

Harjoitus 5 (viikko 7)

- Mikäli tehtävissä on jotain epäselvää, laita sähköpostia vastuuopettajalle (jorma.laurikkala@tuni.fi).
- Muista noudattaa hyvän ohjelmointitavan perusteita (Lausekielinen ohjelmointi I -kurssin luentomateriaalin 11. luku) ja Lausekielinen ohjelmointi II -kurssilla opettuja hyviä tapoja. Muista muun muassa kirjoittaa metodin otsikkoon liittyvä yleisluonteinen kommentti metodin tarkoituksesta sekä metodin mahdollisesti saamista ja palauttamista tiedoista ellei kyseessä ole aksessori. Tee jokaisen tiedoston alkuun kommentti, jossa on harjoituksen ja tehtävän numero sekä nimesi ja sähköpostiosoitteesi. Noudata myös tämän kurssin kuluessa opetettavia hyviä tapoja: älä muun muassa nimeä aksessoreita kahta kieltä käyttäen.
- Lue syötteet *Scanner*-luokalla ja käytä syötteiden lukemiseen vain yhtä *Scanner*-oliota. Esittele lukija vakio-ominaisuutena luokka-attribuuttina, jos syötteitä on luettava kahdessa tai useammassa metodissa.
- Mikroharjoituksissa saa apua ongelmakohtiin. Perjantaisin pidettävässä ohjatussa ryhmässä avustetaan hieman enemmän kuin muissa ryhmissä.
- **Palatusaikaa on jatkettu.** Palauta vastauksesi WETO-järjestelmään (<https://wetodev.sis.uta.fi/>) viimeistään ensi viikon perjantaina 14.2. klo 12.00 (keskipäivä) **23.55.**
- WETO tarkistaa palautuksia automaattisesti sekä tulosteita vertailemalla että tutkimalla luokkien rakennetta ja toiminnallisuutta. Automaattisesta tarkistuksesta ja erityisesti WETO:n tekemistä oletuksista attribuuttien ja niiden aksessoreiden suhteen on kerrottu kurssisivujen *Harjoitukset | Tarkistus | Automaattinen* -kohdassa. Tutustu myös yleisempiä ohjeita sisältävään *Harjoitukset | Tarkistus* -kohtaan.
- **Testaa luokkaasi omatoimisesti tehtävässä kuvatulla tavalla.** Oman testauksen tekemättä jättäminen katsotaan virheeksi. Älä tarkista syötteitä niiden lukemisen yhteydessä ellei tarkistusta erikseen pyydetä, koska metodit tarkistavat parametriarvonsa, mikäli mahdollista.
- **Kaikki opettajien tarkistamat tehtävät arvioidaan hyvän ohjelmointitavan osalta.** Näin esimerkiksi huono sisennys tai puutteellinen kommentointi voi tuottaa nollan, vaikka ohjelma läpäisee WETO:n testit. Muista erityisesti metodin otsikkoon liittyvä yleisluonteinen kommentti.

1. Korvaa *Object*-luokan *toString*-metodi tehtävän 4.5 *Ajatelma*-luokassa siten, että korvattu metodi palauttaa luokan attribuutin arvon lainausmerkkien sisässä. *Override*-annotaatiosta on paljon apua aina, kun tavoitteena on korvata metodi.

Kurssisivujen *Harjoitukset | Tehtävät* -kohdassa on annettu *AjatelmaTesti1*-luokka.

Testiluokan tuloste on alla annettu, kun luokka toteutettu oikein ja testiluokalle on annettu syötteeksi merkkijono "The time is always right to do what is right."

```
Hello! I test proverbs.  
Please, enter a proverb:  
The time is always right to do what is right.  
"The time is always right to do what is right."
```

2. Korvaa *Object*-luokan *equals*- ja *hashCode*-metodit 1. tehtävän *Ajatelma*-luokassa.

Ajatelmat ovat samoja, jos niiden lauseet ovat samat. Muista varautua mahdollisiin virhetilanteisiin *equals*-metodissa joko **try-catch**-lauseella tai valintarakenteella. Korvaa *hashCode*-metodi samalla menetelmällä kuin esimerkissä, joka löytyy osoitteesta: <https://www.sis.uta.fi/~oope/oope1/kevat-2020/luennot/luento10/esimerkit/>. Esimerkin menetelmällä laskien ajatelman hajautuskoodi perustuu attribuutin hajautuskoodiin:

```
int tulos = 17;  
if (lause == null) {  
    tulos = 31 * tulos + 0; // Nollan lisäys selvyiden vuoksi.  
}  
else {  
    tulos = 31 * tulos + lause.hashCode();  
}
```

Harjoitus 5 (viikko 7)

Kurssisivujen *Harjoitukset | Tehtävät* -kohdassa on annettu *AjatelmaTesti2*-luokka.

