

DHJ-3200
Ohjelmistoarkkitehtuurit
Loppukoe 15.3.2010

Tehtävistä 1-5 saa max 6 pistettä, tehtävästä 6 max 4 pistettä (yhteensä kuitenkin max 30 pistettä).
Ei oheiskirjallisuutta. Ei laskinta.

Tentin laatija: Kai Koskimies

1. Anna Strategia –suunnittelumallin kuvaus sisältäen kohdat nimi, konteksti, ongelma, ratkaisu (teksti+kuva UML kaaviona), seuraukset.
2. Selitä lyhyesti, mitä yleisesti tarkoitetaan särkyvän ylikuorman ongelmalla. Anna esimerkki ongelman esiintymisestä.
3. Kirjoita lyhyt (n. 1 käsikirjoitussivu) essee aiheesta ”Arkkitehtuurin arvioinnin avulla voidaan täydentää arkkitehtuurin dokumentointia”.
4. Kännyköille tehdään pelejä, joissa pelaajan kuvake liikkuu erilaisia esteitä sisältävässä kaksiuolotteisessa maailmassa ja kerää sieltä tavaroita. Pelaajan on vältettävä kosketusta tietynlaisten esteiden kanssa ja ammuttava hirviöt, jotka voivat tuhota hänet. Pelaaja saa läpi pelin, kun hän pääsee hengissä riittävän määrän tavaroita keränneenä maaliin. Pelin alussa luodaan satunnainen joukko esteitä, tavaroita ja hirviöitä. Yritys päättää tehdä tuotealustan tukemaan tämän tyyppisten pelien kehittämistä, ja tuotealusta päätetään rakentaa sovelluskehystenä. Suunnittele sovelluskehysten arkkitehtuuri yllä annetun kuvauksen perusteella. Tässä riittää suunnitella ainoastaan pelilogiikasta huolehtiva kehys; oletetaan, että käyttöliittymä on yhdistetty ohjaimen (Controller), joka muuntaa komennot kehysten toiminnoiksi. Voit vapaasti tarkentaa itse kuvausta tarpeen mukaan. Esitä arkkitehtuuri luokkat tai komponenttikaaviona tärkeimpine operaatioineen ja tarpeellisine selityksineen. Lisää kaavioon myös Lord-esimerkkisovellus, jossa riittää, että määritellään pelaajan vuorovaikutus hirviöiden kanssa tietyllä tavalla, ja merkitse sovelluskohtaiset osat.
5. Oletetaan, että olet tekemässä tulkkiarkkitehtuuriin perustuvaa järjestelmää, jossa toiminta kuvataan järjestelmälle tilakoneena. Tätä varten järjestelmään on suunniteltu tulkattavaksi kieleksi alla annettu tilakoneen kuvauskieli. Yhden tilan kuvaus koostuu joukosta tilasiirtymien kuvauksia. Suunnittele kielen olioesitys UML-luokkakaaviona. Voit vapaasti muuttaa kielioppia olioesityksen helpottamiseksi, mutta et kieltä. Syntyykö ratkaisusi Tulkki-suunnittelumallin ilmentymää? Jos syntyy, merkitse se kaavioon. (Vihje tehtävään: käytä mallina luennolla esitettyä esimerkkiä siten, että muutat kieliopin ensin sellaiseen muotoon, josta on poistettu kaikki terminaalisyönteet, joka noudattaa luennolla esitettyä muotoa, ja jossa käytetään rakenteiden lukumäärän ilmaisemiseen UML-tyyppisiä merkintöjä (1..*), (0..*) ja (0..1). Tämän jälkeen voit suoraviivaisesti muuttaa kieliopin luokkakaavioksi.)

StateMachine → “machine” [InitStateDecl “;”] [FinalStateDecl “;”] State {“;” State} “end”

InitStateDecl → “initially” identifier

FinalStateDecl → “finally” IdList

IdList → identifier {“;” identifier}



State → "state" identifier ":" Transition {"," Transition} "end"

Transition → "when" Event ":" (TargetState | Error)

Event → "finished" | identifier | "timeout" float

TargetState → identifier

Error → "error" string

Tässä hakasulkujen sisällä on valinnainen ilmaus (vois siis puuttua) ja aaltosulkujen sisällä olevaa ilmaus voi esiintyä 0 tai useampi kertaa. Lainausmerkkien sisällä olevat merkkijonot ovat kielen varattuja symboleja.

6. (Jokeritehtävä, max 4 pistettä, liittyy vierailuluentoon)

XMLpss on hajautettu järjestelmä suunnittelutiedon jakamiseksi monitoimijaympäristössä. Anna tämän järjestelmän neljä keskeistä laatuvaatimusta, jotka ovat ohjanneet järjestelmän suunnittelua ja kuvaa lyhyesti (1-2 lauseella) miksi kukin näistä on tärkeä tässä järjestelmässä.