

Loppuentti, 23.2.2007 klo 8-12 luentosalissa ML1/ML2.

HUOM! Vasta vain neljään (4) tehtävään.

- Kirjoita funktio, joka laskee argumenttina saamastaan double-taulukosta for- silmukkaa käyttäen keskiarvon ja palauttaa sen (4p).
 - Kirjoita funktio, joka etsii argumenttina saamastaan double-taulukosta while-silmukkaa käyttäen suurimman arvon eli maksimin ja palauttaa sen (4p).
 - Kirjoita pääohjelma, joka testaa edellä ohjelmoituja funktioita (2p).
- Vastaa lyhyesti seuraaviin kysymyksiin.
 - Generoidaanko luokkamalleista (class template) luokkia ohjelman käännöksen vai suorituksen aikana? Mikä on yleisin käyttötarkoitus luokkamalleille? (2p)
 - Mainitse kaksi STL:n säiliöluokkamallia nimeltä ja kerro kummallekin jokin tyypillinen käyttötarkoitus. (1+1p)
 - Mikä on iteraattori (iterator)? Mitä C++:n tietotyyppiä se läheisesti muistuttaa? Kirjoita ohjelmanpätkä, jossa tulostat ruudulle jostakin STL:n säiliöluokkamallista generoidun luokan alkiot ensimmäisestä viimeiseen käyttäen iteraattoria. (2p)
 - Mikä on tietotyyppien tietue (struct) ja luokka (class) oleellisin ero C++ kielessä? (2p)
 - Kerro lyhyesti jokaisen rivin merkitys allaolevassa määrittelyssä. Mitä funktio `for_each` siis tekee yhdellä lauseella sanottuna? (2p)

```
template<class Iter, class Op>
    Op for_each(Iter n, Iter L, Op o) {
    while (n != L) {
        o(*n);
        ++n;
    }
    return o;
}
```

- Määrittele luokka `Henkilo`, jossa on ominaisuudet: henkilötunnus, etunimi, sukunimi, puhelinnumero ja sähköpostiosoite sekä metodit, joilla pystytään tulostamaan henkilön tiedot ja metodit, joilla pystytään

vaihtamaan muuttuneet tiedot (sukunimi, puhelinnumero ja email). (7p)

Lisää henkilöluokkaan muodostimet seuraaville tapauksille: kaikki tiedot annetaan olion luonnin yhteydessä, henkilötunnus ja nimi annetaan olion luonnin yhteydessä, vain henkilötunnus annetaan olion luonnin yhteydessä. (3p)

4. Tarkastellaan virtuaalifunktioiden käyttöä yksinkertaisessa perintähierarkiassa.
 - a) Määrittele luokka `Kuvio`, joka sisältää konstruktorin, destruktorin ja puhtaan virtuaalifunktion `A1a`. Miksi luokalle on joka tapauksessa järkevää kirjoittaa destruktori (4p).
 - b) Peri luokat `Ympyra` ja `Suorakaide` luokasta `Kuvio`. Kirjoita perityille luokille järkevät konstruktorit ja lisää luokkiin jäsenmuuttujat, joita tarvitaan pinta-alan laskentaan. Kirjoita kumpaankin luokkaan toteutus jäsenfunktiolle `A1a`, joka palauttaa kyseisen kuvion pinta-alan (2+2p).
 - c) Kirjoita pääohjelma, jossa kysytään käyttäjältä kuvion tyyppi ja sen jälkeen alan laskemiseen tarvittavat tiedot (ympyrän säde tai suorakulmion sivujen pituudet). Ohjelma tulostaa pinta-alat ruudulle. (2p)
5. Tutkitaan floating point lukujen matriiseja. Toteuta C++:lla luokka `Matrix`, jonka matriiseille on määritelty seuraavat jäsenfunktiot: konstruktori, destruktori, `areAdditive`, `areMultipliable`, `getDim` `add (+)`, `mul (*)`.

Konstruktorille annetaan parametrina matriisin rivien määrä ja sarakkeiden määrä. Konstruktori varaa tilan matriisin alkiolle ja alustaa alkiot nolliksi. Destruktori vapauttaa varatun muistitilan.

Jäsenfunktioilla `areAdditive`, voidaan selvittää ovatko kaksi matriisia yhteenlaskettavia: onko matriiseilla samat dimensiot.

Jäsenfunktioilla `areMultipliable`, voidaan selvittää ovatko kaksi matriisia kerrottavissa keskenään: onko vasemman puolen matriisin sarakkeiden määrä sama kuin oikean puolen matriisin rivien määrä.

Jäsenfunktioilla `getDim` voidaan selvittää matriisin dimensiot: rivien määrä ja sarakkeiden määrät.

Jäsenfunktio `add` laskee yhteen kaksi matriisia.

Jäsenfunktio `mul` suorittaa kahden matriisin kertolaskun.

Kirjoita luokan `Matrix` luokkamäärittely ja toteuta sen yllämainitut jäsenfunktiot.

Jäsenfunktiot `add` ja `mul` saa toteuttaa myös ylikuormitettuna operaattoreina `+` ja `*`.