

Ohjelmistotekniikan menetelmät, kurssikoe 17.12.2014

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

Kokeessa saa olla mukana yksi (1) käsinkirjoitettu A4-kokoinen paperi muistiinpanoja.

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

1. (4p) Kirjoita noin 2 sivun mittainen essee aiheesta "Ohjelmistojen elinkaari ja ohjelmistotuotantoprosessin vaiheet vesiputousmallissa ja ketterissä menetelmissä".

Kirjoita kokonaisia hyvin muotoiltuja lauseita, pelkillä ranskalaisilla viivoilla ei pisteitä ole luvassa.

2. (9p) Luokka- ja sekvenssikaavioita.

Joulupukin pajassa on juuri nyt työtahti kovimmillaan. Varoke Oy on tilannut Heebo Motel -palveluunsa pajasta virtuaalisen version joulun kunniaksi.

Lelupajassa on tonttuja, joilla on eri taitoja. Jotkut ovat puuseppiä, toiset metalliseppiä, kolmannet elektroniikka-asentajia. Osa tontuista on niin vanhoja ja viisaita että heillä on useampi taito. Pajan Mestaritonttu osaa tehdä kaikkea. Tontut tekevät leluja toivelistojen mukaisesti. Kun lelu tulee valmiiksi, se merkitään toivelistalla tehdyksi. Tonttu myös merkitsee siihen puumerkinsä tulostarkkailua ja tammikuun yt-neuvottelujen varten. Kun listan kaikki lelut on merkattu valmiiksi, sihteeritonttu lähettää listasta raportin logistiikkakeskukseen käsiteltäväksi.

- (a) (3p) Mallinna yllä kuvattu järjestelmä luokkakaaviona. Merkitse osallistumisrajoitteet ja suunnat mahdollisimman tarkasti. Itsestäänselviä oliomuuttujia kuten "nimi" ei tarvitse merkitä. Metodeja ei tarvitse merkitä.
- (b) (3p) Tehtäväpaperin lopusta löytyy katkelma Java-koodia. Takaisinmallinna koodi sekvenssikaaviona. Main-luokan main-metodiin on merkitty kohta, josta alkaen sekvenssikaavio piirretään.
- (c) (3p) Mallinna tehtäväpaperin lopusta löytyvän javakoodin pohjalta oliokaavio tilanteesta jossa ohjelma on kun main-metodin viimeinen rivi on suoritettu.

3. (11p)

Yliopiston Flamma-järjestelmän inspiroimana tietojenkäsittelytieteen laitos on päättänyt hankkia uuden uutisten hallinnointiin tarkoitetun sisällönhallintajärjestelmän nimeltään "Pisara". Julkishallinnon luonteesta johtuen kehitystyö on ulkoistettu laitoksen ohjelmistotuotantoprojekteilte, onhan laitoksen motto "Teen itse, säästän ja opin".

Järjestelmä sallii tietyn käyttäjäjoukon (käytännössä ylempi hallinto sekä tiedotus-osasto) luoda, muokata ja poistaa uutisia. Uutiset koostetaan elementeistä, joilla on selkeä hierarkia.

Jokainen uutinen koostuu joukosta lukuja. Luvut puolestaan sisältävät kirjoittajan määräämässä järjestyksessä otsakkeita ja kappaleita. Sekä otsakkeet että kappaleet koostuvat sisältöelementeistä. Sisältöelementti on tässä yhteydessä yleistermi, joka kattaa yksittäisen sanan, linkin sekä kuvan. Järjestelmän linkit voivat viitata joko johonkin verkko-osoitteeseen tai suoraan toiseen järjestelmän tuntemaan uutiseen sen uniikin tunnistenumeron avulla. Sisäisen linkin etu on siis se, että vaikka uutisen verkko-osoite muuttuisikin, pysyy sen uniikki tunniste samana ja siten sisäinen linkki toimii aina, olettaen että viitteen kohteena olevaa uutista ei poisteta. Lisäksi sekä uutisella että jokaisella sen luvulla on kullakin aina yksi otsake, joka toimittaa kyseisen elementin pääotsakkeen roolia.

