

## *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. Selitä (max. 6 riviä/kohta) seuraavat käsitteet ja mitä hyötyä niistä saadaan olio-ohjelmoinnissa.

- |                                                           |                                                                        |
|-----------------------------------------------------------|------------------------------------------------------------------------|
| a) Nimiavaruus ( <i>namespace</i> )                       | d) Lokalisuusperiaate                                                  |
| b) Tapahtumasekvenssi(kaavio) ( <i>sequence diagram</i> ) | e) Näkyvyysmääre <i>protected</i>                                      |
| c) Rajapintaluokka ( <i>interface class</i> )             | f) Vakio-osoittimet ja -viitteet ( <i>X const * ja X const &amp;</i> ) |

2. Olio-ohjelmointi, olioiden elinkaari ja C++:n purkajat.

- a) Luettele eri tilanteita, joissa C++:ssa oliion elinkaari loppuu ja sen purkajaa kutsutaan. Yritä keksiä mahdollisimman monta *erilaista* tilannetta C++-kielen kannalta, ei vain sama tilanne eri olosuhteissa!
- b) Mitä hyötyä C++:n purkajista on dynaamisessa muistinhallinnassa? Entä tuovatko ne uusia vaaroja dynaamiseen muistinhallintaan?
- c) Monissa muissa oliokielistä ei C++:n purkajia vastaavaa rakennetta ole, ja automaattinen roskienkeruu vapauttaa varatun muistin. Pohdi, miten olioiden elinkaaren hallinta noissa kielissä eroaa C++:sta. Yritä löytää sekä plussia että miinuksia!

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

3. Vastaa lyhyehkösti (mutta kattavasti) esim. ranskalaisilla viivoilla ja lyhyillä lauseilla seuraaviin kysymyksiin.

- a) Millaisia mahdollisuuksia yleiskäyttöiseen koodiin *periytyminen* antaa olio-ohjelmoinnissa?
- b) Mitä tarkoitetaan *dynaamisella sitomisella* ja mitä siihen liittyy?
- c) Mitä ovat *abstraktit kantaluokat*? Mitä hyötyä niistä on tavallisiin luokkiin verrattuna?

4. Piirrä seuraavan kuvauksen perusteella "huonekalujen hinnoittelujärjestelmään" liittyvä luokkakaavio UML-notaatiolla. Pyri tekemään kaaviosta mahdollisimman yksityiskohtainen (tehtävänannon puitteissa). Merkitse kaavioon selkeästi myös, mihin alla olevaan numeroituun kohtaan mikäkin kaavion osa liittyy. Kommentoi myös, jos joku lueteltu asia ei näy luokkakaaviosta tai jos mielestäsi on useita vaihtoehtoisia tapoja tehdä asia.

- a) Järjestelmässä on tällä hetkellä kahdenlaisia huonekaluja, pöytiä ja tuoleja.
- b) Muitakin huonekalutyyppejä voidaan koodata mukaan myöhemmin.
- c) Kullakin huonekalulla on materiaali, joka on joko pyökki, tammi tai saarni.
- d) Lisäksi huonekaluksi lasketaan yhtenä kokonaisuutena myytävä "pöytäryhmä", jolla tarkoitetaan yhtä pöytää sekä vähintään kahta tuolia.
- e) Jokaiselle huonekalulle voidaan laskea sen hinta. Pöydän hinta on  $10 \times$  *pöydän paino*, tuolin hinta on  $4 \times$  *jalkojen lukumäärä* ja pöytäryhmän hinta 90 % sen osasten hintojen summasta.
- f) Järjestelmään kuuluu myös huonekaluluettelo, josta löytyvät kaikki myynnissä olevat huonekalut.