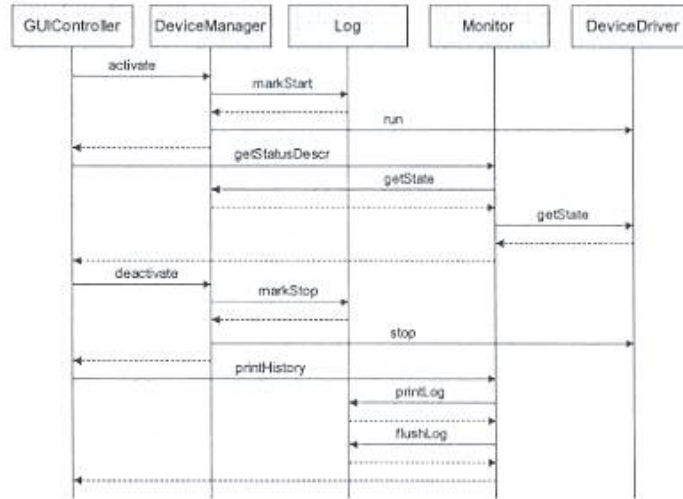


- Seuraavassa sekvenssikaaviossa on kuvattu tyypillinen vuorovaikutus komponenttien välillä eräässä sovelluksessa. Johda sekvenssikaaviosta mahdolliset roolirajapinnat komponenteille ja kuvaa komponenttien ja rajapintojen suhteet UML:n komponenttikaaviona tarjottuine ja vaadittuine rajapintoineen. Rajapintoja ei tarvitse nimetä, mutta merkitse niihin näkyviin operaatiot.

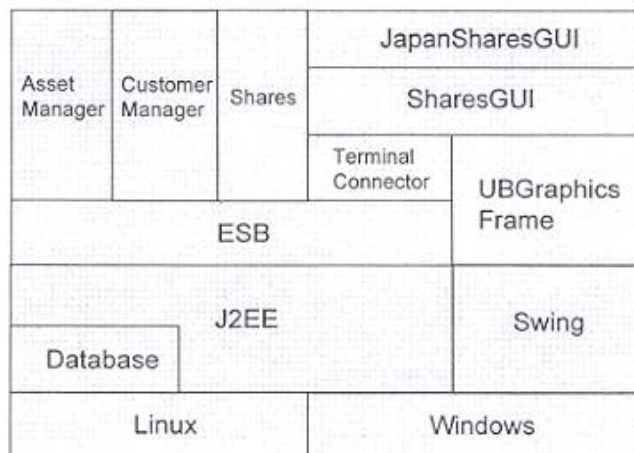


(6p)

- Kuvaa viestinvälitysarkkitehtuurityyli (message dispatcher) a) antamalla kuva ja siihen liitetty lyhyt selitys, b) luettelemalla tyylin edut ja c) luettelemalla tyylin ongelmat.

(6p)

- Seuraavassa on esitetty erään Japanin markkinoille tarkoitetun osakekaupan hallintaa tukevan järjestelmän kerrosarkkitehtuuri:



KÄÄNNÄ

Järjestelmä käyttää Linux-palvelimia, jotka ylläpitävät mm. asiakas- ja arvopaperirekistereita. Järjestelmä perustuu J2EE:hen, jonka tarjoaman viestinvälitysmekanismiin (JMS) päälle on rakennettu korkean tason tuki erilaisten liiketoimintapalvelujen viestipohjaiselle kommunikaatiolle (ESB = Enterprise Service Bus). Tätä käyttävät hyväkseen mm. asiakkaiden- ja arvopaperienhallintakomponentit (AssetManager, CustomerManager). Varsinainen osakemarkkinoiden logiikasta huolehtiva osa järjestelmää on Shares, joka puolestaan käyttää ESB:n kautta Manager-komponenttien palveluja. Järjestelmän terminaalit ovat Windows-pohjaisia, ja ne käyttävät TerminalConnector-komponentin kautta ESB:n viestinvälityspalveluja. Terminaaleissa oleva käyttöliittymä perustuu Swingiin (Javan yleinen korkean tason GUI-kehys), jonka päälle on rakennettu ko. pankin oma GUI-kehys (UBGraphicsFrame). Tämän päällä toimii osakesovelluksen oma käyttöliittymä (SharesGUI), joka edelleen erikoistetaan tietyille kielelle/maalle (tässä japanille). Järjestelmällä voidaan esimerkiksi antaa pankille osakkeiden myyntimääräys siten, että järjestelmä ensin tarkistaa käyttöoikeudet, kysyy sitten halutun yhtiön, osakemäärän ja rajahinnan, ja myynnin tapahduttua lähettää ilmoituksen asiakkaalle.

Merkitse kerrosarkkitehtuuriin näkyviin tuoterunkoarkkitehtuurien nelikerrosmallin mukaiset kerrokset: mitkä osat kuuluvat varsinaiseen sovellukseen, mitkä sovellusalustaan (application platform), mitkä arkkitehtuurialustaan (architecture platform), mitkä resurssialustaan (resource platform). Voit joko kopioida yllä olevan kuvan ja merkitä nelikerrosmallin mukaiset osat siihen, tai voit merkitä nämä osat suoraan tehtäväpaperiin.

(6p)

Oletetaan, että suunnitellaan kehystä, jonka pohjalle tullaan rakentamaan erilaisia graafisia editoreja. Nimeä kolme suunnittelumallia, joita todennäköisesti sovellettaisiin kehyksessä. Anna kullekin suunnittelumallille jokin kehyksen osa (luokkakaaviona), jossa suunnittelumallia on sovellettu. Anna näissä suunnittelumallien ilmentymissä esiintyville luokille ja operaatioille kehyksen käsitteisiin sopivat nimet, ja merkitse mitä osaa (roolia) ne edustavat suunnittelumallissa. Selitä myös lyhyesti, minkä ongelman ne ratkaisevat kehyksessä.

(6p)

Tarkastellaan tehtävän 3 järjestelmää. Lue uudestaan tehtävän 3 kuvaus.

- a) Anna järjestelmän fyysinen näkymä siihen parhaiten sopivalla UML:n kaaviolla.
- b) Anna järjestelmän looginen näkymä siihen parhaiten sopivalla UML:n kaaviolla.
- c) Anna järjestelmän skenaarionäkymä siihen parhaiten sopivalla UML:n kaaviolla.