

OHJ-1450 Olio-ohjelmoinnin jatkokurssi

Tentti 18.9.2006

Tentissä ei saa käyttää ylimääräistä kirjallista materiaalia, laskimia, tietokoneita tai muita lunttausvälineitä.

Muutama sana tenttivastauksen kirjoittamisesta:

1. Mieti etukäteen esim. ranskalaisilla viivoilla vastauksesi pääkohdat ja lajittele ne johdonmukaiseen järjestykseen — älä kirjoita yhteen pötköön kaikkea mielen tulevaa, se on varma tapa unohtaa olennaista.
2. Muista vastata kaikkiin tehtävän kysymyksiin, täysiä pisteitä ei voi saada jos kaikkiin kysytyihin asioihin ei ole vastattu.

Palauta kaikki *selkeästi* nimetyt vastauspaperit omiin pinoihinsa!

..... Tehtävät 1. & 2. omalle paperilleen! Nimi paperiin!

1. Selitä (max. 6 riviä/kohta) seuraavat olio-ohjelmoinnin ja C++:n käsitteet ja mitä hyötyä niistä saadaan olio-ohjelmoinnissa. *Älä* selitä niistä pelkkää syntaksia tms, vaan kerro etupäässä, mitä ko. käsitteet *tarkoittavat*.
 - a) Poikkeushierarkia (*exception hierarchy*)
 - b) Funktio-olio (*function object, functor*)
 - c) Matalakopiointi (*shallow copy*)
 - d) Metafunktio (*metafunction*)
 - e) Viipaloituminen (*slicing*)
 - f) Suunnittelumalli (*design pattern*)
2. Periytyminen.
 - a) Sekä periytyminen että templatet (toteutusmallit) antavat mahdollisuuden yleiskäyttöisien, mutta silti muunneltavien komponenttien tekemiseen. Miten tämä onnistuu periytymisellä, miten templateilla? Millaisissa tilanteissa periytyminen on ainoa vaihtoehto? Entä millaisissa tilanteissa templatet?
 - b) Oletetaan, että C++:stä poistettaisiin *virtuaalifunktiot* ja *dynaaminen sitominen*. Mitkä *oliosuunnittelun* asiat muuttuisivat hankalammiksi toteuttaa? Yritä myös keksiä tapoja, joilla virtuaalifunktioiden ja dynaamisen sitomisen puute voitaisiin koodattaessa kiertää.

KÄÄNNÄ!