

## Ohjelmoinnin jatkokurssi, koe 24.6.2009/AW

Kirjoita jokaisen vastauspaperisi alkuun kurssin nimi ja kokeen päivämäärä sekä nimesi, opiskelijanumerosi ja allekirjoituksesi. Kokeessa mikään lisämateriaali ei ole sallittu.

1. Käytössäsi on luokka KokonaislukuJoukko, joka tarjoaa palvelut:

- **public KokonaislukuJoukko()** konstruoi KokonaislukuJoukko-olion, joka esittää tyhjää joukkoa. Huomaa että luokalla on siis *ainoastaan* tämä yksi konstruktori.
- **public void lisää(int luku)** lisää joukkoon alkion, ellei se jo ollut siellä
- **public void poista(int luku)** poistaa joukosta luvun, jos se siellä sattui olemaan
- **public boolean kuuluu(int luku)** selvittää, kuuluuko luku joukkoon
- **public int koko()** joukon koko (eli joukon alkoiden lukumäärä)
- **toString()**-metodi tuottaa joukosta String-esityksen, esimerkiksi {-3, 1, 76}

Mitään muuta et KokonaislukuJoukko-luokasta tiedä. Erityisesti et tunne mitään luokan määrittelemiä kenttiä tms. Sinulle on lisäksi kerrottu, että joukon alkioit saa käytyä läpi "for-each"-lauseella seuraavaan tapaan:

```
KokonaislukuJoukko luvut = new KokonaislukuJoukko();
// ... joukkoon viedään alkioita ...
for (int alkio : luvut) {
    // tänne saadaan joukon alkioit käyttöön yksi kerrallaan muuttujan alkio arvona
}
// esimerkkinä vaikkapa alkoiden summan tulostus:
int summa = 0;
for (int alkio : luvut) {
    summa = summa + alkio;
}
System.out.println(summa);
```

Et tiedä, miten tuo on toteutettu, mutta tyydy vain käyttämään sitä...;-)

Laadi tälle luokalle aliluokka KokonaislukuJoukkoPlus, jonka täydentää ylluokkaansa seuraavin joukko-operaatioin:

- **public KokonaislukuJoukkoPlus yhdiste(KokonaislukuJoukkoPlus toinen)** palauttaa arvonaan *this*-olion ja parametrina annetun olion yhdisteen. Yhdiste on joukko, joka sisältää molempien joukkojen kaikki alkioit
- **public KokonaislukuJoukkoPlus leikkaus(KokonaislukuJoukkoPlus toinen)** palauttaa arvonaan *this*-olion ja parametrina annetun olion leikkauksen. Leikkaus on joukko, joka sisältää vain ne alkioit, jotka kuuluvat molempiin joukkoihin
- **public KokonaislukuJoukkoPlus erotus(KokonaislukuJoukkoPlus toinen)** palauttaa arvonaan *this*-olion ja parametrina annetun olion erotuksen. Erotus on joukko, joka sisältää vain ne *this*-joukon alkioit, jotka eivät kuulu parametrjoukkoon

Ohjelmoi aliluokkaan vain tarpeelliset osat. Havainnollista pääohjelmalla luokan KokonaislukuJoukkoPlus käyttöä. Anna valaisevia esimerkkejä ainakin olion luonnista sekä operaatioista *lisää*, *poista*, *kuuluu*, *yhdiste*, *leikkaus* ja *erotus*.

(16 pistettä)

2. Vastaa selkeästi seuraaviin kysymyksiin:

- Mihin tarkoitukseen käytetään ilmausta *super*? Entä ilmausta *super(parametreja)*? Anna valaisevia ohjelmaesimerkkejä.
- Miten *rajapintaluokka* ja *polymorfismi* liittyvät toisiinsa? Anna valaisevia ohjelmaesimerkkejä.

Koko vastauksen maksimipituus on yksi konseptipaperiarkki eli 4 sivua.

(16 pistettä)

3. Tee ohjelma TulostaMuunnettuna, jolla voi tulostaa tiedoston näytölle siten, että valitun sisältöiset rivit korvataan toisella rivillä. Ensin ohjelma pyytää tiedoston nimen ja sitten kysyy käyttäjältä korvattavat ja korvaavat rivit. Lopuksi ohjelma tulostaa näytölle tiedoston muunnettuun rivein.

Ohjelman on varauduttava virheisiin ja käsiteltävä poikkeukset. Virheilmoitusten pitää olla havainnollisia.

### Vihjeitä:

- Saat ja kannattaa (mutta pakko ei ole!) käyttää HashMap<K,V>-luokkaa. Jos et käytä sitä, voit olettaa, ettei korvattavia rivejä ole enempää kuin 50.
- Jos käytät HashMap<K,V>-luokkaa, seuraavasta API-kuvauksen osasta voi olla hyötyä:
  - **public HashMap<K,V>()** luo tyhjän Hashtable-olion, joka kuvaa K-tyyppisiä olioita V-tyyppisiksi olioiksi
  - **public V put(K key, V value)** liittää HashMap-olioon assosiaation *key->value*; jos avaimen jo liittyy assosiaatio, metodi korvaa sen uudella ja palauttaa arvonaan vanhan merkityksen; jos avainta ei löytynyt, metodi palauttaa arvon *null*
  - **public V get(Object key)** palauttaa arvonaan *key*-avaimen liitetyn arvon; jos avainta ei löydy, metodi palauttaa arvon *null*
  - **public boolean containsKey(Object key)** *true* jos avainten joukossa on *key*, muuten *false*
  - **public V remove(Object key)** poistaa avaimen *key* ja siihen liittyneen arvon, ts. assosiaatio *key->value* poistetaan, palauttaa arvonaan poistetun arvon *value*; palauttaa *null*, jos avainta ei löydy

▪ **public String toString()** tuottaa merkkiesityksen HashMap-oliosta, komponentit muokataan niiden omilla toString()-metodeilla

- Rivin uudeksi riviksi kuvaava HashMap on tyypiltään HashMap<String,String>
- Käyttöesimerkki (käyttäjän teksti alla kursivilla):

```
Listattava tiedosto? (tyhjä merkkijono keskeyttää ohjelman)
aasikirja.txt
Tiedostoa aasikirja.txt ei löydy!
Syöttötiedosto? (tyhjä merkkijono keskeyttää ohjelman)
asiakirja.txt
OK!
Anna korvattava rivi (tyhjä merkkijono lopettaa rivien lisäämisen)
Ne varsin pulahtelevat,
Anna korvaava rivi (tyhjä merkkijono lopettaa rivien lisäämisen)
Ne varsin valehtelevat,
Anna korvattava rivi (tyhjä merkkijono lopettaa rivien lisäämisen)
arvelevat kitaraa
Anna korvaava rivi (tyhjä merkkijono lopettaa rivien lisäämisen)
arvelevat kanteletta
```

Tiedosto muunnettuna:

```
Ne varsin valehtelevat,
tuiki tyhjeä panevat,
jotka soittoa sanovat,
arvelevat kanteletta
Väinämöisen veistämäksi,
Jumalan kuvoamaksi,
hauin suuren hartioista,
veen koiran koukkuluista.
```

(18 pistettä)

*Onnea kokeeseen & lämmintä kesää!*