T-106.1206 Ohjelmoinnin perusteet Y (Java). Tentti 9.5.2007

Kirjoita jokaisen vastauspaperisi alkuun kurssin nimi, kokeen päivämäärä, nimesi, opiskelijanumerosi (myös tarkistuskirjain), vastauspaperiesi kokonaismäärä sekä allekirjoituksesi.

1. Kirjoita luokka `Jalkapallojoukkue` yhden liigaan kuuluvan jalkapallojoukkueen kuvaamiseen. Luokalla on oltava kentät `nimi`, `pisteet`, `tehdytMaalit` ja `paastetytMaalit`. Kentässä `pisteet` pidetään kirjaa joukkueen tähän asti keräämistä pisteistä, kentässä `tehdytMaalit` joukkueen tekemistä maaleista ja kentässä `paastetytMaalit` joukkueen otteluissa vastustajien tekemistä maaleista. Valitse itse kentille sopivat tyypit ja näkyvyydet. (Tehtävän yksinkertaistamiseksi joitakin muita jalkapallojoukkueen oleellisia tietoja on tässä tehtävässä jätetetty pois.)

   Kirjoita luokkaan konstruktori, joka saa parametrina luotavan jalkapallojoukkueen nimen. Uudella joukkueella pisteet, tehdyt maalit ja päästetyt maalit ovat nollia. Kirjoita sitten luokkaan seuraavat metodit (jos haluat, saat lisäksi kirjoittaa muitakin metodeita):
   - `String kerroNimi()` palauttaa tämän jalkapallojoukkueen nimen.
   - `int kerroPisteet()` palauttaa tämän jalkapallojoukkueen pisteet.
   - `void muutaPisteita(int uudetPisteet)` muuttaa tämän joukkueen pisteitä, jos parametri ei ole negatiivinen. Uudet pisteet on annettu parametrina. Jos parametri on negatiivinen, metodi ei tee mitään.
   - `void ottelu(Jalkapallojoukkue vastustaja, int omatMaalit, int vastustajanMaalit)` metodi saa parametrina yhden ottelun vastustajan sekä ottelussa nykyisen joukkueen ja vastustajan tekemät maalit. Metodi muuttaa sekä nykyisen joukkueen että vastustajan piste- ja maalitietoja ottelun lopputuloksen mukaisesti. Voitosta saa kolme pistettä, tasapelistä yhden ja tappiosta nolla pistettä.
   - `boolean parempi (Jalkapallojoukkue toinenJoukkue)` palauttaa arvon `true`, jos nykyinen joukkue on parempi kuin parametrina annettu joukkue ja muuten arvon `false`. Joukkueiden paremmuuden ratkaisee ensi sijassa pistemäärä. Jos molemmilla joukkueilla on sama pistemäärä, on parempi joukkue se, jolla tehtyjen maalien ja päästettyjen maalien erotus on suurempi. Jos tämäkin erotus on sama, joukkueet ovat yhtä hyviä, jolloin metodi palauttaa arvon `false`.
   - `String toString()` palauttaa merkkijonon, joka sisältää joukkueen nimen, maalitiedot ja pisteet.

   Kirjoita myös pääohjelma, joka luo ensin kolme jalkapallojoukkuetta ja kutsuu sitten kolmesti `ottelu`-metodia. Tämän jälkeen ohjelman on verrattava kahden joukkueen paremmuutta `parempi`-metodilla ja tulostettava paremman joukkueen nimi. Jos joukkueet ovat yhtä hyviä, ei ole väliä, kumman joukkueen nimen ohjelma tulostaa. Lopuksi ohjelman on tulostettava kaikkien joukkueiden tiedot (nimi, maalitiedot ja pisteet). Voit päättää itse joukkueiden ja otteluiden tiedot. Pääohjelman ei siis tarvitse kysyä mitään tietoja käyttäjältä. (28 p)

2. Oleta, että sinulla **on käytössäsi** luennolla esitetty luokka `OpeyOpiskelija` (sitä ei siis pyydetä tehtävässä ohjelmoimaan). Luokalla on mm. julkiset metodit `String kerroNimi()` ja `int laskeKokonaisarvosana()`, joista ensimmäinen palauttaa opiskelijan nimen ja jälkimmäinen opiskelijan kokonaisarvosanan ohjelmointikurssilta. Oleta, että sinulla on myös käytössäsi luokka `Ohjelmointikurssi`, jonka yhtenä kenttänä on taulukko `private OpeyOpiskelija[] opiskelijat`.

   Kirjoita luokkaan `Ohjelmointikurssi` metodi
   `public int laskeArvosanojenMaara(int arvosana)`
   joka tutkii `opiskelijat`-taulukon ja laskee, kuinka monella siinä olevalla opiskelijalla on parametrina annettu kokonaisarvosana. Metodi palauttaa näiden opiskelijoiden lukumäärän. Esimerkki: jos metodille annetaan parametrina 4, se laskee niiden `opiskelijat`-taulukossa olevien opiskelijoiden määrän, joiden kokonaisarvosana on 4 ja palauttaa tämän määrän. Saat olettaa, että taulukko ei sisällä `null`-alkioita.

