Demo 9 / 10.11

Tehtävät

1. **Ville:** Tee Villellä (ks. https://trac.cc.jyu.fi/projects/ohj1/wiki/ville) tehtävät: Rekursio. Muuta tekemiesi tehtävien määrä suhteeksi välille [0,1] ja laita pistemääräksi 0.2 tarkkuudella pyöristetty luku. Esim. jos teit 120/195 tehtävää = 0.615, niin laita 0.6 pistettä. Jos teit 140/195 = 0.72, laita 0.8 tehtävää).

2. **Oliot**: Ota vanha aliohjelma `Portaat` ja muuta sitä niin, että seuraava pääohjelma toimi:

```
public static void main(String[] args) {
    Window window = new Window();
    window.scale(0,-1,10,10);
    RPoint next = new RPoint(0,0);
    next = porras(window,next);
    next = porras(window,next);
    next = porras(window,next);
    next = porras(window,next);
    next = porras(window,next);
    next = new RPoint(next.getX()-1,next.getY());
    next = porrasAlas(window,next);
    next = porrasAlas(window,next);
    next = porrasAlas(window,next);
    next = porrasAlas(window,next);
    next = porrasAlas(window,next);
    window.showWindow();
}
```

   Vinkki: katso Graphics -kirjaston `Rpoint` -luokan dokumentaatiota. Eli ideana on, että portaalle viedään parametrinä `piste`-olio kahden reaaliluvun sijaan. Kun porras piirtää itsensä, se palauttaa sen pisteen., johon se päättyi. Näin seuraavan portaan piirtäminen on helppoa.

3. **Silmukat**: Tee edellisiä käyttäen tarvittavat aliohjelmat, jotta seuraava pääohjelma toimii (jos n=5, tekisi juuri saman kuvan kuin edellinen pääohjelma):

```
public static void main(String[] args) {
    int n = 50;
    Window window = new Window();
    window.scale(0,-1,2*n,n+10);
    RPoint next = new RPoint(0,0);
    next = portaat(window,next,n);
    next = new RPoint(next.getX()-1,next.getY());
    portaatAlas(window,next,n);
    window.showWindow();
}
```

4. **Listat**: Katso mallia: LaskeAKirjaimet.java. Tee mallia mukaellen aliohjelma, joka etsii listan pisimmän sanan. Pääohjelma tulostaa pisimmän sanan ja poistaa sitten listasta KAIKKI tämän sanan esiintymät.

5. **Listat**: Kirjoita edelliseen vielä aliohjelma `kopioi(sanat,pituus)`, joka palauttaa uuden listan, johon on kopioitu kaikki ne listan sanat, joiden pituus on `pituus`.

6. **Tiedostot**: Kirjoita edellisen tehtävän aliohjelmaa hyväksi käyttäen ohjelma, joka ottaa pääohjelman argumentista tiedoston nimen ja sitten tulostaa tiedoston niin, että tulostetaan vain ne rivit, jotka ovat yhtä pitkiä kuin pisin rivi. Helpotus: jos tiedostoa et saa muuten auki, niin ohjelman parametrinä saat antaa koko polun tiedostolle.

B1. Katso Wikistä OhjOpetuksenTyopaja kohdasta `ComTest` kuinka ComTestiä käytetään. Kirjoita testit ja aja ne tehtävien 4 ja 5 aliohjelmiin.

GURU-tehtävät

G1-2. Tee ohjelma, joka lukee kaikki komentoriviparametreissä annetut tiedostot, lajittelee tuloksen ja poistaa kaikki samanlaisten rivien esiintymät. Samanlaisiksi tulkitaan rivit joissa on samat merkit isoista ja pienistä kirjaimista välittämättä. Lopuksi ohjelma tulostaa nuo rivit. Käyttötarkoitus voisi olla esimerkiksi sellainen, että on useita postilistoja joista halutaan koota yksi yhteinen, jossa kukin listalainen esiintyy vain yhden kerran. Koosta ohjelma riittävän pienistä paloista ja testaa `ComTestillä` kukin aliohjelma. Käytä mahdollisimman paljon valmiita listoja ja niihin liittyviä metodeja.