

t1)

- a) Ei totta. Ei ole tarkasti määritelty. Määrittelyjä löytyy useita erilaisia. (<http://wiki.c2.com/?DefinitionsForOo>)
- b) Ei totta. Isonkin tietojärjestelmän koodaaminen on *mahdollista* ilman mutatoituvia jäsenmuuttujia. Mutta onko se järkevää? Uuden olion konstruoiminen voi olla työlästä aina sen tilan muuttuessa. Toisaalta Javan ronskienkeruu hoitaa käyttämättä jäävät oliot pois muistista.
- c) Totta. OOP-kielissä reaali maailman asioita on helppo mallintaa luokilla.
- d) ?

t2)

- a) Kielimäärittely: Object on kiinteä osa Javan arkkitehtuurin toteutusta
- b) Staattisen metodin kutsu sidotaan käännösvaiheessa.
- c) Ajon aikana, metodi voidaan ylikirjoittaa Esim-luokan toteuttavassa luokassa.
- d) Suoritus aikana. Reflektion avulla kutsuttua metodia ei käännösaikana käännetä, vaan se haetaan dynaamisesti.
- e) Suoritusaikana? Riippuen JVM-versiosta olion koko voi viedä eri määrät tilaa (esim. JVM 32- ja 64-bittiset toteutukset).
- f) Suoritusaikana? Rekursiivinen metodi kuluttaa muistia ja päättyy siis silloin kun saatavilla oleva muisti loppuu.
- g) Goto-määreet ovat määriteltynä tavukoodissa.
- h) final-määreiset jäsenet sidotaan käännösaikana. Niitä ei voi muuttaa ajon aikana.
- i) final-määreiset jäsenet sidotaan käännösaikana. Niitä ei voi muuttaa ajon aikana.

t3)

a ja b koodiesimerkki

- c) Kieli ei voi tehdä mitään. Käyttäjän pitää tarkistaa onko ihmisellä näköä / Napata poikkeus, jos ei ole näköä.
- d) Ohjelmoijan täytyy varautua siihen, ettei oliolla ole näköä.