   Huom! Sinun ei tarvitse kirjoittaa luokkaa `OpeyOpiskelija` eikä luokan `Ohjelmointikurssi` muita osia. Sinun ei tarvitse myöskään luoda `opiskelijat`-taulukkoa, lisätä siihen opiskelijoita tai muuttaa opiskelijoiden osasuoritusarvosanoja. Kirjoita vain pyydetty metodi. Huomaa, että voit tehdä tämän tehtävän, vaikka et muistasikaan mitään luennolla esitetystä `OpeyOpiskelija`-luokasta, sillä kaikki tehtävän ratkaisemiseen tarvittava tieto tuosta luokasta on annettu tässä tehtävänannossa. (20 p)

3. a) Kerro, mitä tarkoitetaan sillä, että luokka perii toisen luokan. Anna lyhyt esimerkki, josta näkyy muun muassa, miten yliluokan konstruktoria voidaan kutsua perivän luokan konstruktorista ja mitä tarkoitetaan metodin korvaamisella. Laadi esimerkki siten, että konstruktorilla ja metodilla on parametreja. Kerro myös, mitä hyötyä perinnästä on (mainitse ainakin kolme asiaa). (15 p)

   b) Selitä lyhyesti (muutamalla virkkeellä) seuraavat asiat (4 p / kohta)
   - mikä ero on alkeistyyppisellä (engl. primitive data type) ja viittaustyyppisellä (engl. reference type) muuttujalla
   - mihin Java-ohjelmissa käytetään poikkeuksia (engl. exception) ja miten poikkeuksia voidaan käsitellä (hyvin lyhyt periaatteellinen kuvaus riittää, ei tarvitse esittää tarkkaa Java-koodia)
   - mitä tarkoittaa abstrakti luokka (engl. abstract class) ja mitä hyötyä on abstraktista luokasta

4. Käytössäsi on luokka `Syottotiedosto` tekstitiedostojen lukemiseen. Luokalla on konstruktori `Syottotiedosto(String nimi)`. Parametrina konstruktorille annetaan tiedoston nimi. Luokassa on metodit:
   - `public String lueRivi()` palauttaa arvonaan syöttötiedoston seuraavan rivin. Jos kunnollisen tiedoston luonti epäonnistui ja metodia silti kutsutaan, metodi palauttaa arvon `null`.
   - `public boolean onLoppu()` palauttaa arvon `true`, jos tiedosto on jo luettu loppuun. Jos metodi palauttaa arvon `false`, tiedostosta on vielä lukematta ainakin yksi rivi.
   - `public boolean onkoAuki()` palauttaa arvon `true`, jos konstruktori onnistui luomaan kunnollisen tiedoston.

   Luokka `Syottotiedosto` pitää itse huolen poikkeuksista, sinun ei tarvitse varautua niihin mitenkään kirjoittamassasi ohjelmassa.

   Ratkaise luokan `Syottotiedosto` avulla seuraava ongelma (sinun **ei siis tarvitse** ohjelmoida `Syottotiedosto`-luokkaa, vaan voit olettaa sen olevan valmiina):

   Kirjoita ohjelma, joka saa kaksi komentoriviparametria: ensimmäisenä komentoriviparametrina teksititiedoston nimen ja toisena komentoriviparametrina yhden sanan. Ohjelma lukee tekstitiedoston ja tulostaa kaikkien niiden rivien numerot, joilla toisena komentoriviparametrina annettu sana esiintyy. Tekstitiedoston ensimmäisen rivin numero on 1, toisen 2 jne. Rivin numero tulostetaan myös siinä tapauksessa, että sana esiintyy rivillä toisen sanan osana. Jos sana esiintyy samalla rivillä useampaan kertaan, riittää kuitenkin se, että rivin numero tulostetaan vain kerran.

   Jos sana ei esiinny lainkaan tiedostossa sen millään rivillä, ohjelma ilmoittaa lopuksi käyttäjälle, että sanaa ei löytynyt.

   Jos ohjelmalle ei ole annettu komentoriviparametreja tai niitä on annettu väärä määrä, ohjelma ilmoittaa virheestä ja lopettaa toimintansa. Myös siinä tapauksessa, että tiedoston avaaminen ei onnistu, ohjelma ilmoittaa virheestä ja lopettaa toimintansa.

   Vinkki: Voit käyttää sanan etsimiseen riviltä `String`-luokan metodia `public int indexOf(String str)` joka palauttaa sen indeksin, josta parametrina annettu merkkijono (esimerkiksi sana) alkaa nykyisessä merkkijonossa. Jos parametrina annettu merkkijono ei sisälly lainkaan nykyiseen merkkijonoon, metodi palauttaa arvon -1. Jos siis metodi palauttaa tätä suuremman arvon, tiedät parametrina annetun merkkijonon esiintyvän nykyisessä merkkijonossa. (25p)

Muista vastata kurssin palautekyselyyn! Kyselyyn vastaamisesta saa 200 harjoitustehtäväpistettä. Kysely on auki 23.5. asti.