

# Ohjelmistoarkkitehtuurit

## Loppukoe 14.01.2008

Kaikista tehtävistä voi saada max 6 pistettä. Ei oheismateriaalia.

1. Oletetaan, että halutaan suunnitella yleinen ohjelmistoalusta (tuoterunko), jonka päälle voidaan helposti rakentaa erilaisia vuokrausjärjestelmiä (autonvuokraus, videovuokraus, konevuokraus, ...). Vuokrausjärjestelmään ajatellaan kuuluvan aina mm. käsitteet "asiakas", "vuokrauskohde", "vuokratoimipiste", "varasto". Nämä toteutetaan kussakin vuokrausjärjestelmässä komponenteilla, jotka noudattavat samoja rajapintoja. Alustan on annettava tukea uusien käsiteilmentymien luomiselle ja käsittelemiselle ajoaikana (esim. uuden videon luomiselle ja vuokraamiselle käyttöliittymästä annettujen tietojen perusteella). Esitä, miten Abstrakti tehdas – mallia voidaan soveltaa tässä ohjelmistoalustassa takaamaan, että alusta voi luoda käsiteilmentymiä yhtenäisellä tavalla. Kuvaa Abstrakti tehdas –suunnittelumalli ensin yleisesti luokkakaaviona, ja anna sitten sen sovellus tämän järjestelmän tapauksessa luokkakaaviona (+ lyhyt selitys).
2. Selitä MVC (Model-View-Controller) arkkitehtuurityyli. Kuvaa a) tyylin motivointi: minkä ongelman tyyli ratkaisee, b) perusidea sopivan kaavion avulla, c) mitä osia arkkitehtuuriin sisältyy ja mitä vastuita niillä on, ja d) mitä potentiaalisia ongelmia tyyliin liittyy.
3. Henkivakuutusten hallintajärjestelmä on tarkoitettu asiakaspisteessä toimivalle virkailijalle. Se sisältää mm. seuraavat toiminnallisuudet: asiakas- ja vakuutustietojen ylläpito, asiakaskorttien tunnistus, vakuutusmaksujen ennakkolaskenta, asiakkaiden ja vakuutusten tilastointiraportit, käyttäjien hallinnointi. Kaikki vakuutustoimintaan liittyvä tieto on talletettu tietokantaan. Anna järjestelmän kerrosarkkitehtuuri. Sijoita kerroksiin seuraavat komponentit tai alijärjestelmät:  
AWT (Javan yleinen grafiikkatuki), InsBase (tietokannan vakuutusabstraktiokerros), DB (tietokanta), DBMS (tietokannan hallintajärjestelmä), InsLogic (yleinen tuki vakuutussovelluksille), InsCard (kortinlukijan abstraktio), CardReader (kortinlukijan ajuri), LifeInsLogic (henkivakuutusten logiikkaosa), InsGraphics (yleinen graafinen tuki vakuutussovelluksille), LifeInsGUI (järjestelmän GUI), Swing (Javan GUI-kirjasto), InsReporter (yleinen tuki vakuutustietojen raportoinnille), ja LifeReporter (henkivakuutusten raportointikomponentti) ja Linux (käyttöjärjestelmä).
4. Javaa muistuttavalle skriptikielelle tehdään tulkintaan perustuva toteutus, jossa tulkki suorittaa koodin olioesitystä. Kielessä on mm. seuraavat rakenteet:

```
Program ::= BlockStatement
Statement ::= AssignStatement | IfStatement | BlockStatement
AssignStatement ::= identifier "=" Expression
Expression ::= Expression Op Expression | Primary
Op ::= "+" | "-" | "*" | "/"
Primary ::= integer | float | identifier
IfStatement ::= "IF" "{" Expression "}" Statement
BlockStatement ::= "{" StatementList "}"
StatementList ::= Statement {";" Statement}
```

KÄÄNNÄ

Tässä metakielen aaltosulut tarkoittavat toistoa  $n$  kertaa ( $n=0,1, \dots$ ) ja pystyviiva tarkoittaa vaihtoehtoista rakennetta. Kokonaislukuja, liukulukuja ja tunnuksia tarkoittavat tekstialkiot (integer, float, identifier) on jätetty tässä tarkentamatta.

a) Anna kielen olioesityksen rakennemalli UML:n luokkakaaviona. (4p)

b) Esiintyykö ratkaisussasi Tulkki-suunnittelumallin ilmentymiä jossain muodossa? Merkitse mahdolliset ilmentymät luokkakaavioon. (2p)

5. a) Kuvaa lyhyesti ATAM-menetelmän perusajatus. (3p)

b) Tarkastellaan matkojen hallintaan tarkoitettua järjestelmää, jonka avulla työntekijät tekevät työmatkaesitykset ja matkalaskut. Järjestelmä huolehtii esitysten ja laskujen tarkistuttamisesta ja hyväksyttämistä asianomaisilla vastuuhenkilöillä, ja laskujen maksattamisesta henkilön tilille. Järjestelmän tilaaja vaatii, että järjestelmän on oltava hyvin käytettävä, suorituskykyinen ja muunneltava. Tee ATAMin laatupuu näille laatuominaisuuksille tarkennuksineen (2 kullekin päälaatuominaisuudelle) ja skenaarioineen. Voit tehdä tarpeen mukaan lisäoletuksia järjestelmästä. (3p)