

Tentin kesto 3h

Ei oheismateriaalia

1. Kuvaa lyhyesti tietovuoarkkitehtuurin (pipes-and-filters) perusidea. Millaisiin järjestelmiin se sopii? Mitä etuja ja ongelmia liittyy tietovuoarkkitehtuuriin? Anna esimerkki järjestelmästä, jossa tietovuoarkkitehtuuria on sovellettu. Käytä UML:n komponenttikaaviota (mahdollisine lisäselityksineen) järjestelmän kuvauksessa.

(6p)

2. Oletetaan, että eräessä sovelluksessa tarvitaan komponentti C, joka tarjoaa palvelun p. Arkkitehti huomaa kuitenkin, että komponenttia C voitaisiin käyttää muissakin yrityksen tekemissä sovelluksissa, jos eräs sen tarvitsema palvelu q voitaisiin antaa sovelluskohtaisena toteutuksena. Niinpä arkkitehti päättää liittää komponentin C yrityksen yleiskäyttöiseen kirjastoon.

a) Anna UML-komponenttikaavio, joka kuvaa tähän tarvittavan ohjelmistoratkaisun tapauksessa, missä sovelluskohtaiset komponentit A ja B käyttävät C:tä (komponentit + rajapinnat).

b) Kun järjestelmä on tehty, jonkin ajan kuluttua käy ilmi, että komponentti C joudutaan vaihtamaan toiseksi komponentiksi C', joka tarjoaa loogisesti samat palvelut mutta erilaisella rajapinnalla p', ja C':n tarvitsemat sovelluskohtaiset palvelut pyydetään rajapinnan q' kautta. Anna uusi arkkitehtuuriratkaisu UML-komponenttikaaviolla siten, että komponentteja A ja B ei tarvitse muuttaa.

(6p)

3. Suunnittele hajautettua ohjelmistokehitystä tukevan UML-työkalun arkkitehtuuri. Työkalun on sallittava yhteisten UML-mallien luominen ja editointi eri työasemissa. Työkalussa on poliittisista syistä käytettävä alustana Eclipseä ja GEFiä. Mallin työstäminen ei saa olla olennaisesti hitaampaa kuin yhden käyttäjän työkalussa. Mallin eri osia (pakkauksia) on voitava käsitellä samanaikaisesti, kuitenkin riittää, että samaa pakkausta käsittelee kerrallaan korkeintaan yksi käyttäjä. Tietokantana on käytettävä kaupallista oliokantaa nimeltä ODB. Arkkitehtuurin on tuettava kannan vaihtamista toiseksi (esim. yhden käyttäjän kannaksi). Anna arkkitehtuurin kuvaus kaaviona, joka kuvaa järjestelmän fyysisen näkymän siten, että kuvasta käy ilmi eri ohjelmiston osat ja niiden riippuvuudet. Luettele myös kunkin komponentin tai osajärjestelmän päävastuut järjestelmässä. Mitä arkkitehtuurityylejä sovellat järjestelmässä?

(6p)

4. Millainen on hierarkkinen ohjelmistokehitys, ja mitä etuja siitä on? Anna esimerkki sovellettuna pankkijärjestelmiin.

(6p)

5. Mitä ovat ATAM-menetelmässä a) riskit (risk), b) herkkyyskohdat (sensitivity point) ja c) tasapainokohdat (tradeoff point)?

(6p)