

Kamra 1.0

Anyagköltség kalkulátor és készletkezelő program éttermek számára

Szakács István étterem tulajdonos szeretné kézben tartani éttermének alapanyag felhasználását. Olyan programra van szüksége, amely

- megadja az étlapon szereplő egyes ételek alapanyag-költségét
- készlet nyilvántartást végez:
 - az eladott ételekkel csökkenti
 - a beszerzésekkel növeli a készletet
- Megmondja, hogy egy étel rendelés teljesíthető-e a készlet alapján
- Rendezvény költség kalkulációt lehessen vele készíteni. Az összeállított menüsor és adagmennyiségek alapján megadja az alapanyagok önköltségét.
- Figyelmeztessen a szükségessé váló beszerzésekre, ha valamely alapanyagból a készlet mennyisége a beállított szint alá megy.
- Különböző lekérdezéseket lehessen tenni:
 - megadott intervallumban az egyes alapanyagokból mennyi fogyott
 - mely ételekből mennyi fogyott
 - Leltár funkció: mekkora az eltérés a felvett leltár és a nyilvántartás adatai között

A program bővíthető egyéb költség befolyásoló tényezők figyelembevételének hozzáadásával. Pl.: egyes ételekhez nem kapcsolódó anyagköltségek (pl.: sütóolaj), energia-költségek, bérköltségek stb. Szavatossági idő figyelembevétele (csak addig használható fel egy alapanyag, amíg nem járt le a szavatossága).

A program felépítése, funkciói

Adminisztráció

- receptek felvitele / módosítása / törlése
- alapanyag fajták felvitele (neve, mennyiség egysége, min. mennyiség)

Rögzítés

- eladott ételek felvitele (dátum, étel, mennyiség (db))
- beszerzések felvitele (dátum, alapanyag, mennyiség, beszerzési ár)

Lekérdezések

- Kiválasztott étel önköltsége
- Összes készlet lekérdezése
- Egy kiválasztott anyag lekérdezése
- Fogytán lévő anyagok lekérdezése
- Statisztikák

Rendezvény kalkulátor

- a kiválasztott ételek és adagok alapján megadja az anyagköltséget

Tervezett adatbázis szerkezet:

Alapanyag

- id (int)
- nev (String)
- me (String)
- min (double)

Etel

- id (int)
- nev (String)
- elad_ar (int)

Recept

- etel_id (int)
- alap_id (int)
- mennyiség (int)

Beszerzes

- id (int)
- alap_id (int)
- datum (date)
- mennyiség (double)
- egysegar (int)
- atvezetve (boolean)

