

Harjoitustyö 2 - Karkkitukkuimplementaatio

Verkon välityksellä toimiva karkkitukku **nekku ja nani** on huomannut, ettei varastokirjanpito yrityksessä pysy ajantasalla. Yritys tarvitsee kipeästi pientä tietokoneohjelmaa, jonka avulla voi pitää kirjaa karkkien myynnistä, varaston täydennyksistä ja varaston tilasta. Karkit ovat varastossa laatikoittain. Tartut tuumasta toimeen ja lähdet toteuttamaan ohjelmaa.

Keskityt tässä ensimmäisessä toteutusvaiheessa tekemään tietorakenteen, joka pystyy tehokkaasti huolehtimaan varastosta. Karkkivarastoon pitää voida mahdollisimman sujuvasti, keskimäärin $O(1)$ ajassa, voida lisätä ja poistaa karkkilaatikoita sekä kysyä, onko haluttua karkkia riittävästi varastossa.

Karkkivarasto

Tukun varastossa on tyypillisesti n. tuhatta (1000) erilaista makeista saatavilla. Ohjelman on kuitenkin voitava käsitellä suurempaakin varastoa eli se ei voi rajoittaa valikoiman suuruutta. Varastossa olevat makeiset on jaettu makeisryhmiin. Käytössä olevat ryhmät sekä niiden varastotunnukset näet taulukosta 1.

Taulukko 1: Varaston makeisryhmät

Tuoteryhmä	Tunnus
Lakritsit	LKS
Purukumit	GUM
Salmiakit	SMK
Sekalaiset	MIX
Suklaat	SUK
Tikkarit	TIK
Vahtokarkit	VHT
Viinikumit	WIN

Tuotekoodi

Jokaisella makeisella on oma, uniikki, tuotekoodinsa. Koodissa on varastotunnus, valmistajan ID ja tuote ID. Koodi rakentuu seuraavasti:

$$\underbrace{ABC}_{\text{Varastotunnus}} : \underbrace{nnnn}_{\text{valmistajan ID}} : \underbrace{mmm}_{\text{Tuote-ID}}$$

Varastotunnus on taulukossa 1 oleva kolmikirjaiminen tunnus. Valmistajan ID **nnnn** on positiivinen kokonaisluku 1000-9999 ja tuote-ID **mmm** on positiivinen kokonaisluku 100-999. Tunnisteen osat erotellaan toisistaan kaksoispisteellä.

Varastotiedosto

Tukun karkkivaraston tila on tallennettuna tiedostoon `varasto.txt`. Ohjelman aluksi tämän tiedoston tiedot luetaan tietorakenteeseen.

Tuotekoodin lisäksi jokaisesta tuotteesta tiedetään varastosaldo, hyllypaikka ja tuotenimi. Varastosaldo on ei-negatiivinen kokonaisluku. Hyllypaikka määritellään koodilla `xx.yy`, jossa sekä `xx` että `yy` ovat kokonaislukuja 1-99. Tuotenimi on merkkijono, joka voi sisältää välilyöntejä ja väliviivoja, mutta ei muita välimerkkejä. Tiedostossa jokaisen tuotteen tiedot on omalla rivillään. Tiedoston formaatti on siis:

```
tuotekoodi;lukumäärä;hyllypaikka;tuotenimi
```

eli

```
abc:nnnn:mmm;lk;xx.yy;abc def
```

Esimerkiksi:

```
LKS:1234:123;152;42.2;Panda Lakupala
GUM:2345:222;500;58.92;Jenkki Pro Freshmint
VHT:9876:321;5;12.1;Jet-Puffed Marshmallows
SMK:1212:965;1234;8.97;Tyrkisk Peber
...
```

Ohjelman toiminta

Ohjelman tunnistamat komennot on esitelty parametreineen ja kuvauksiineen taulukossa [2](#). Parametreista `<ID>` on edellä kuvatus mukainen tuotekoodi, `<n>` on kokonaisluku ja `<p>` edellä kuvatus mukainen hyllypaikka, `<k>` on merkkijono, joka voi sisältää välilyöntejä, mutta ei muita välimerkkejä.

