
Write code that makes proper use of assertions, and distinguish appropriate from inappropriate uses of assertions. Identify correct statements about the assertion mechanism.

Mitä pitäisi osata?

Assert-lauseet ovat JDK 1.4:n uusi ominaisuus. Vanhemmissa Javan versioissa tätä ominaisuutta ei ole ollut käytössä ja se saattaa siksi olla monille hieman vieras. Sertifiointikokeessa vaaditaan kuitenkin assert-lauseiden syntaksin hallinta. Periaatteessa kyse on melko yksinkertaisesta toiminnallisuudesta, jossa tärkein muistettava asia on oikea syntaksi.

Vaikeustaso: keskivaikea

Mitä assert-lauseet ovat?

Assert-lause on eräänlainen väittämä, jossa tehdään jokin oletus ohjelmakoodista. Ohjelmoija olettaa tämän väittämän olevan aina voimassa kun koodia suoritetaan. Assert-lauseella tarkistetaan, että näin todellakin aina on. Mikäli sääntö ei ole voimassa, on kyseessä virhetilanne, joka aiheuttaa poikkeuksen.

Esimerkki

Käsiteltäessä ihmisten ikiä oletetaan, että ikä on aina suurempi kuin nolla. Koodiin voitaisiin periaatteessa kirjoittaa eri kohtiin tarkistuksia tyyliin `if (ikä < 0) ...`, mutta tästä tulee melko työlästä ja tarkistusten kirjoittaminen vaatii joka kohtaan usean rivin koodia. Tämä työläs tarkistusten kirjoittaminen on myös melko turhaa, koska käytännössä aina ihmisten ikä on suurempi kuin nolla. Assert-lauseella tarkistus voidaan tehdä yksinkertaisesti yhdellä koodirivillä.

Mihin assert-lauseita tarvitaan?

Käytännössä assert-lauseita käytetään ohjelman testaamiseen. Niiden tarpeellisuutta voi verrata koodin kommentoinnin tarpeellisuuteen: koodi toimii aivan hyvin vaikka siinä ei olisi yhtään kommenttia, mutta sitä on hankalaa lukea ja ylläpitää. Samoin koodi toimii normaalisti ilman yhtäkään assert-lauseetta, mutta assert-lauseilla koodin testaaminen helpottuu ja laatu paranee.

Koska assert-lauseiden pääasiallinen tarkoitus on koodin laadun parantaminen ja testauksen helpottaminen, ovat assert-lauseet oletusarvoisesti pois päältä. Jos Java-koodia suoritetaan normaalisti käskyllä

```
java OhjelmanNimi
```

ei assert-lauseita huomioida ollenkaan. Assert-lauseet täytyy erikseen ottaa käyttöön käskyllä

```
java -enableassertions OhjelmanNimi
```

! Sertifiointikokeessa voi olla kysymys, jossa assert-lauseita sisältävää koodia ajetaan käskyllä `java OhjelmanNimi`. Tällöin on muistettava, että koodissa ohitetaan kaikki assert-lauseet.

Vaihtoehtoisia tapoja `java -enableassertions OhjelmanNimi -käskyn` lisäksi saada assert-lauseet käyttöön on käyttää `-ea` määritystä tyyliin

```
java -ea OhjelmanNimi
```

Lisäksi on mahdollista ottaa assert-lauseet käyttöön vain tietyssä luokassa tai paketissa tyyliin

```
java -enableassertions:paketti.alipaketti.Luokka OhjelmanNimi
```

Assert-lauseen syntaksi

Assert-lauseen perussyntaksissa annetaan käsky `Assert` sekä jokin lause, joka palauttaa boolean-arvon. Syntaksi menee silloin seuraavasti:

```
assert booleanlause;
```

Esimerkki

Ilmoitetaan assert-lauseella, että ika-muuttujaan tallennetun henkilön iän pitää olla aina yhtä suuri tai suurempi kuin nolla:

```
assert (ika >= 0);
```

Toinen tapa assert-lauseen käyttöön on antaa boolean-lauseen lisäksi jokin informatiivinen viesti tai virheilmoitus (`String`), joka liitetään poikkeukseen siinä tapauksessa, että assert-tarkistus ei mene läpi. Tämä viesti voi olla suoraan merkkijono, tai se voi olla esimerkiksi metodikutsu, joka palauttaa merkkijonon. Syntaksi on tällöin

```
assert booleanlause : virheilmoitus;
```

Esimerkki

Ilmoitetaan assert-lauseella, että ika-muuttujaan tallennetun henkilön iän pitää olla aina yhtä suuri tai suurempi kuin nolla:

```
assert (ikä >= 0) : "Ikä ei voi olla pienempi kuin nolla.";
```

Jos assert-tarkistus ei mene läpi (ts. boolean-lause palauttaa arvon false), syntyy poikkeus `java.lang.AssertionError`. Mikäli assert-lauseessa on annettu virheilmoitus, tulee tämä virheilmoitus poikkeuksen ilmoitukseksi.

Esimerkki

Ohjelma, joka olettaa saavansa aina vähintään yhden parametrin.

```
public class ParametriTarkistus {  
    public static void main(String args[]) {  
        assert args.length == 0 : "Parametri puuttuu.";  
        System.out.println(args.length);  
    }  
}
```

Jos ohjelmaa kutsuttaisiin ilman parametreja, syntyisi poikkeus ja konsoliin tulostuisi teksti

```
Exception in thread "main" java.lang.AssertionError: Parametri puuttuu  
    at ParametriTarkistus.main(ParametriTarkistus.java:3)
```

Käytännössä assert-lausetta ei kuitenkaan tulisi käyttää tällaiseen tarkistukseen, koska assert-lauseet eivät ole oletusarvoisesti päällä. Parametrien määrän tarkistus tulisi tehdä tässä yksittäisessä tapauksessa mieluummin if-lauseella.