

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 esittämään kysymykseen, joka tuntee kohtalaisen hyvin ohjelmistotekniikan aihealuetta muutoin paitsi tämän kysymyksen osalta. Muista että vastauksesi tarkoitus on vakuuttaa tarkastaja siitä, että osaat.
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.

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

1. Seuraavassa on joukko väittämiä olio-ohjelmoinnista ja C++:sta. Mitkä väittämät ovat oikein, mitkä väärin? Perustele mielestäsi vääristä väittämistä parilla lauseella, *miksi/miten* väittäjä on väärin ja miten asia todellisuudessa on. (6 p)
 - a) Jos kaksi oliota kuuluu samaan luokkaan, C++:ssa olio pääsee käsiksi toisen olion private-osaan.
 - b) Periytymisen yhteydessä aliluokka perii kantaluokalta kaiken muun toteutuksen paitsi kantaluokan private-osan.
 - c) Luokan vastuualueella tarkoitetaan sitä osaa ohjelmasta, joka on vastuussa luokan olioiden luomisesta ja tuhoamisesta.
 - d) Dynaaminen sitominen (*dynamic binding*) tarkoittaa sitä, että jäsenfunktion kutsussa kutsuttava koodi valitaan vasta ajonaikana.
 - e) Tapahtumasekvenssit ovat UML:n notaatio, jolla erotellaan toisistaan ohjelman mahdolliset ja mahdottomat tapahtumat.
 - f) Jos dynaamisesti new'llä luodun olion jättää tuhoamatta *deletellä*, se ei haittaa koska ohjelman lopussa käyttöjärjestelmä vapauttaa muistin kuitenkin.
2. Vastaa seuraaviin kysymyksiin. (Keskity olennaiseen, liika jaarittelukin vähentää vastauksen arvoa.)
 - a) const-sanan merkitys ja käyttötavat C++:lla *olio-ohjelmointia* tehtäessä. (3 p)
 - b) Periytyminen olio-ohjelmoinnissa. Mitä se tarkoittaa? Mihin eri tarkoituksiin sitä voi käyttää? Mitä hyötyä siitä on, ts. miten se helpottaa ohjelmointia? *Älä* tuhlaa aikaasi C++:n syntaksin ja erikoisuuksien selittelyyn vaan keskity periytymiseen olio-ohjelmoinnissa ja -suunnittelussa yleensä. (6 p)

..... Tehtävät 3. & 4. **omalle paperilleen!** Nimi paperiin!

3. Rajapinnan (esim. luokan) suunnitteluun liittyy (ainakin) kaksi näkökulmaa — rajapinnan käyttäjä ja sen toteuttaja. Luettele tärkeitä rajapinnan suunnitteluun liittyviä asioita ja pohdi *lyhyesti* kunkin vaikutusta sekä rajapinnan käyttäjän että toteuttajan kannalta. (6 p)

KÄÄNNÄ!

4. Alla on osa TTY:n kursseja ja niiden suorituksia kuvaavaa luokkakaaviota (josta on yksinkertaisuuden vuoksi jätetty monet rajapinnat pois). Vastaa sen perusteella seuraaviin kysymyksiin:
- Mitä tarkoittaa Kurssisuorituksen ja Osasuorituksen välinen "avoin salmiakki"? (1 p)
 - Mahdollistaako kaavio sen, että jonkin kurssin opiskelija voisi toimia myös luennoijana? Perustele vastauksesi. (2 p)
 - Kopioi kaavio vastauspaperille ja lisää luokkien välisiin suhteisiin lukumäärämerkinnät ja tarvittaessa suunnat. Perustele lyhyesti valintasi. (2 p)
 - Mikä merkitys on merkinnöillä "{abstract}"? (1 p)
 - Onko kaaviossa virheitä tai jotain kyseenalaista? (1 p)