Valmiit osat

Harjoitustyössä harjoitellaan tietorakenteen toteuttamista itse. Toteuta tietorakenteesi `.cc`-päätteiseen tiedostoon. Pohjan tiedostolle saat versionhallinnasta. Tämä toteutetaan käytännössä antamalla toteutus valmiina annetulle luokkarajapinnalle.

Valmiina tarjotaan pääohjelma, joka toteuttaa syötteen käsittelyn, tietorakenteen luomisen sekä tietorakenteen jäsenfunktioiden kutsumisen. Sen tehtävä on testata toteutettua toiminnallisuutta. Pääohjelmaa ei tarvitse eikä tulekaan muuttaa.

Taulukko 2: Ohjelman ymmärtämät komennot

Komento	Kuvaus
R <F>	Lukee varastotilanteen tiedostosta <F>. Jos tiedostonimeä ei ole annettu, lukee tilanteen tiedostosta varasto.txt.
A <ID> <p> <n> <k>	Lisää makeista <ID> ja nimeltä <k> <n> laatikkoa hyllypaikkaan <p>. Jos makeista <ID> on jo saatavilla, lisää uudet sen yhteyteen. Voit olettaa, että hyllypaikka ei muutu.
D <ID> <n>	Vähentää makeista <ID> <n> laatikkoa. Tulostaa jäljelle jääneen varastosaldon ja tuotteen hyllypaikan. Jos makeista <ID> ei ole varastossa riittävästi, tulostaa virhekoodin EI_VARASTOA, varastosaldon ja hyllypaikan. Jos makeista <ID> ei ole varastossa, tulostaa virhekoodin EI_SAATAVILLA. Jos varasto tyhjenee, poistaa makeisen varastosta.
F <ID>	Selvittää, onko tuotetta <ID> varastossa. Tulostaa tuotenimen, tuotteen hyllypaikan ja varastosaldon. Jos makeista <ID> ei ole varastossa, tulostaa virhekoodin EI_SAATAVILLA.
C	Tulostaa varastossa olevien erilaisten makeisten määrän.
E	Tyhjentää tietorakenteen.
Q	Lopettaa ohjelman. Tulostaa varaston tilanteen tiedostoon varasto.txt.

Määritelty rajapinta annetaan valmiina C++:lla. Rajapinnan funktioiden toteutukset ja tarvittavat jäsenmuuttujat on toteutettava itse. Julkista rajapintaa ei saa muuttaa.

Valmiit osat ovat saatavilla opiskelijan harjoitustöitä varten saamassa git-repositoriossa. Lisäksi versionhallinnassa annetaan ohjelman testaamiseen käytettävä testiaineisto, jota vastaan harjoitustyön toiminnallisuutta arvioidaan.

Virhetilanteet

Pääohjelma huolehtii virhetilanteista sikäli kun ne liittyvät komentojen parametreihin. Osa virheilmoituksista on käytännön syistä tulostettava tietorakenteen puolella, nämä on kuvattu seuraavassa. Virhekoodit on määritelty datastructure.hh:ssa vakioina.

- Jos etsittyä makeista ei ole varastossa riittävästi, tulostaa ohjelma virhekoodin EI_VARASTOA, makeisen saatavilla olevan varastosaldon ja hyllypaikan.
- Jos etsittyä makeista ei ole varastossa lainkaan, ohjelma tulostaa virhekoodin EI_SAATAVILLA.

Esimerkki ohjelman toiminnasta

```
> R
> A LKS:1234:123 42.2 50 Panda Lakupala
> A MIX:3679:999 1.11 25 Re-mix
> A SUK:8787:324 93.93 124 Suffeli
> F LKS:1234:123
Panda Lakupala 150 42.2
> F WIN:5654:236
Tuotetta ei varastossa.
> D SUK:8787:324 22
Saldo: 102 Hylly: 93.93
> D MIX:3679:999 58
Varastosaldo ei riittävä.
Saldo: 25 Hylly: 1.11
> C
854 erilaista makeista.
> Q
```

Rajoitteet

C++11:n standardikirjastoa saa käyttää soveltuvien osien. Kuitenkin assosiativisten säiliöiden `unordered_map`, `unordered_set`, `map` ja `set` sekä näiden `multimap`/`set`-versioiden käyttö on kiellettyä. Samoin sarjoista `list` on kielletty. Muut sarjarakenteet käyvät, kunhan niitä käytetään vakiokokoisina.