

Ohjelmistojen mallintaminen, syksy 2010, laskuharjoitus 4

huom: laskarit pidetään salissa B221

Tehtävät 1 ja 2 liittyvät luokkakaavioihin. Tehtävä 4 liittyy sekvenssikaavioihin, eli luennon 3 kalvoihin 39-59. Tehtävä 5 liittyy käsiteanalyysiin eli luennon 3 kalvoihin 16-38.

Kaikki alla olevat tehtävät tehdään etukäteen. Paikanpäällätehtävät tehtävät löytyvät laskarisivulta <https://wiki.helsinki.fi/pages/viewpage.action?pageId=76261353>

1. Seuraavassa on lueteltu käsitteitä, jotka ovat jossain suhteessa keskenään. Mitä suhdetyyppiä (normaali yhteys, kompositio, perintä, riippuvuus, ...) kussakin tapauksessa kannattaa käyttää (siis jokaiseen palloon oma suhde). Piirrä esim. pieni luokkakaavio jokaisesta tilanteesta.
 - Ruokaileva filosofi käyttää haarukkaa
 - Unixissa tiedosto on joko tavallinen tiedosto tai hakemisto
 - Tiedosto muodostuu tavuista
 - Monikulmion kulmat ovat järjestetty jono pisteitä
 - Opiskelija hallitsee Java-kielen
 - Ihmisellä on syntymävuosi
 - Hiiri, näppäimistö ja Näyttö ovat I/O-laitteita (eli tietokoneen syötelaitte)
 - Polku yhdistää kaksi kylää
 - Monopolissa on 2-8 Pelaajaa. Pelaaja voi olla yksittäinen henkilö tai joukkue. Joukkue koostuu yksittäisistä henkilöistä
 - Kaikki tiet vievät roomaan
2. Tehtäväpaperin lopusta löytyy katkelma Java-koodia. Takaisinmallinna koodi luokkakaaviona.
3. Takaisinmallinna lopun koodi oliokaaviona. Oliokaavio piirretään rivin

```
ohma.lisaaOpiskelija(o2);
```

jälkeisestä tilanteesta.

4. Takaisinmallinna lopun koodi sekvenssikaaviona. Sekvenssikaavio piirretään tilanteesta, jossa kutsutaan luokan Paaohjelma main-metodia.
5. Toisissa laskareissa oli tehtävänä laatia käyttötapausmalli Harjoitustöiden tuntikirjanpitoa tukevaan järjestelmään.

Tee määrittelyvaiheen luokkakaavio tuntikirjanpitojärjestelmästä.

Etene samaan tapaan kuin luennon elokuvateatteri- ja kampaamoesimerkeissä, eli etsi ensin luokkakandidaatit keräämällä viikon 2 laskareissa olevassa tehtäväkuvauksessa esiintyvät substantiivit. Karsi sen jälkeen ylimääräiset luokkakandidaatit, eli attribuutit, synonyymit, tekemistä tarkoittavat sanat ja asiaan kuulumattomat. Tämän jälkeen mieti minkälaisia yhteyksiä luokkien olioilla on. Lopuksi tarkenna yhteydet osallistumisrajoitteilla sekä nimeä yhteydet ja roolit tarvittaessa.

Huomioi luennon 3 kalvon 33 toteamus:

yleensä ei jakseta kerätä kaikkia substantiiveja vaan tehdään esim. synonyymien ja tekemistä tarkoittavien sanojen karsinta samalla kun "kunnollisten" substantiivien etsintä etenee

Laskuharjoituksissa paikanpäällä tehtävät tehtävät löytyvät laskarisivulta:

<https://wiki.helsinki.fi/pages/viewpage.action?pageId=76261353>

```

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);
        Opiskelija o3 = new Opiskelija("Antti", 55);

        ohma.lisaaOpiskelija(o1);
        ohma.lisaaOpiskelija(o2);
        ohma.lisaaOpiskelija(o3);

        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 );
    }
}
```