

# OHJ-3200 Ohjelmistoarkkitehtuurit Tentti 18.4.2012

Tentissä ei saa käyttää ylimääräistä kirjallista materiaalia, laskimia, tietokoneita tai muita lunttausvälineitä.

Tentin tarkistaa Johannes Koskinen.

---

Jokaisesta tehtävästä saa max. 6 pistettä.

## 1. Tehtävä

Kassapäätejärjestelmä palvelee kaupan kassoja. Järjestelmä tarjoaa mm. seuraavat toiminnallisuudet:

- kortinluku (pankki, luotto, bonus)
- myyntikirjanpito
- palautusten käsittely
- myyntiraporttien tulostus
- käyttäjän tunnistus
- käyttäjien hallinnointi.

Kaikki liiketoimintaan liittyvä tieto on talletettu tietokantaan.

Anna järjestelmän kerrosarkkitehtuuri. Sijoita kerroksiin seuraavat komponentit tai alijärjestelmät: **AWT** (Javan yleinen grafiikkatuki), **DBA** (tietokannan abstraktio), **DB** (tietokanta), **SaleLogic** (yleinen kassapääteologiikka), **CardReader** (kortinlukijan abstraktio), **CRD** (kortinlukijan ajuri), **BarCode** (viivakoodilukijan abstraktio), **RetailerLogic** (myymäläketjukohtainen logiikka), **SaleGraphics** (yleinen graafinen tuki kassapäätesovelluksille), **RetailerGUI** (myymäläketjukohtainen GUI), **Swing** (Javan GUI-kirjasto) ja **BCSD** (viivakoodilukijan ajuri).

## 2. Tehtävä

Kirjoita lyhyt (n. 1-2 käsikirjoitussivua) essee aiheesta "Ketterä arkkitehtuurin arviointi". Rakenna essee seuraavan jaotelmun mukaan (kustakin kohdasta noin 1 kappale tekstiä):

- Arkkitehtuuriarvioinnin tarpeellisuus ohjelmistotuotannossa
- DCARin yleinen luonne ja tarkoitus
- Miten DCAR-prosessi eroaa ATAM-prosessista?
- Mitä ongelmia DCARin käytöstä voisi olla ATAMiin verrattuna?

## 3. Tehtävä

Selitä lyhyesti, mitä ongelmia liittyy tuoterunkojen käyttöön?

## 4. Tehtävä

Kännyköille tehdään pelejä, joissa pelaajan lähettämä hahmo liikkuu erilaisia esineitä sisältävässä kaksiulotteisessa maailmassa. Osa esineistä hajoaa pelaajan hahmon osuessa siihen, osa aiheuttaa pelaajan hahmon liikesuunnan muutoksen ja osa kerryttää pelaajan pistesaalista. Pelaaja saa läpi pelin, kun hän on saanut riittävän monta pistettä. Pelin alussa luodaan maailma luomalla sinne ennalta määrätty joukko esineitä ja pelaajan hahmon lähtöpaikka. Voit vapaasti tarkentaa pelin kuvausta tarpeen mukaan.

Yritys päättää tehdä tuotealustan tukemaan tämän tyyppisten pelien kehittämistä, ja tuotealusta päätetään rakentaa sovelluskehityksenä. Suunnittele sovelluskehityksen arkkitehtuuri yllä annetun kuvauksen perusteella. Tässä riittää suunnitella ainoastaan pelilogiikasta huolehtiva kehys; oletetaan, että käyttöliittymä on yhdistetty ohjaimen (Controller), joka muuntaa komennot kehityksen toiminnoiksi.

Esitä arkkitehtuuri luokka- tai komponenttikaaviona tärkeimpine operaatioineen ja tarpeellisine selityksineen. Lisää kaavioon myös "KiltitSiat"-esimerkkisovellus, jossa pelaaja lähettää singolla sikoja ja pelaajan on tarkoitus osua lintuihin. Riittää, että määritellään pelaajan lähettämän hahmon vuorovaikutus lintujen kanssa tietyllä tavalla. Merkitse kaavioon sovelluskohtaiset osat.

## 5. Tehtävä

Alla on kuvattu esimerkkimalli, joka käyttää pientä UML:n luokkakaavion osajoukkoa. Anna tämän osajoukon meta-malli UML:n luokkakaaviona. Varmista, että metamalli ilmaisee mm. seuraavat asiat:

KAÄNNÄ 

- a) kullakin luokan attribuutilla on tyyppinä jokin luokka
- b) kullakin luokalla on korkeintaan yksi ylliluokka
- c) luokalla voi olla mielivaltainen määrä aliluokkia

