

## Harjoitus 3 (viikko 5)

- 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 opetettuja 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 vakioomuotoisena luokka-attribuuttina, jos syötteitä on luettava kahdessa tai useammassa metodissa.
- Mikroharjoituksissa saa apua ongelmakohtiin. Perjantaisin pidettävässä ohjatussa ryhmässä avustetaan hieman enemmän kuin muissa ryhmissä.
- Palauta vastauksesi WETO-järjestelmään (<https://wetodev.sis.uta.fi/>) viimeistään ensi viikon torstaina 30.1. klo 12.00 (keskipäivä).
- 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.
- Ota yhteyttä harjoitusryhmäsi vetäjään tai kurssin vastuuopettajaan, jos et keksi järjellisessä ajassa miksi WETO hylkää palautuksesi.

1. Kirjoita Java-kielellä *Rosvo*-luokka. Rosvon *anastukset*-attribuutti (**int**) kertoo montako omaisuusrikosta rosvo on suorittanut. *Hyvis*-attribuutin (**boolean**) arvo on **true**, jos rosvo onkin oikeasti hyvis.

Molemmat attribuutit ovat kätkeytyjä ja niillä on julkiset aksessorit. Kirjoita parametrin oletusrakentaja, jossa attribuuteille asetetaan arvot 2 ja **false**. Tee myös kaksiparametrinen rakentaja, jonka ensimmäinen parametri on anastuksille. Rakentajat ovat julkisia.

Tarkista anastuksiin liittyvien parametrin arvot parametrillisessa rakentajassa ja asetusmetodeissa. Anastusten lukumäärä muuttuu vain, jos parametrin arvo on suurempi tai yhtä suuri kuin nolla ( $\geq 0$ ). Attribuutille jää Javan antama alkuarvo, jos sen parametriarvo on virheellinen. Jälkimmäisen attribuutin arvo asetetaan aina suoraan.

Kirjoita omaan tiedostoonsa pääohjelman sisältävä *RosvoTesti*-luokka. Tee testiluokassa seuraavaa:

- luo rosvo-olio parametrittomalla rakentajalla,
- tarkista, että olion tila on 2 ja **false** tulostamalla lukumetodien paluuarvot näytölle,
- yritä muuttaa olion tilaa kutsumalla asetusmetodeja arvoilla -1 ja **true**,
- tarkista, että olion tila on kutsujen jälkeen 2 ja **true**,
- luo toinen rosvo-olio antamalla parametrillisen rakentajan ensimmäiselle parametrille nollaa suurempi arvo ja toiselle parametrille **true**-arvo,
- tarkista toisen olion tila,
- luo kolmas rosvo-olio antamalla parametrilliselle rakentajalle arvot -1 ja **true** ja
- tarkista, että olion tila on 0 ja **true**.

## Harjoitus 3 (viikko 5)

---

2. Siirrä *Poliisi*-luokalle (tehtävä 2.4) ja *Rosvo*-luokalle (1. tehtävä) yhteinen *hyvis*-attribuutti ja attribuutin aksessorit abstraktiin *Henkilo*-luokkaan. Kirjoita luokalle rakentaja, jonka ainoa parametri on **boolean**-tyyppinen. Luokalla ei ole muita rakentajia. Rakentajassa ja asettavassa aksessorissa ei tehdä tarkistuksia.

Peri *Poliisi*- ja *Rosvo*-luokat *Henkilo*-luokasta. Kutsu aliluokkien kaikissa rakentajissa *Henkilo*-yliluokan parametrillista rakentajaa *super*-metodia käyttäen. Poliisin oletusrakentajan *super*-kutsussa parametrin arvo on **true**, kun taas rosvon oletusrakentajassa parametrin arvo on **false**. Molempien aliluokkien parametrillisissa rakentajissa *super*-metodin parametrin arvoksi annetaan aliluokan rakentajaan välitetty parametriarvo.

Testaa luokkia erillisellä *HenkiloTesti*-nimisellä luokalla, jossa luot olioita *Poliisi*- ja *Rosvo*-luokista ja tulostat näytölle kaikkien lukuaksessoreiden palauttamat arvot. Testiluokassa ei tarvitse lukea syötteitä käyttäjältä.

3. Kurssisivujen *Harjoitukset | Tehtävät* -kohdassa on annettu abstrakti *Voimala*-luokka. Peri tästä luokasta konkreettinen *Ydinvoimala*-aliluokka, jolla ei ole omia attribuutteja. Tee *Ydinvoimala*-luokkaan yksiparametrinen rakentaja, jossa kutsut *Voimala*-luokan yksiparametrista rakentajaa. Toteuta *hajoa*-metodi siten, että metodissa kuvaillaan kuinka *ydinvoimala* hajoa.

