor (IČO, rodné číslo, ... – evidujte ako reťazec)
* adresa
* zoznam tovarov, ktoré odobral (zoznam všetkých ukončených dodávok tovaru)

Pre každý tovar evidujte aspoň nasledovné údaje:

* Názov
* EAN kód (evidujte ako reťazec)
* dátum výroby
* dátum najneskoršej spotreby
* jedinečný výrobný kód/číslo šarže (celočíselný identifikátor)
* cenu

Pre každý expedovaný tovar evidujte navyše aspoň:

* EČV auta, ktoré ho preváža
* dátum začatia prevozu
* dátum predpokladaného ukončenia prevozu

Informačný systém musí umožňovať tieto základné operácie (operácie sú zoradené podľa početnosti ich využívania):

1. vyhľadanie zadaného počtu tovarov s konkrétnym EAN kódom a s dátumom najneskoršej spotreby väčším ako zadaný dátum nachádzajúcich sa v zadanom veľkosklade (identifikovaný svojim identifikátorom), ktoré nie sú expedované
2. zistenie počtu tovarov s konkrétnym EAN kódom nachádzajúcich sa v zadanom veľkosklade (identifikovaný svojim identifikátorom), ktoré nie sú expedované
3. vyhľadanie tovaru (identifikovaný svojim výrobným kódom) a výpis všetkých o ňom dostupných informácii (v prípade prebiehajúcej expedície aj informácie o nej) – vrátane informácie o tom, v ktorom veľkosklade sa nachádza
4. naskladnenie (pridanie) tovaru do veľkoskladu (identifikovaný svojim identifikátorom)
5. vyhľadanie lokálneho odberateľa podľa jeho identifikátoru a priradeného veľkoskladu (identifikovaný svojim identifikátorom)
6. vykonanie záznamu o začiatku expedovania daného tovaru (identifikovaný výrobným kódom) do iného veľkoskladu (identifikovaný svojim identifikátorom)
7. vykonanie záznamu o začiatku expedovania daného tovaru (identifikovaný výrobným kódom) k odberateľovi (identifikovaný svojim identifikátorom)
8. vykonanie záznamu o vyložení tovaru (identifikovaný výrobným kódom) – koniec expedovania do veľkoskladu (nepoznáme jeho identifikátor), alebo k odberateľovi (nepoznáme jeho identifikátor)
9. výpis odberateľov zadaného veľkoskladu (identifikovaný svojim identifikátorom) zotriedený podľa ich identifikátorov
10. výpis práve expedovaných tovarov z daného veľkoskladu (identifikovaný svojim identifikátorom)
11. výpis uskutočnených dodávok k zadanému odberateľovi (identifikovaný svojim identifikátorom) priradeného k veľkoskladu (identifikovaný svojim identifikátorom) - požadujú sa informácie: začiatok expedovania, koniec expedovania, EČV
12. výpis uskutočnených dodávok k zadanému veľkoskladu (identifikovaný svojim identifikátorom) - požadujú sa informácie: začiatok expedovania, koniec expedovania, EČV
13. výpis tovarov na danom veľkosklade (identifikovaný svojim identifikátorom), ktorým končí spotreba do zadaného počtu dní (napr. do 5 dní) od zadaného dátumu
14. pridanie veľkoskladu
15. pridanie odberateľa veľkoskladu (identifikovaný svojim identifikátorom)
16. vyradenie konkrétneho tovaru z evidencie (identifikovaný výrobným kódom)
17. výpis celkového množstva a sumárnej hodnoty tovarov (podľa EAN kódov) pre daný veľkosklad (identifikovaný svojim identifikátorom), to znamená uviesť koľko kusov tovaru sa nachádza v sklade s daným EAN kódom a aká je sumárna cena za tieto tovary
18. zrušenie veľkoskladu (identifikovaný svojim identifikátorom) - celá jeho agenda sa presunie do iného veľkoskladu (identifikovaný svojim identifikátorom)
19. zrušenie odoberateľa (identifikovaný svojim identifikátorom) veľkoskladu (identifikovaný svojim identifikátorom)

**Vypracovanie**

Ako štruktúra použitá pre túto databázu bola vybraná štruktúra červeno-čierny strom (Red Black Tree). Súčasťou tejto práca bola aj samotná implementácia tejto štruktúry.