Testiluokan tuloste on alla annettu, kun luokka toteutettu oikein.

```
Hello! I test proverbs.  
"The time is always right to do what is right."  
"The time is always right to do what is right."  
"Haste makes waste."  
true  
true  
true  
false  
false  
false  
-1167813138  
-1167813138  
885149235
```

3. Peri 2. tehtävän *Ajatelma*-luokasta *Aforismi*-luokka, jolla on kätkeyty *lausuja*-attribuutti (*String*) aforismin tunnetulle tai oletetulle kirjoittajalle.

Tee attribuutille julkiset aksessorit. Kirjoita luokalle julkinen kaksiparametrinen rakentaja, jonka jälkimmäinen parametri on lausujalle. Kutsu rakentajassa ylliluokan parametrilista rakentajaa. Tarkista asettavan aksessorin parametri ja rakentajan toinen parametri. Molemmista metodeista heitetään *IllegalArgumentException*-poikkeus, jos parametrin arvo on **null** tai siinä on vähemmän kuin kaksi merkkiä. Huomaa, että ylliluokan rakentaja tarkistaa rakentajan ensimmäisen parametrin. Kerro molempien metodien otsikossa **throws**-ilmauksella minkä poikkeuksen metodi voi heittää. Voit lyhentää tässäkin tehtävässä ohjelmasi kutsumalla rakentajassa asetusmetodia.

Korvaa *Object*-luokan *toString*-, *equals*- ja *hashCode*-metodit *Aforismi*-luokassa. Tee korvaukset ketjuttaen sitten, että kutsut korvatuissa metodeissa **super**-viitteen avulla ylliluokan *Ajatelma* metodeja osoitteessa: <https://www.sis.uta.fi/~oope/oope1/kevat-2020/luennot/luento10/esimerkit/> annetun esimerkin tapaan.

Aforismin merkkijonoesitys koostuu kahdesta rivistä. Ensimmäisellä rivillä on ylliluokan *toString*-metodin palauttama arvo. Toinen rivi alkaa yhdysmerkillä (miinusmerkki) ja välilyönillä, joiden jälkeen tulee aliluokan oman attribuutin arvo. Rivit erotetaan toisistaan järjestelmäriippumattomasti toimivalla *System.lineSeparator*-metodilla.

Aforismit ovat samoja, jos niiden lause ja lausuja ovat samat.

Kurssisivujen *Harjoitukset | Tehtävät* -kohdassa on annettu *AforismiTesti*-luokka. Näet kommentteista mitä testiluokka tulostaa, kun *Aforismi*-luokka on toteutettu oikein.

4. Kurssisivujen *Harjoitukset | Tehtävät* -kohdassa on annettu on annettu *Akku*-luokka.

Korvaa luokassa *Object*-luokan *toString*-metodi siten, että metodi palauttaa merkkijonon, joka koostuu alusta loppuun lukien luokan nimestä, yhdestä välilyönnistä ja lataustasosta. Näin paluuarvo voisi olla esimerkiksi "Akku K". Selvitä luokan nimi *Class*-luokan *getSimpleName*-metodin avulla.

Harjoitus 5 (viikko 7)

Korvaa *Object*-luokan *equals*- ja *hashCode*-metodit. Akut ovat samat, jos niiden varaus-
tasot ovat samat. Saat merkin hajautuskoodin nopeiden selville *Character*-luokan *hash-
Code(char)*-metodilla. Laske hajautuskoodi luento esimerkissä käytetyllä menetelmällä
(<https://www.sis.uta.fi/~oope/oope1/kevat-2020/luennot/luento10/esimerkit/>).

Testaa korvattuja metodeja *AkkuTesti*-luokassa. Tarkista *equals*-metodin virheenkäsitte-
lyn toimivuus myös siten, että vertaillet akkua **null**-arvoon ja väärän tyyppiseen olion.

