

Ohjelmistojen mallintaminen, koe 16.12.2009

Vastaa tehtävät 1, 2 ja 3 **erillisille** konsepteille. Kirjoita jokaiseen palauttamaasi konseptiin kurssin nimi, kokeen päivämäärä, nimi, nimikirjoitus ja opiskelijanumero.

Vastaukset palautetaan tehtäväkohtaisiin pinoihin. Vaikka jättäisit johonkin tehtävään vastaamatta, tulee vastauspaperi siinäkin tapauksessa palauttaa.

1. (8p) Tässä tehtävässä kiinnitetään huomiota kurssin suuriin linjoihin. Tehtävä on jaettu alakohtiin, mutta voit myös vastata yhden kaikenkattavan esseen avulla. Tässä ei siis haeta mitään yhden sanan vastauksia/ranskalaisia viivoja, vaan järkevästi muotoiltua ja perusteltua tekstiä. Muista kuitenkin keskittyä olennaiseen. Koko tehtävän vastauksen tuskin tarvitsee ylittää kahta A4-sivua.
 - a. (2p) Mistä vaiheista ohjelmistotuotantoprosessi koostuu? Mitä kunkin vaiheen aikana tapahtuu?
 - b. (3p) Mitä tarkoitetaan vesiputousmallilla? Entä iteratiivisella ja ketterällä ohjelmistotuotantomenetelmillä? Mikä johti ketterien ja iteratiivisten menetelmien kehittämiseen?
 - c. (3p) Miten UML liittyy kaikkeen edelliseen?
2. (10p) Luokka- ja sekvenssikaavioita.
 - a. (7p) Mallinna seuraavassa kuvattu tilanne luokkakaaviona. Tässä tehtävässä luokille ei tarvitse määritellä metodeja eikä attribuutteja. Yhteyksien kytkentärajoitteet ja tyypit sekä tarvittaessa myös nimet ja roolinimet on merkittävä tarkasti.

Unix-tyyppisissä järjestelmissä "tiedostoja" ovat sekä normaalit tiedostot, hakemistot että ns. laitetiedostot.

Laitetiedostoon liittyy jokin I/O-laite, esim. hiiri, näppäimistö tai näyttö. I/O-laitetta kohti on tasan yksi laitetiedosto. Normaali tiedosto on jono tavuja. Tietty tavu kuuluu vain yhteen tiedostoon.

Hakemisto koostuu joukosta muita tiedostoja (jotka voivat siis olla normaaleja tiedostoja, hakemistoja tai laitetiedostoja). Tiedosto on kerrallaan vain yhdessä hakemistossa, mutta ei välttämättä koko aikaa samassa hakemistossa¹.

Unixissa on käyttäjiä ja käyttäjäryhmiä. Käyttäjä voi kuulua useampaan ryhmään ja ryhmässä voi olla useita käyttäjiä. Käyttäjät käyttävät I/O-laitteita. Jokaisella tiedostolla on omistajana tasan yksi käyttäjä. Käyttäjä voi omistaa useita tiedostoja. Tiedostoihin voi olla käyttöoikeuksia sekä käyttäjillä että ryhmillä. Sekä käyttäjällä että ryhmällä voi olla käyttöoikeuksia useisiin tiedostoihin.

Oikeasti Unixeissa käyttöoikeudet jakautuvat luku-, kirjoitus- ja suoritusoikeuksiin, mutta tässä tehtävässä emme siitä välitä, vaan mallinamme ainoastaan

¹Oikeasti Unixissa tiedosto voi olla yhtäaikaan useassa hakemistossa, me emme nyt kuitenkaan siitä välitä.

yhden tyyppisen käyttöoikeuden. Eli esim. käyttäjällä joko on tai ei ole käyttöoikeutta tiettyyn tiedostoon. Yhdellä tiedostolla voi olla useita käyttöoikeuden omaavia käyttäjiä ja ryhmiä

- b. (3p) Viimeiseltä sivulta löytyy katkelma Java-koodia. Piirrä sekvenssikaaviona tilanne, jossa kutsutaan (jostain koodin ulkopuolella olevasta metodista) ensin Machine-luokan konstruktoria ja sen jälkeen luodun Machine-olion `drive()`-metodia.

3. (10p) Tarkastellaan harjoitustyön seurantaan liittyvää tietojärjestelmää.

Aloittaessaan harjoitustyön opiskelija saa ohaaajan ja aiheen. Opiskelija tapaa ohjaajana toimivan opettajan kerran viikossa sovittuna aikana. Opiskelija voi valita ajan opettajan tarjolle laittamien aikojen joukosta. Sovittu aika voi vaihdella viikottain. Kussakin tapaamisessa kirjataan työhön siihen mennessä käytetty aika sekä työn etenemisen aste. Opettaja voi lisäksi kirjata kommentteja tapaamiseen liittyen. Kommentit ja kirjatut tiedot ovat vain opettajan luettavissa. Sekä opettaja että opiskelija näkevät järjestelmästä tapaamisaikavarauksensa.

Opiskelija jättää harjoitustyöstä tarkastukseen yhden väliraportin ja loppuraportin, palautus tapahtuu järjestelmän välityksellä. Näistä kummastakin hän saa kirjallisen palautteen ja loppuraportista lisäksi arvosanan. Opiskelija saa palautteen järjestelmän kautta.

- a. (5p) Laadi kuvatusta tietojärjestelmästä karkean tason käyttötapausmalli, eli etsi käyttäjät ja käyttötapaukset nimen tasolla. Määrittele kunkin käyttötapauksen tarkoitus lyhyesti (eli maksimissaan rivi per käyttötapaus) tekstinä. Piirrä myös käyttötapauskaavio.
- b. (2p) Kuva *yksi* käyttötapauksista tarkemmalla tasolla, ns. Cockburnin käyttötapauspohjan tai luentomonisteen tekstuaalisten käyttötapauskuvausten tyyliin.
- c. (3p) Laadi järjestelmän kuvauksen perusteella määrittelytason (eli kohdealueen) luokkakaavio. Metodeja luokille ei tässä vaiheessa laiteta. Yhteyksien tarkempi laatukaan (normaali, kompositio vai kooste ja navigointisuunnat) ei ole vielä tärkeää. Merkkää kuitenkin kaikkiin yhteyksiin kytkentärajoitteet ja nimeä yhteydet ja yhteysroolit tarvittaessa. Ilmeisimmät attribuutit luokille kannattaa merkitä.

```

public class Machine {
    private FuelTank tank;
    private Engine engine;
    public Machine() {
        tank = new FuelTank();
        tank.fill(40);
        engine = new Engine(tank);
    }
    void drive() {
        engine.start();
        boolean running = engine.isRunning();
        if { running } engine.useEnergy();
    }
}

class FuelTank {
    private int fuelContents;
    FuelTank() {
        fuelContents = 0;
    }
    void fill(int amount) {
        fuelContents = amount;
    }
    int contentsLeft() {
        return fuelContents;
    }
    void consume(int amount) {
        fuelContents = fuelContents - amount;
    }
}

class Engine {
    private FuelTank fuelTank;
    Engine(FuelTank tank) {
        fuelTank = tank;
    }
    void start() {
        fuelTank.consume(5);
    }
    boolean isRunning() {
        return fuelTank.contentsLeft() > 0;
    }
    void useEnergy() {
        fuelTank.consume(10);
    }
}

```