**Dátový model databázy:**

Logická štruktúra databázy:



**Výpočtová zložitosť operácií:**

Nech máme v systéme:

* *n* produktov (*e* rôznych EAN kódov, *d* rôznych minimálnych dátumov, *p* rôznych výrobných čísel) = *e+d+p*
* *c* odberateľov
* *w* skladov

Zložitosť operácie

1. *log(w) + log(e) + log(d) + p*
2. *log(w) + e + d + p = log(w) + n*
3. *log(n)*
4. *log(e + d + p) = log (n)*
5. *log(c)*
6. *log(w) + log(n) + ( log(n) + 1 ) ; log(n) + 1* = vymazanie z naskladnených a vloženie do transakcií
7. *log(w) + log(n) + ( log(n) + 1 )*
8. *log(w) + log(n)*
9. *log(w) + c*
10. *log(w) + n*
11. *log(c) + n*
12. *log(w) + n*
13. *log(w) + log(e) + log(d) + p*
14. *log(w)*
15. *log(w) + log(c)*
16. *log(n) + log(n)*
17. *log(w) + n*
18. *log(w) + log(w) + n*
19. *log(c)*

Diagramy tried celého systému:













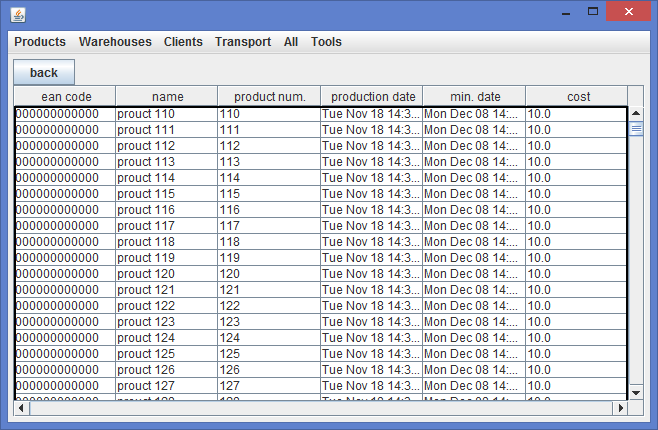
**Implementácia:**

Na základe navrhnutého dátového modelu boli vytvorené všetky potrebné štruktúry a triedy objektov.

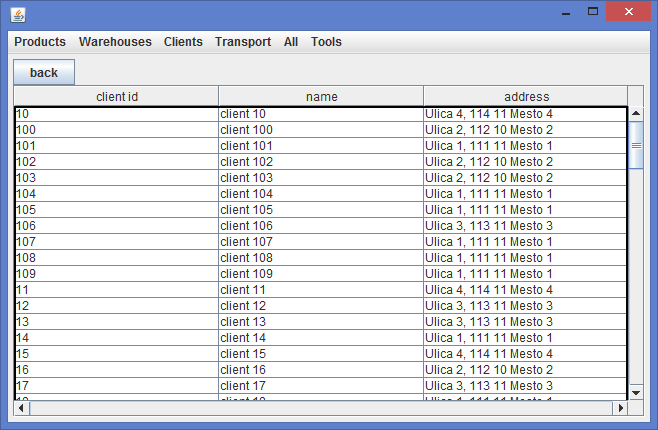
Implementácia samotnej databázy bola rozdelené do viacerých balíčkov:

* rb
  + samotná implementácia dátovej štruktúry červeno-čierny strom
* core
  + implementácia samotnej databázy
  + *Db,java*
    - obaľuje celú databázu a ponúka prístup k funkciám databázy
    - obsahuje referencie na hlavné dátové stromy:
      * *RBTree<Integer> warehousesById*
        + obsahuje všetky sklady v databáze
      * *RBTree<Integer> itemsByProductNumber*
        + obsahuje všetky kusy tovarov, ktoré sú buď v nejakom sklade alebo sú práve expedované a zatiaľ nedoručené
      * *RBTree<String> allClients*
        + obsahuje všetkých odberateľov v systéme
    - navyše implementuje inicializačnú metódu pre základné naplnenie databázy
  + *ClientNode.java, WareHouseNode.java, TransportNode.java*
    - reprezentácie jednotlivých prvkov systému v štruktúre RBTree
  + *DateNode.java, EanNode.java, ProductNumberNode.java*
    - reprezentácia pomocných medzištruktúr použitých pre triedenie jednotlivých tovarov – zabezpečené triedenie podľa dátumu, EAN kódu a výrobného čísla tovaru
  + *StorageDatabase.java*
    - rozhranie opisujúce funkcionalitu ktorú má zabezpečovať databáza
  + *CSVHandler.java*
    - trieda zabezpečujúca exportovanie do CSV formátu a načítavanie z vytvorených CSV súborov
* core.data
  + obsahuje základné dátové triedy v databáze
  + *PrductPlace.java*
    - rodičovský prvok tried Client a Warehouse
    - obsahuje štruktúru so všetkými dodanými tovarmi – t. j. ukončené expedície do tohto miesta
  + *Client.java*
    - reprezentácia odberateľa skladu
    - obsahuje referenciu na veľkosklad, ktorému je priradený
  + *Product.java*
    - reprezentuje jeden kus tovaru v sklade
    - okrem základných dátových prvkov obsahuje navyše:
      * aktuálne miesto kde je umiestnený produkt – ak nie je práve expedovaný
      * referencia na aktuálnu expedíciu – ak prebieha expedícia tohto kusu tovaru
      * referencie na pomocné stromy zabezpečujúce triedenie tovarov (*dateTree, eanTree, productNumbersTree*)
  + *WareHouse.java*
    - reprezentácia veľkoskladu v systéme
    - obsahuje referencie na štruktúry:
      * *RBTree<String> storedByEan*
        + štruktúra pre zabezpečenie triedenia produktov
        + prvok stromu obsahuje tovar s rovnakým EAN kódom
      * *RBTree<Integer> dispatchedByPN*
        + prvok stromu obsahuje jednu neukončenú expedíciu do iného skladu alebo ku odberateľovi
      * *RBTree<String> clientsById*
        + štruktúra obsahujúca všetkých priradených odberateľov
  + *TransportProduct.java*
    - reprezentácia expedície jedného kusu tovaru z jedného skladu do iného alebo ku odberateľovi
    - v inštanciách tejto triede je zaznamenaná informácia o expedovaní produktu
      * referencia na konkrétny produkt
      * referencia na cieľové miesto a miesto odkiaľ bola expedícia vykonaná
      * dátum vykonania expedície
      * očakávaný dátum prijatia
      * skutočný dátum prijatia
      * EČV aut, ktorá vykonáva expedíciu
* gui a gui.tables
  + implementácia grafického rozhrania pre databázu a pomocné triedy pre zobrazovanie výsledkov jednotlivých funkcií
  + zabezpečuje vyvolanie jednotlivých operácií pomocou grafických okien
* tests
  + implementácia JUnit testov pre otestovanie všetkých operácií