5. Tee Javalla geneerinen *Pari*-luokka, jolla on tyyppiparametrit *T* ja *U*. Luokan otsikko on:

```
public class Pari<T, U>
```

Luokalla on kätetty *T*-tyyppiparametrilla esitelty *ensimmäinen*-attribuutti parin ensim-
mäiselle arvolle. Luokan toinen attribuutti on kätetty, tyyppiparametriltaan *U* ja nimel-
tään *toinen*. Kahden tyyppiparametrin avulla luokan oliot voivat säilöä erityyppisten arvo-
jen pareja. Tyyppiparametrien arvoiksi voidaan kiinnittää esimerkiksi *String* ja *Integer*.

Kirjoita luokalle parametrillinen rakentaja, jonka parametrit on esitelty tyyppiparametreja
käyttäen. Ensimmäisen parametrin tyyppiparametri on *T*. Toisen parametrin tyyppipara-
metri on *U*. Tee molemmille attribuuteille julkiset aksessorit, joissa käytetään tyyppipa-
rametreja lukevien parametrin paluuarvoissa ja asettavien metodien parametreissa sa-
maan tapaan kuin osoitteessa: [https://www.sis.uta.fi/~oope/oope1/kevat-
2020/luennot/luento09/esimerkit/geneerinen_laatikko/](https://www.sis.uta.fi/~oope/oope1/kevat-2020/luennot/luento09/esimerkit/geneerinen_laatikko/) annetussa *Laatikko*-esimerkissä.
Älä tarkista parametrin arvoja rakentajassa tai asettajassa, vaan aseta attribuutille suo-
raan parametrin arvo.

Kurssisivujen *Harjoitukset* | *Tehtävät* -kohdassa on annettu *PariTesti*-luokka. Näet kom-
menteista mitä testiluokka tulostaa, kun *Pari*-luokka on toteutettu oikein.

6. Uudessa Tampereen yliopistossa on seitsemän tiedekuntaa. Tee Javalla *Tiedekunta*-
niminen lueteltu tyyppi, jonka arvot ovat tiedekuntien lyhenteet, jotka on annettu osoit-
teessa: <https://www.tuni.fi/fi/tutustu-meihin/yliopisto/tiedekunnat>.

Tulosta luetellun tyypin arvot näytölle *TiedekuntaTesti*-nimisessä testiluokassa. Voit kat-
soa tähän tehtävään mallia *Maa*-esimerkistä: [https://www.sis.uta.fi/~oope/oope1/kevat-
2020/luennot/luento09/esimerkit/lueteltu_tyyppi/](https://www.sis.uta.fi/~oope/oope1/kevat-2020/luennot/luento09/esimerkit/lueteltu_tyyppi/).

TiedekuntaTesti-ohjelma tulostaa seuraavaa:

```
Hello! The faculties of the University of Tampere are:  
ITC  
MAB  
EDU  
MET  
BEN  
ENS  
SOC
```

Harjoitus 5 (viikko 7)

7. Tee Javalla metodi, joka laskee moniko sille parametrina välitetyn yksiulotteisen *Object*-tyyppisten alkioden taulukon alkioista on samoja *equals*-metodilla verraten. Itseensä verrattuja olioita ei lasketa samoiksi. Metodi palauttaa samojen alkioden lukumäärän. Paluuarvo on -1, jos parametri on **null** tai taulukossa on yksi tai useampi **null**-arvoinen alkio.

Jos parametrin arvo on esimerkiksi {'x', "abc", 1, 2.0, 1, 'x'}, on paluuarvo ~~kaksi~~ neljä, koska ensimmäiseen ja viimeiseen alkioon liittyvät oliot sekä kolmanteen ja toiseksi viimeiseen alkion liittyvät oliot ovat *equals*-metodin mukaan samoja. Jos parametri on {'x', 'x'}, on paluuarvo kaksi. Jos parametri on {'x', 'x', 'x'}, on paluuarvo kolme. Paluuarvo on nolla, jos parametri on esimerkiksi {"abc"}, koska listalla ei ole muita alkioita.

Metodin otsikko on:

```
private public static int laskeSamat(Object[] tiedot)
```

Parametrin nimen saa muuttaa paremmaksi.

Sijoita metodisi pääohjelman seuraksi *SamojenLaskuri*-nimiseen luokkaan. Kutsu metodiasi pääohjelmasta eri parametriarvoilla. Anna parametriksi myös **null**-arvo ja **null**-arvon sisältävä taulukko.

8. Jatka omien luokkiesi parissa 4.7. tehtävään pohjautuen siten, että korvaat *Object*-luokan *toString*-metodin sekä ali- että ylikuokassa. Kutsu ylikuokan korvattua metodia aliluokan metodista **super**-viitteen avulla.

Testaa korvausta erillisessä testiluokassa.