

Tehtävä oli sama kuin 22.6.2000 pidetyssä kokeessa.

Tehtävästä oli mahdollista saada maksimissaan 14 pistettä. Täydet pisteet oli mahdollista saada jopa pelkillä faktoilla. Lisäksi jaossa oli plussapisteitä selkeästä esityksestä ja havaintoesimerkeistä.

Tehtävän aihepiiri käsitteli luokkien lataamista muistiin ja luokan ilmentymien luomista. Tämän lisäksi haettiin selkeää esitystä luokka- ja ilmentymäkohtaisuuden eroista.

Faktoista jaetut pisteet:

Jokaisesta palluralla listatusta kohdasta oli mahdollista poimia itselleen piste. Virheellisistä lauselmista ei vähennetty pisteitä. Mikäli samassa vastauksessa oli esitetty sama asia oikein ja virheellisesti, piste jäi luonnollisesti saamatta. Hieman vääriä termivalintojakin hyväksyttiin, mikäli vastauksesta näki että asia on ymmärretty.

Luokkien lataaminen:

- Luokkamuuttujat alustetaan ja niille annetaan alkuarvot luokan lataamisen yhteydessä.
- Luokan lataamisen yhteydessä suoritetaan staattiset alustuslohkot.
- Luokka ladataan kun siihen viitataan ensimmäistä kertaa (siis ennen kuin ensimmäistäkään ilmentymää on luotu).
- Ainoastaan tarvittavat luokat ladataan.
- Vain kerran ohjelman suorituksen aikana.
- Kun luokka on ladattu, käytössä vain staattinen kalusto.

Ilmentymän luominen

- Ilmentymämuuttujat saavat alkuarvonsa.
- Suoritetaan konstruktori.
- Ilmentymä luodaan käyttämällä *new* operaatiota.
- Kun ilmentymä on luotu, käytössä on sekä staattinen kalusto että ei-staattinen kalusto.

Mitä luokan (static) attribuutiksi?

- Jokin kuvaava esimerkki.
- Selvitys miksi. Maininta siitä, että on kaikille olioille yhteinen riitti.

Mitä ilmentymän attribuutiksi?

- Jokin kuvaava esimerkki.

- Selvitys miksi. Maininta siitä, että jokaisella ilmentymällä on oma versio kentästä riitti.

Pelkästään mainitsemalla, että muuttujia on staattisia ja ei staattisia (esim. private int) ei herunut pisteen pistettä. Vastauksesta ei tule millään tavoin esille, tietääkö kirjoittaja mitä niillä tarkoitetaan.

Esimerkeistä ja esityksestä jaetut pisteet:

Plussapisteitä jaettiin selkeästä esityksestä ja kuvaavista esimerkeistä. Selkeistä koodiesimerkeistä oli mahdollista poimia pisteitä myös edellisen pallolistan kohtiin. Koodiesimerkistä saattoi esimerkiksi selvittää, että kirjoittaja tietää ilmentymän syntyvän *new* operaatiolla, tai esimerkissä saattoi olla hyvät esimerkit siitä, mikä laitettaisiin luokan ja mikä ilmentymän muuttujaksi. Pisteitä ei saanut, mikäli selitettiin selkeästi, mutta väärää asiaa, esimerkiksi olio-ohjelmoinnin hyötyjä.

- Esimerkki siitä, mitä on mahdollista käyttää missäkin vaiheessa. Esimerkiksi pieni ohjelma (luokka) jossa sekä luokan, että ilmentymän kalustoa ja vaikkapa main -metodi, johon kommentoitu milloin luokka latautuu muistiin ja milloin mitäkin käytössä, missä luodaan ilmentymä jne.
- Esimerkki staticin ja ei-staticin eroista. Usein käytetty esimerkki oli jonotuskone: *Pankissa voi olla luokan Jonotuskone ilmentymiä, jotka jakavat numeroita yhteisestä (static) numero -muuttujasta. Lisäksi jokainen ilmentymä laskee asiakkaiden määrää omalta kohdaltaan. Lukumäärää ylläpidetään ilmentymämuuttujassa.*
- Ajatellun ystävän huomioon ottaminen, eli esityksen havainnollisuus. Pisteitä sai, mikäli oli selvittänyt suhteellisen kattavasti ja tarpeeksi havainnollisesti edellä mainittuja asioita.