

OHJ-1151 Ohjelmointi Iie, tentti

16.05.2008, Essi Lahtinen

Kirjoita jokaiseen vastauspaperiisi selkeästi ylläolevat otsikkotiedot, nimesi, opiskelijanumerosi, koulutusohjelmasi, vuosikurssisi, vastauspaperin järjestysnumero ja jättämiesi vastauspapereiden kokonaismäärä. Jos olet korottamassa aikasempaa suoritustasi, kerro myös, miltä vuodelta suoritukseksi ovat. *Jätä vastauspapereihisi vähintään 3cm ulkoreunamarginaalit. Jätä ensimmäiseen paperiin vähintään 10 cm yläreunamarginaali tai piirrä valmis taulukko tehtävien pistemäärille.* Kirjoita selkeällä käsialalla. Tentin tarkastaja ei ole hieroglyfien erikoisasantuntija. Onnea tenttiin!

Tehtävä 1

Selitä *lyhyesti*, mitä seuraavat termit tarkoittavat:

1. luokka (*class*) (1p.)
2. kopiorakentaja (*copy constructor*) (1p.)
3. muistivuoto (*memory leak*) (1p.)
4. moduuli (*module*) (1p.)
5. riippuvuus Makefilessä (*dependency in Makefile*) (1p.)
6. oletusparametri (*default parameter*) (1p.)

Tehtävä 2

Vastaa *lyhyesti*, esimerkiksi ranskalaisia viivoja käyttäen.

1. Millaista STL:n säiliöistä ja muista C++:n rakenteista koottua tietorakennetta käyttäisit tehdessäsi ohjelmaa, joka tallentaa tietoja erilaisista viineistä? Viinipullojen tiedot pitää pystyä tulostamaan nimien mukaisessa aakkosjärjestyksessä. Toisaalta tiedot pitää pystyä tulostamaan myös siinä järjestyksessä, missä pullot sijaitsevat omistajan viinikellarin hyllyillä rivissä. Tärkeiden tietojen kuten vuosikerran ja rypälelajikkeen lisäksi jokaiseen viiniin liittyy joukko kuvaavia sanoja (esim. pyöreä, viipyilevä ja hieman pirskahteleva), jotka halutaan myös tulostaa aina aakkosjärjestyksessä. Kirjoita rakenteiden määrittelyt C++:na. Tarpeen vaatiessa voit myös piirtää rakenteesta kuvan. (4p.)
2. Millä tavoin tieto-ohjattua ohjelmointia (*data driven programming*) olisi voinut hyödyntää tekemäsi Ohjelmointi 2e:n harkkatyön komentoriviparametrien käsittelemisessä? Mitä hyötyä tästä olisi ollut varattuna jollain muulla tavalla toteutettuun komentoriviparametrien käsittelyyn? (2p.)

Tehtävä 3

Matlabin käyttöliittymässä on muutamia ärsyttäviä ominaisuuksia, joiden vuoksi haluat aivan ehdottomasti toteuttaa oman paremman ohjelman matriisilaskujen tarkastamiseen. Haluat toteuttaa ohjelmassasi matriisin abstraktina tietotyypinä. Suunnittele ja kirjoita C++:na alustavat versiot abstraktin tietotyypin rajapinnoista. (6p.)

Tehtävä 4

Oletetaan, että määriteltynä on tietuetyyppi `Alkio`, joka sisältää mm. `Alkio*`-tyyppisen kentän `seuraava` ja `int`-tyyppisen kentän `data`. Lisäksi `alku`-nimisen osoittimen päähän on edellä kuvatuista dynaamisesti varatuista alkioista muodostettu linkitetty lista.

1. Oletetaan, että tarvittavat `include`-direktiivit jne. ovat käytössä. Minkä toimenpiteen seuraava ohjelmapätkä suorittaa kuvatulle tietorakenteelle, tai toisin sanottuna miten tietorakenne muuttuu kun koodipätkä suoritetaan? (1p.)

```
Alkio* tmp = alku;
while( tmp->seuraava->seuraava != 0 ) {
    tmp = tmp->seuraava;
}
tmp->seuraava->seuraava = alku;
alku = alku->seuraava;
tmp->seuraava->seuraava->seuraava = 0;
tmp = 0;
```

2. Ohjelmapätkää toteutettaessa on tehty hölmö ratkaisu, jonka vuoksi se toimii joissain tilanteissa väärin. Minkälaisissa tilanteissa se toimii väärin (1p.)
3. Miten virhe ilmenee? (1p.)
4. Korjaa koodipätkä niin, että et vain lisää `if`-rakennetta, joka tekee ylimääräisen tarkastuksen, vaan niin, että muutat algoritmin järkevämmän toimivaksi. (2p.)
5. Miksi ohjelmapätkän viimeinen rivi on tärkeä? (1p.)

Tehtävä 5

1. Kirjoita rekursiivinen funktio, joka laskee linkitetyn listan alkioiden sisältämien `data`-kenttien summan. Voit käyttää esim. samanlaisia alkioita, kuin tehtävän 4 linkitetyssä listassa oli. (3p.)
2. Onko edelliseen kohtaan kirjoittamiasi funktio häntärekursiivinen vai ei? Perustele. (1p.)
3. Mitä etua häntärekursiosta on verrattuna ei-häntärekursioon? (2p.)