Testaa luokkasi erillisessä *YdinvoimalaTesti*-luokassa siten, että luot *ydinvoimalaolion* yksiparametrillisella rakentajalla ja kutsut toteuttamaasi metodia olion kautta. Testiluokassa ei tarvitse lukea syötteitä käyttäjältä.

4. Kurssisivujen *Harjoitukset | Tehtävät* -kohdassa on annettu *Ohjattava*-rajapinta ja *LaiteTesti*-luokka. Kirjoita *Laite*-luokka, jolla on kätkeyty **int**-tyyppinen *tila*-attribuutti laitteen tilalle. Tee luokalle oletusrakentaja, jossa laitteen tilaksi asetetaan *Ohjattava.SAMMUTETTU*. Älä kirjoita aksessoreita, vaan toteuta luokassa *Ohjattava*-rajapinnan metodit siten, että ne toimivat rajapinnan kommentteissa esitetyllä tavalla. Käytä toteutetuissa metodeissa rajapinnan vakioita lukujen 1, 2 ja 3 asemasta.

Näet testiluokan kommentteista mitä testiluokan tulisi tulostaa näytölle, kun metodit on toteutettu oikein. Ohjelma on helpointa kääntää ja ajaa, kun molemmat luokat ja rajapinta ovat samassa hakemistossa.

5. Kirjoita *Nisakas*-luokasta periytyvä konkreettinen *Karhu*-luokka. Luokkaan ei tarvitse keksiä omia attribuutteja tai metodeja.

Tee karhulle parametrin rakentaja, jossa kutsut yliluokan parametrillista rakentajaa parametriarvoja **true** ja 120 käyttäen.

Korvaa *onkoIso*-metodi siten, että yli 250 kilon painoiset karhut ovat suuria.

Toteuta *syö*-metodi sopivaksi katsomallasi tavalla. Toteuta luokassa *Tervehtiva*-rajapinnan abstrakti metodi. *Syö*- ja *moikkaa*-metodien tulosteet ovat vapaavalintaisia sillä rajoituksella, että tulosteet ovat pituudeltaan maltillisia. Nämä metodit eivät tulosta mitään, jos karhu on kuollut.

## Harjoitus 3 (viikko 5)

---

*Nisakas*-luokka ja *Tervehtiva*-rajapinta ovat kuudennen luennon esimerkeissä: <http://www.sis.uta.fi/~oope/oope1/kevat-2020/luennot/luento06/esimerkit/rajapinta/>.

Testaa luokkasi erillisessä *KarhuTesti*-luokassa siten, että luot ensin pienen ja suuren karhun ja tarkistat sitten *onkoIso*-metodin paluuarvot tulostamalla, että olet korvannut metodin oikein. Kutsu *syö*- ja *moikkaa*-metodeja jommankumman tai kummankin olion kautta. Testiluokassa ei tarvitse lukea syötteitä käyttäjältä.

*Karhu*-luokka on helpointa kääntää ja testata, kun se, yliluokka ja rajapinta ovat samassa hakemistossa.

6. Keksi tehtävässä 2.7 tekemällesi omalle luokalle joko abstrakti tai konkreettinen yliluokka tai konkreettinen aliluokka. Uudella luokalla tulee olla valinnastasi riippumatta ainakin yksi attribuutti, aksessorit ja rakentaja.

Varmista ennen uuden luokan toteutusta, että luokkien välinen periytymissuhde on käsitteellisesti luonteva. Lauseiden “Aliluokka *IsA* Yliluokka.” ja “Aliluokka on Yliluokka.” tulisi vaikuttaa luontevilta. Esimerkiksi “Auto *IsA* Ajoneuvo.” kuulostaa järkevältä.

Jos teit nyt yliluokan, niin kirjoita siihen metodi, jonka korvaat (konkreettinen yliluokka) tai toteutat (abstrakti yliluokka) aliluokassasi. Tämä metodi ei saa olla aksessori. Jos teit nyt aliluokan, niin korvaa yliluokan metodi aliluokassa. Korvattava metodi ei saa olla aksessori.

Testaa luokkia erillisessä testiluokassa. Testiluokassa ei tarvitse lukea syötteitä käyttäjältä. Luokat on sijoitettava omiin tiedostoihinsa.