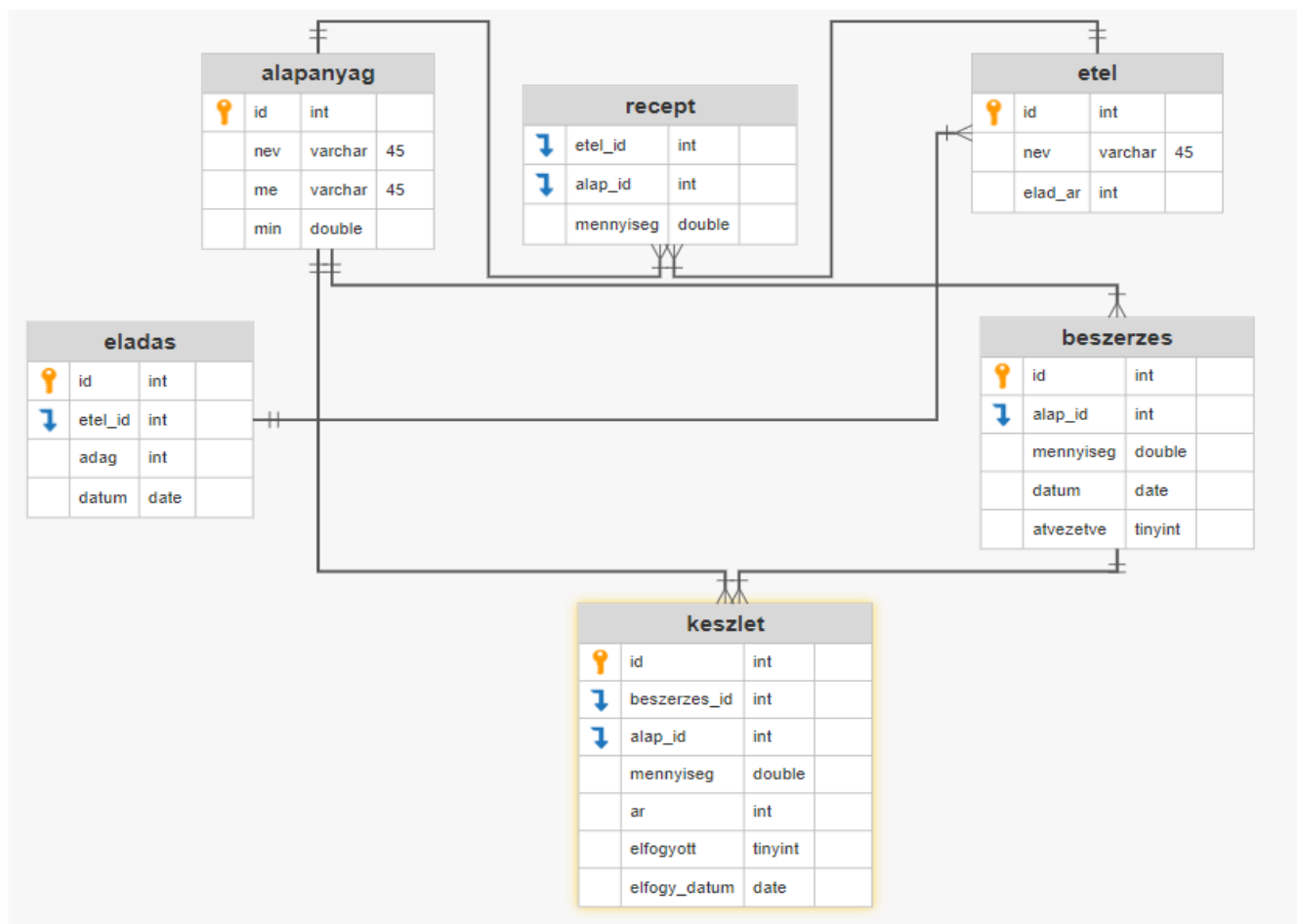
Eladas

- id (int)
- etel_id
- adag (double)
- datum (date)
- atvezetve (boolean)

Keszlet

- id
- beszerzes_id
- alap_id
- mennyiség
- ar
- elfogyott (boolean)
- elfogy_datum (date)

ERD Diagram



Alapanyag beszerzéskor először a „Beszerzes” táblába kerülnek be a vásárolt tételek. Hasonlóképpen eladáskor az „Eladas” táblába.

A készletben egy *átvezetés* funkció használatával frissülnek az adatok. Minden beszerzés új sorként szerepel a készletben. Egy alapanyagból így több eltérő árú is lehet. Eladáskor az „eladas” táblával egyidejűleg a „készlet” tábla is frissül az adott étel alapanyagszükségletének megfelelően, amely a „recept” táblában található. Mindaddig a kisebb id-jű mennyiségét csökkentjük (használjuk fel) a készletből, amíg az el nem fogy. Ha a mennyiség 0 lesz, akkor: *elfogyott* = true, és beírjuk a dátumot. Külön funkcióként lehet törölni is a 0-ás mennyiségű tételeket. Árképzés esetén így mindig a korábbi beszerzési árat tudjuk először használni.

A program konkrét megvalósítása még nem teljesen kristályosodott ki bennem.

Amire eddig gondoltam, hogy kellene fog egy DAO osztály az adatbázis műveletkehez. A tanfolyami példák alapján ez lenne `KamraRepositoryJDBCimpl.java`. Ennek interface-e a `KamraRepository.java`. A kivételkezelő osztály lehetne a `KamraDAOException.java`.

A további osztályszerkezetben nem vagyok teljesen biztos. Az egyes adattáblákhoz hozzak létre egy-egy osztályt? Vagy a tervezett menüszerkezet legyen az osztályok alapja?

Továbbá a grafikus, Swing-es felület létrehozásával kapcsolatban is sok még bennem a kérdés. Hogyan álljak neki ennek a megtervezéséhez?

Első körben tehát a koncepcióval, a tervezett adatbázis-szerkezettel, és az imént említettekkel kapcsolatban szeretnék oktatói útmutatást kérni.