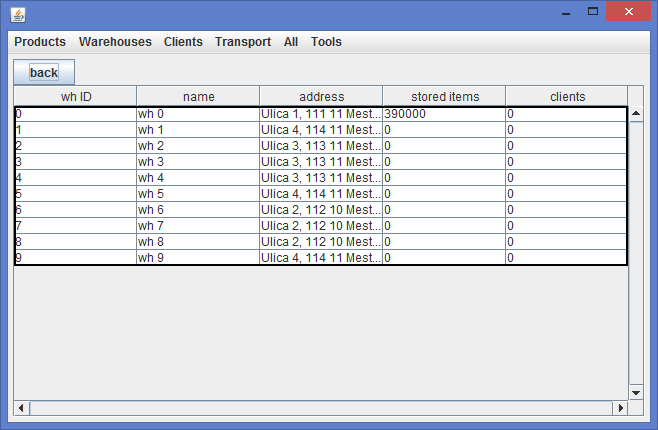
**Ukážkové výstupy systému:**



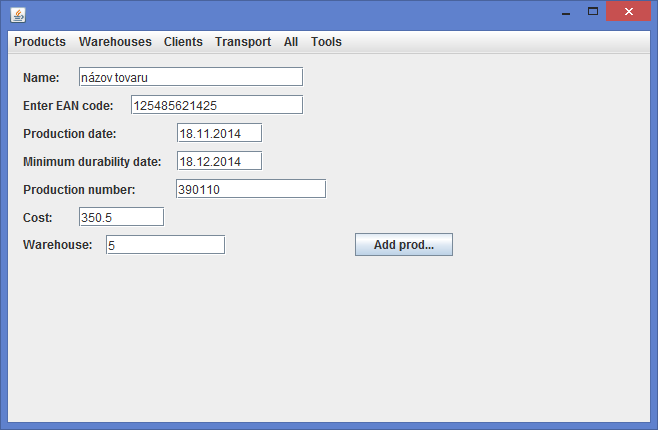
Obrázok 1 výpis všetkých produktov



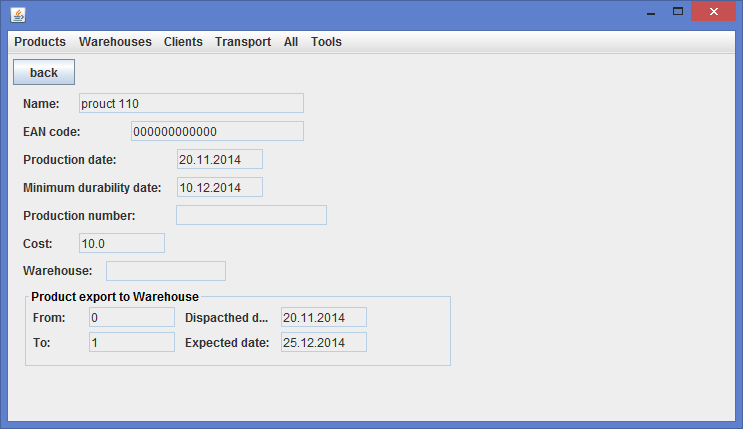
Obrázok 2 výpis všetkých odberateľov



Obrázok 3 výpis všetkých skladov



Obrázok 4 pridanie tovaru



Obrázok 5 zobrazené informácie o vyhľadanom tovare

**Záver:**

Požiadavky zadania boli splnené nasledovne v implementácii:

* Databáza implementuje všetky požadované funkcie a výstupy vie zobraziť v implementovanom užívateľskom rozhraní.
* Žiadne dáta neukladá počas behu do súborov, a teda všetky dáta s ktorými pracuje sú len v operačnej pamäti.
* Každý kus tovaru je evidovaný v jeho priradenom sklade. Po dokončení expedície na iné miesto je tento tovar vymazaný z pôvodného skladu. V prípade že je tovar expedovaný k odberateľovi, po ukončení expedície je tento tovar dostupný len cez štruktúru priradenú konkrétnemu klientovi (nie je dostupný v systému, tak ako ostatný naskladnený tovar).
* Každý sklad pozná svojich odberateľov a každý odberateľ pozná sklad, ku ktorému je priradený.
* Expedície sú vytvárané zo skladu kde sa nachádza tovar. Po začatí expedície je tento tovar odstránený z naskladnených tovarov a po ukončení expedície (po doručení do cieľa) je expedovaný produkt odstránený zo záznamov skladu.
* Exportovanie databázy zabezpečuje trieda CSVHandler.java ktorá využíva na zápis do súboru voľne dostupnú knižnicu opencsv-3.1.jar (<http://opencsv.sourceforge.net/>). Exportovaná databáza je uložená v aktuálnom priečinku odkiaľ je aplikácia spúšťaná. Pri exporte sa vytvorí priečinok *csvExport*, kde sú vytvorené súbory:
  + arrivedToClients.csv – tovar doručený ku všetkým odberateľom
  + arrivedToWH.csv – tovar doručený do všetkých skladov
  + clients.csv – všetci odberatelia skladov
  + dispatchedItems.csv – záznamy o expedíciách
  + productsByPN.csv – všetok naskladnený aj expedovaný tovar
  + warehouses.csv – všetky sklady

Generátor pre testovanie databázy pridá do databázy:

* 10 skladov
* 10 odberateľov pre každý sklad
* 130 rôznych EAN kódov
* 20 rôznych dátumov minimálnej trvanlivosti tovaru v rozsahu od dnes+20 dní do dnes +40 dní
* 150 produktov pre každú kombináciu EAN kódu a dátumu

Celá aplikácia je v spustiteľnom archíve sem1.jar, obsahujúcom všetky zdrojové kódy a použité knižnice.