

OHJ-1400 Olio-ohjelmoinnin peruskurssi

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

Muutama sana tenttivastauksen kirjoittamisesta:

1. Vastauksessa olet vastaavasi sellaisen ihmisen kysymykseen, joka tuntee kohtalaisen hyvin ohjelmistotekniikan aihealuetta muutoin paitsi tämän kysymyksen osalta. Muista että vastauksesi tarkoitus on vakuuttaa tarkastaja osaamisestasi.
2. Mieti etukäteen esim. ranskalaisilla viivoilla vastauksesi pääkohdat ja lajittele ne johdonmukaiseen järjestykseen — älä kirjoita yhteen pötköön kaikkea mieleen tulevaa.
3. Jos vastaus vaatii ohjelmakoodin kirjoittamista, sen ei tarvitse olla pilkulleen syntaksiltaan oikein.

1. Selitä (n. max. 6 riviä/kohta) seuraavat olio-ohjelmoinnin käsitteet ja mitä hyötyä niistä saadaan olio-ohjelmoinnissa. *Älä* selitä niistä pelkkää syntaksia tms. vaan kerro etupäässä, mitä ko. käsitteet *tarkoittavat*.
 - a) CRC-kortti (*CRC card*)
 - b) Ennakkoesittely (*forward declaration*)
 - c) Koostumissuhde (*aggregation*)
 - d) Luokkahierarkia (*class hierarchy*)
 - e) Nimiavaruus (*namespace*)
 - f) Virtuaalifunktio (*virtual function*)
 - g) const-jäsenfunktio
2. Periytyminen on yksi keskeisiä olio-ohjelmoinnin mekanismeja.
 - a) Mistä siinä oikein on kysymys?
 - b) Pohdi, millaisia uusia asioita periytyminen tekee mahdolliseksi?
 - c) Entä mitä haittavaikutuksia periytyminen käyttämisellä voi olla?
3. Seuraavassa on joukko väittämiä olio-ohjelmoinnista ja C++:sta. Mitkä väittämät ovat oikein, mitkä väärin? Perustele mielestäsi vääristä väittämistä parilla lauseella, *miksi/miten* väittämä on väärin ja miten asia todellisuudessa on.
 - a) Vakio-osoittimen (esim. Kirja const*) läpi oliolle saa kutsua vain jäsenfunktioita, jonka parametrin ovat vakio-osoittimia tai -viitteitä.
 - b) Luokan rajapinta on sitä parempi, mitä laajempi se on.
 - c) Jos dynaamisesti new'llä luodun olion jättää tuhoamatta *deletellä*, se ei haittaa koska ohjelman lopussa käyttöjärjestelmä vapauttaa muistin kuitenkin.
 - d) Kapseloinnilla tarkoitetaan olio-ohjelmoinnissa sitä, että olion sisäinen toteutus ei pääse käsiksi muuhun ohjelmaan.
 - e) Luokan vastuualueella tarkoitetaan sitä osaa ohjelmasta, jonka toiminnallisuuden toteuttamisesta luokka vastaa.
 - f) Olion tilalla tarkoitetaan olion sisältämää informaatiota.
 - g) UML:ssä periytymisnuolen yhteydessä oleva lukumäärämerkintä ilmoittaa, montako aliluokkaa kantaluokasta on periytetty.
 - h) Jos olio jäsenmuuttujina on toisia oliolta, nämä jäsenmuuttujaoliot tuhoetaan ja niiden purkaja suoritetaan, vaikka isäntäoliolle ei olisikaan kirjoitettu purkajaa.

..... *KÄÄNNÄ!*

4. Alla on luokkakaavio, jossa on neljä luokkaa ja niiden välisiä yhteyksiä.
- Kirjoita kaavion perusteella luokkien mahdolliset esittelyt (siis "class X {...}";, ei jäsenfunktioiden toteutusta). Perustele lyhyesti esittelyihin, miksi olet päätenyt mihinkin ratkaisuun.
 - Salliiko luokkakaavio sellaisen tilanteen, että teoskokoelma koostuisi useista toisista teoskokoelmista? Perustele vastauksesi.
 - Riippuen siitä, mitä vastasit edelliseen kohtaan, vastaa seuraavista toiseen. Piirrä vastauspaperiin uusi luokkakaavio.
 - Jos mielestäsi luokkakaavio sallii tilanteen, että teoskokoelma koostuisi useista toisista teoskokoelmista, niin muuta luokkakaaviota niin, että teoskokoelma *ei* voi koostua toisista teoskokoelmista, vain yksittäisistä teoksista.
 - Jos taas mielestäsi kaavio ei salli teoskokoelman koostumista toisista teoskokoelmista, muuta kaaviota niin että tuollainen koostuminen on mahdollista.