Uutisessa näkyy uutisen alkuperäisen luoja ja viimeisimpänä uutista muokanneen henkilökunnan jäsenen nimi.

Järjestelmä myös sallii kirjautuneiden käyttäjien lisätä uutisille lipukkeita ("tägejä"). Kaikki käyttäjät (myös vierailijat) voivat hakea uutisia lipukkeiden perusteella.

- (a) (3 + 1p) Etsi järjestelmästä käyttäjät ja ainakin viisi käyttötapausta. Nimeä käyttötapaukset kuvaavasti ja selkeästi, ja piirrä käyttötapauksista käyttötapausta-kaavio. Kuvaa lisäksi yksi käyttötapauksista tarkemmalla tasolla, ns. Cockburnin käyttötapauspohjan tai luentomonisteen tekstuaalisten kuvausten tyyliin.
- (b) (4p) Laadi järjestelmän kuvauksen perusteella määrittelyvaiheen (eli kohdealueen) luokkakaavio. Merkitse yhteyksiin kytkentärajoitteet. Nimeä yhteydet ja yhteysroolit tarvittaessa. Attribuutteja ei tarvitse merkitä, mutta nimeä luokkasi niin, että niiden rooli käy lukijalle selväksi. Muista, että toiminnallisuutta ei kannata määrittelyvaiheen luokkamalliin laittaa!
- (c) (3p) Laadi oliokaavio, jossa kuvaat järjestelmän seuraavassa tilassa: Järjestelmällä on kaksi käyttäjää ("Jukka" sekä "Jyrki") sekä yksi julkistettu uutinen. Uutisen pääotsikko on "Moi kaikki!". Uutisessa on yksi luku, jonka pääotsikko on "Testi-uutinen". Luvussa on yksi kappale jonka sisältö on sana "heips", jota seuraa kuva tiedostosta "kissat-on-kivoja.png". Uutisen alkuperäinen luoja on Jukka, mutta myös Jyrki on muokannut uutista myöhemmin.

Tehtävään 2 (b) liittyvä ohjelmakoodi:

```
public class Joulupukki {
    // ...

    public void annaLahja(Lapsi lapsi) {
        int kiltteys = lapsi.getKiltteys();

        if (kiltteys >= 10) {
            Lahja lahja = new Hammaslaakarikaynti();
            lapsi.otaLahja(lahja);

        } else {
            Lahja lahja = new Risupakkaus();
            lapsi.otaLahja(lahja);
        }
    }

    // ...
}
```

```
import java.util.ArrayList;
import java.util.List;

public class Lapsi {

    private String nimi;
    private int kiltteys;
    private List<Lahja> lahjat;

    public Lapsi(String nimi) {
        this.nimi = nimi;
        this.kiltteys = 10;
        this.lahjat = new ArrayList<>();
    }

    public void otaLahja(Lahja lahja) {
        this.lahjat.add(lahja);
    }

    public void kiroile() {
        System.out.println(this.nimi + ":-##!@@$$");
        this.kiltteys--;
    }

    public int getKiltteys() {
        return kiltteys;
    }
}
```

```
public interface Lahja {  
  
    void avaa();  
}
```

```
public class Risupakkaus implements Lahja {  
  
    @Override  
    public void avaa() {  
        // ...  
    }  
}
```

```
public class Hammaslaakarikaynti implements Lahja {  
  
    @Override  
    public void avaa() {  
        // ...  
    }  
}
```

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Joulupukki arto = new Joulupukki();  
  
        Lapsi doge = new Lapsi("dogeman");  
        Lapsi loezi = new Lapsi("loeziman");  
  
        // mallinna sekvenssikaavio tästä tilasta lähtien  
        arto.annaLahja(doge);  
        doge.kiroile();  
        arto.annaLahja(doge);  
        doge.kiroile();  
        arto.annaLahja(loezi);  
        // mallinna oliokaaviona edeltävän rivin suorituksen jälkeinen tila  
    }  
}
```