Arvosteluperusteet Ohjelmoinnin perusteet -kurssin kurssikokeen 16.10. tehtävään 1.

Arvosteluperusteet laati Jukka Stenlund.

> Tehtävänanto:
>
> Selitä lyhyesti ja selkeästi käsitteet *muuttujan tyyppi, parametri, kapselointi, new-ilmaus, toString()-metodi, indeksi.*
>
> (18 pistettä)

Tehtävässä annettiin kunkin käsitteen määrittelystä 0-3 pistettä. Seuraavassa on annettu käsitteille esimerkkiluonnehdinnat. Hyvinkin lyhyestä vastauksesta saattoi saada täydet pisteet, jos vastaus kertoi käsitteestä oleellisen.

Muuttujan tyyppi

Vastauksista, joissa vain lueteltiin muuttujien mahdollisia tyyppejä annettiin kaksi pistettä. Täyteen kolmeen pisteeseen vaadittiin sen selittämistä, mitä tarkoittaa, että muuttujalla on jokin tyyppi. Muuttujan tyyppi ilmaisee, millaista tietoa muuttuja voi edustaa. Esim. int-tyyppinen muuttuja voi saada arvokseen vain kokonaislukutyyppisiä arvoja.

Jossain vastauksessa oletettiin, että seuraavassa koodissa viimeisellä rivillä esiintyvä sijoitus olisi virheellinen

```
int a=3;
double b;
b=a;
```

Todellisuudessa ylläoleva koodinpätkä on mahdollinen: kokonaislukutyyppisen arvon voi sijoittaa liukulukutyyppiseen muuttujaan. Toiseen suuntaan sijoitus ei onnistu. Liukuluku voi sisältää desimaaliosan, joka kadotettaisiin, jos luku sijoitettaisiin kokonaislukutyyppiseen muuttujaan. Tenttiä arvosteltaessa ei kuitenkaan sakotettu siitä, jos vastauksessa väitettiin, että yllä annettu koodipätkä on virheellinen.

Parametri

Metodin määrittelyssä - nimen jälkeen sulkeissa - määriteltyjä muuttujia, joiden avulla metodille voidaan antaa lähtötietoja. Parametrien avulla voidaan vaikuttaa metodin käyttäytymiseen kutsukertakohtaisesti.

Tehtävässä piti osata kirjoittaa parametreista metodin parametrit Javassa -merkityksessä.

Vastauksista, joissa parametrin kerrottiin olevan yksinkertaisesti muuttujan arvo tms. sai yhden pisteen.

Vastauksista, joissa annettiin esimerkki parametrejä sisältävän metodin otsakkeesta, mutta ei sen

kummemmin selitetty asiaa, sai kaksi pistettä.

Kolmen pisteen vastauksissa osattiin kertoa, että metodin parametreille annetaan arvot metodin kutsun yhteydessä. Metodi-termiä ei välttämättä tarvinut käyttää, sai puhua esim. funktiosta. Oleellista oli jotenkin viitata siihen, että parametrien avulla välitetään tietoa metodista toiseen ja jotenkin selvittää sitä, miten tämä tapahtuu, eli että parametrit ovat metodin muuttujia, joille annetaan arvot metodin kutsussa.

Kapselointi

Tiedon sisäiseen esittämiseen käytettävien kenttien piilottamista tietorakenteen käyttäjältä siten, että tietoa voi käsitellä vain aksessorien kautta. Mahdollistaa tiedon sisäisen esitysmuodon vaihtamisen ilman, että muutos heijastuu rakenteen kyttäjiin. Menettelyn avulla voidaan myös asettaa rajoja sille, millaisiin tiloihin olio voidaan asettaa, so. millaisia arvoja olion kentät voivat saada. Kapselointi mahdollistaa tämän, koska kenttien arvoja päivitetään vain aksessorimetodien kautta, ja näihin voidaan ohjelmoida tarkistuksia.

Vastauksista, joissa todettiin kapseloinnista vain, että se selkiyttää koodia, että siinä piilotetaan ohjelmasta osa, että se on menetelmä, jolla ohjelma jaetaan osiin tai että olio-ohjelmointi on kapselointia sai yhden pisteen.

Kuvainnollinenkin teksti saattoi riittää kolmeen pisteeseen.

Täydet pisteet saadakseen ei tarvinnut kertoa, mihin kapselointia tarvitaan, jos osasi kertoa, miten se toteutetaan.

new-ilmaus

new-ilmauksella luodaan uusi olio eli luokan ilmentymä.

Vastauksista, jotka olivat muuten hyviä, mutta joissa puhuttiin *olion ilmentymän* luonnista sai kaksi pistettä. new-ilmauksella ei luoda ilmentymää oliosta vaan luokasta. Luokan ilmentymät ovat olioita, mutta olion ilmentymän käsite on ainakin tämän kirjoittajalle tuntematon.

Tehtävässä ei sakotettu siitä, jos vastauksessa annettiin syntaksiltaan virheellinen esimerkki new-ilmauksen käytöstä.

toString-metodi

- toString()-metodin on tarkoitus palauttaa arvonaan merkkijonoesitys, joka kuvaa havainnollisesti olion tilaa. Jotta näin tapahtuisi omista luokista tehtyjen olioiden kohdalla, on omiin luokkiin itse ohjelmoitava järkevästi toimiva toString()-metodi.
- Jos olio tulostetaan seuraavaan tapaan, kutsutaan olion toString()-metodia ja tulostetaan palautettu arvo:

```
Viljavarasto varasto=new Viljavarasto(....);
System.out.println(varasto);
```

Vastauksesta, jossa ei kerrottu, että toString()-metodia kutsutaan, kun olio tulostetaan yllä olevan

esimerkin tapaan, sai korkeintaan kaksi pistettä.

Myös vastauksista, joissa todettiin vain yksinkertaisesti, että toString()-metodin avulla saadaan merkkijonoesitys olion tilasta, annettiin kaksi pistettä.

Joissakin vastauksissa puhuttiin siitä, että toString()-metodi palauttaa olion kenttien arvoista merkkijonoesityksen, mutta jäi epäselväksi, tiesikö vastauksen kirjoittaja, että näin tapahtuu vain, jos olion luokkaan on kirjoitettu näin toimiva toString()-metodi. Tällaisistakin vastauksista annettiin kaksi pistettä.

Vastauksista, jotka liikkuivat oikean asian liepeillä siihen kuitenkaan menemättä sai yhden pisteen.

Jos toString()-metodia väitti String-luokan metodiksi, sai korkeintaan kaksi pistettä. Toki String-luokallakin on toString()-metodi, koska kaikilla luokilla on, mutta koska toString()-metodi on *kaikilla* luokilla, ei ole mielekästä sanoa, että se olisi String-luokan metodi.

indeksi

Indeksin avulla yksilöidään taulukon alkio.

Jos totesi indeksistä vain, että se on osoitin, mutta ei kirjoittanut mitään indeksin käyttämisestä taulukkojen yhteydessä, sai yhden pisteen.

Jos totesi, että indeksi kertoo, jontako alkiota taulukossa on, eikä kirjoittanut mitään siitä, että indeksin avulla voidaan viitata taulukon yksittäiseen alkioon, sai yhden pisteen.

Jos kertoi indeksistä muuten oikein, mutta antoi ymmärtää, että Javassa taulukoiden ensimmäinen indeksi on 1 (eikä 0, kuten oikeasti on asia) sai silti täydet kolme pistettä. Käytännön ohjelmoinnin kannalta tämä voi tietysti olla kohtalokaskin virhe, muttei indeksin käsitteen ymmärtämisen kannalta.