

OHJ-1400 Olio-ohjelmoinnin peruskurssi

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

1. Oleta 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 vastauksesi pääkohdat ja lajittele ne johdonmukaiseen järjestykseen — älä kirjoita yhteen pötköön kaikkea mieleen tulevaa.

..... 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 max. 6 rivillä, *miksi/miten* väittäjä on väärin ja miten asia todellisuudessa on. (6 p)
 - a) Periytymisen yhteydessä aliluokka perii kantaluokalta kaiken muun toteutuksen paitsi kantaluokan *private*-osan.
 - b) Periytymisen "aliluokan olio kuuluu myös kantaluokkaan" tarkoittaa, että aliluokan olioiden toiminnallisuus on sama kuin kantaluokan olioiden.
 - c) UML:ssä periytymisnuolen yhteydessä oleva lukumäärämerkintä ilmoittaa, montako aliluokkaa kantaluokasta on periytetty.
 - d) Dynaaminen sitominen (*dynamic binding*) tarkoittaa sitä, että jäsenfunktion kutsun yhteydessä kutsuttava koodi valitaan vasta ajon aikana.
 - e) Määreellä *protected* merkitään luokassa ne jäsenfunktiot ja -muuttujat, joiden halutaan näkyvän aliluokille muttei muualle.
 - f) Luokkahierarkioiden tarkoituksena on esittää, mitkä luokat käyttävät minkäkin toisen luokan palveluita.
2. Millaisia tapoja C++:ssa on välttää muistivuotoja ja muita dynaamiseen muistinhallintaan liittyviä vaaroja? Mitkä näistä tavoista ovat olio-ohjelmoinnin "ansiota" ja miksi? Entä tuovatko oliot C++:ssa uusia vaaroja dynaamiseen muistinhallintaan? (6 p)

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

3. Vastaa lyhyillä (mutta täsmällisillä) lauseilla ja ranskalaisilla viivoilla seuraaviin:
 - a) "Normaalisti" luokan jäsenfunktiota voi kutsua minkä tahansa luokan olion kautta. Millaisia tapoja C++:ssa luokan kirjoittajalla on rajoittaa sitä, millaisille olioille jäsenfunktiota saa kutsua? (4 p)
 - b) Mitä vaihtoehtoja C++:ssa on sen suhteen, missä luokan rajapinnan määrittelemä jäsenfunktio on toteutettu (tässä *ei* tarkoiteta kooditiedoston nimeä tms.)? (3 p)
4. Vastaa toisella paperilla olevan ohjelmalistauksen perusteella alla oleviin tehtäviin:
 - a) Piirrä kuva (vapaamuotoinen), josta käy ilmi, mitä olioita ja muuttujia ohjelmassa on olemassa rivin 50 suorittamisen jälkeen. Kuvasta tulee selvitä myös muuttujien ja olioiden jäsenmuuttujien arvot. Osoittimista tulee selvitä, mihin olioon ne osoittavat. Olioiden nimien ja tyyppien (=luokkien) tulee myös selvitä kuvasta (nimi vain niistä olioista, joilla on nimi). (5 p)
 - b) Ohjelmassa on vikaa (muistivuoto/-vuotoja). Mitä lisäyksiä ohjelmaan täytyy tehdä (ja mihin kohtaan), jotta ne korjaantuvat? (Koodia ei ole pakko kirjoittaa, mutta toki saa) (2 p)
 - c) Ohjelmassa on myös muutama muu "hyvän C++-olio-ohjelmointitavan" vastainen asia. Millaista huomautettavaa löydät? (2 p)

..... *Käännä paperi!*