

Ohjelmistojen mallintaminen, koe 15.12.2010

Vastaa tehtäviin 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) Vastaa seuraaviin kysymyksiin kokonaisin lausein esseemäisesti. Keskity vain olennaisiin asioihin, vastausten yhteenlasketun pituuden ei tarvitse ylittää kahta A4-sivua.
 - (a) (5p) Vertaile vesiputousmallia ja ketterää ohjelmistotuotantoa. Kuvaile mitä samaa ne sisältävät ja mitkä ovat tärkeimmät eroavaisuudet.
 - (b) (3p) Kerro ainakin kahdesta oliosuunnittelun periaatteesta. Mikä on motivaatio periaatteiden takana? Saako periaatteita koskaan rikkoa? Miten käsite koodihaju (engl. code smell) liittyy oliosuunnitteluun?
2. (10p) Luokka- ja sekvenssikaavioita.
 - (a) (5p) Terraariossa on monia erilaisia eliöitä. Eliöt ovat joko kasveja tai liikkuvia eliöitä. Kukin eliö kuuluu pysyvästi johonkin lajiin. Osa lajeista on toistensa vihollisia. Yhdellä lajilla voi olla useita eri vihollislajeja. Liikkuvat eliöt muodostavat laumoja. Laumaan kuuluu useita eliöitä, mutta eliö voi kerrallaan kuulua vain yhteen laumaan. Liikkuvat eliöt toteuttavat liikkumisominaisuuden. Liikkumisominaisuus säätelee eliön liikkumista terraariossa. Myös laumat toteuttavat oman liikkumisominaisuutensa joka vaikuttaa jokaisen lauman sisältämän eliön liikkeeseen. Mallinna terraario luokkakaaviona.
 - (b) (1+4p) Tehtäväpaperin lopusta löytyy katkelma Java-koodia. Takaisinmallinna koodi luokka- ja sekvenssikaaviona. Sekvenssikaavio piirretään tilanteesta, jossa kutsutaan luokan Paaohjelma main-metodia. Luokkakaavioon ei tarvitse merkitä metodien nimiä.
3. (10p) Sanomalehden toimitus tarvitsee toimintansa tehostamiseksi ajanmukaisen tietojärjestelmän. Asiakaspalaverin aikana lehden edustaja kertoo seuraavaa:

Lehtemme ilmestyy joka päivä. Lehdessä julkaistaan mainoksia ja artikkeleja. Artikkelit koostuvat tekstikappaleista ja kuvista. Kukin artikkeli on pääosin yhden toimittajan kirjoittama. Joskus kirjoittajia voi olla useampia. Jokainen kuva on luonnollisesti vain yhden kuvaajan ottama. Artikkelin kirjoittaja valitsee artikkelissa käytettävät kuvat. Jokaisesta mainoksesta vastaa joku lehtemme mainosmyyjistä.

Artikkelitekstit, kuvat ja mainokset lisätään järjestelmään internetiselaimen avulla. Järjestelmän kaikki muukin toiminnallisuus lienee parasta hoitaa selaimella.

Jokainen artikkeli liittyy tietyn toimituksen (kotimaa, ulkomaa, talous, kulttuuri, urheilu) alaisuuteen. Kunkin toimituksen päällikkönä työskentelee yksi toimittaja. Toimituspäälliköillä tulee olla mahdollisuus muokata järjestelmään lisättyjä artikkeleja.

Jokainen artikkeli on pääosin tarkoitettu julkaistavaksi tietyssä lehden numerossa. Toimituspäällikkö voi myös siirtää artikkelin kokonaan toiseen numeroon. Sama mainos voi ilmestyä useassa lehdessä ja kuvaa voidaan käyttää useassakin artikkelissa.

Toimituksen tietojärjestelmää ei käytetä lehden ladontaan eli valmiin painoasun tekemiseen. Seuraavan aamun lehden artikkelit, niihin liittyvät kuvat ja mainokset siirretään toimituksen tietojärjestelmästä joka ilta erilliseen ladontajärjestelmään. Siirron hoitaa joku toimituspäälliköistä.

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

Tehtävään 2 (b) liittyvä ohjelmakoodi:

```
public class Paaohjelma {
    public static void main(String[] args) {
        Kurssi ohma = new Kurssi();
        Opiskelija o1 = new Opiskelija("Arto", 45);
        Opiskelija o2 = new Opiskelija("Matti", 27);

        ohma.lisaaOpiskelija(o1);
        ohma.lisaaOpiskelija(o2);

        ohma.arvosteleOpiskelijat();
        ohma.tulostaKurssinTulokset();
    }
}

public class ArvosanaLaskin {
    public int selvitaArvosana(int pisteet) {
        if ( pisteet<30 )      return 0;
        else if ( pisteet<35 ) return 1;
        else if ( pisteet<40 ) return 2;
        else if ( pisteet<45 ) return 3;
        else if ( pisteet<50 ) return 4;

        return 5;
    }
}
```

```

public class Kurssi {
    private ArrayList<Opiskelija> opiskelijat;
    private ArvosanaLaskin arvosanaLaskin;

    public Kurssi(){
        opiskelijat = new ArrayList<Opiskelija>();
        arvosanaLaskin = new ArvosanaLaskin();
    }

    public void lisaaOpiskelija(Opiskelija opisk){
        opiskelijat.add(opisk);
    }

    public void arvosteleOpiskelijat(){
        // for-each lause käy jokaisen opiskelijat-listassa olevan olion läpi
        for ( Opiskelija opisk : opiskelijat ) {
            int pisteet = opisk.getPisteet();
            int arvosana = arvosanaLaskin.selvitaArvosana( pisteet );
            opisk.setArvosana(arvosana);
        }
    }

    public void tulostaKurssinTulokset(){
        for ( Opiskelija opisk : opiskelijat )
            opisk.tulostaTiedot();
    }
}

public class Opiskelija {
    private String nimi;
    private int pisteet;
    private int arvosana;

    public Opiskelija(String nimi, int pisteet) {
        this.nimi = nimi;
        this.pisteet = pisteet;
    }

    public int getPisteet(){
        return pisteet;
    }

    public void setArvosana(int arvosana){
        this.arvosana = arvosana;
    }

    public void tulostaTiedot(){
        System.out.println( nimi+" "+arvosana );
    }
}

```