

Olio-ohjelmoinnin perusteet – Demo 4

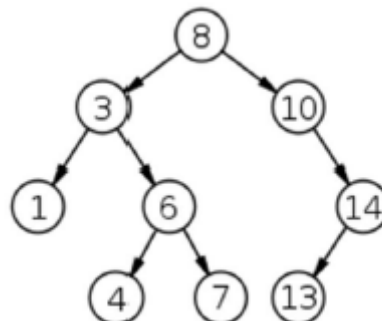
Huom! 5 tehtävää.

1. Tehtävänäsi on mallintaa lentoyhtiön varausjärjestelmän luokkakaavio. Järjestelmän piiriin kuuluu ainakin kohteita, lentoja (loma- ja reittilentoja), lentokoneita, eritasoisia asiakkaita ja henkilökuntaa. Jotkut asiakkaat omistavat lisäksi bonuskortin. (20 p.)
2. Viimeistellään **Jalkapallojoukkue**-luokan toteutus seuraavasti:
 - a. Lisää joukkueelle attribuutti budjetti (int), joka määrittää kuinka paljon rahaa joukkueella on käytettävissä pelaajien palkkoihin. Lisäksi luokkaan kirjoitetaan staattinen luokkamuuttuja PALKKAKATTO, joka määrää budjetin ylärajan. Attribuutin asetusmetodissa tulee ottaa tämä huomioon.
 - b. Kirjoita poikkeusluokka BudjettiYlitytetty. Poikkeus heitetään, jos pelaajan lisäämisen myötä yhteispalkka ylittää asetetun budjetin.
 - c. Kirjoita lopuksi testiohjelma, joka lisää pelaajia joukkueeseen ja ottaa kiinni mahdollisen poikkeuksen.(20 p.)
3. Muutetaan binääripuun toteutusta niin, että alkiot lisätään puuhun oikeaan järjestykseen. Alkiot ovat järjestyksessä, kun jokaisen alkion vasen alipuu sisältää sitä pienemmät alkiot ja oikea suuremmat. Esimerkki järjestetystä binääripuusta löytyy kuvasta 1.

Huom! Jotta puun alkioden suuruutta voidaan vertailla, tulee niiden toteuttaa rajapintaluokka Comparable. Näin ollen geneerisen luokan määrittelyn pitää olla muotoa

```
public class Binääripuu<E extends Comparable<E>>{}
```

(20 p.)



Kuva 1 Binäärinen hakupuu järjestyksessä. Lähde: Wikipedia

4. **Tiedostoja** voidaan käyttää ohjelman tilan tallentamiseen. Esimerkiksi jonkin ohjelman asetukset tai pelin tila voidaan tallentaa tiedostoon, jolloin ohjelman käyttöä voidaan jatkaa siitä mihin viimeksi jäätiin. Kannattaa katsella Javan APIsta esimerkiksi luokka *File*, *FileWriter* sekä *Scanner*. Javan olioiden suoraan levyllä kirjoitukseen kannattaa katsella luokkia *ObjectOutputStream* ja *ObjectInputStream*.

Kirjoita luokka **TiedostoLukija**, johon toteutetaan seuraavat metodit:

```
// Metodi kirjoittaa valitussa polussa olevaan tiedostoon numerot välillä 0...määrä omille riveilleen.
```

```
public static void kirjoitaNumerot(int määrä, String polku)
```

```
// Metodi lukee numerot annetusta tiedostopolusta ja palauttaa niiden summan.
```

```
public static int lueNumerot(String polku)
```

```
(20p)
```

5. Keksi olio-ohjelmointiin liittyvä demotehtävä ja kirjoita sille malliratkaisu. Tehtävässä tulee hyödyntää monipuolisesti kurssilla esitettyjä käsitteitä ja tekniikoita. (20 p.)