

Harjoitustehtäviä: Testauksen suunnittelua

1 Toisen asteen yhtälön juuret

Tutkitaan lukiomatematiikasta tuttua toisen asteen yhtälön $ax^2 + bx + c = 0$ ratkaisua. Toisen asteen yhtälön ratkaisukaavan on $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

Yhtälön ratkaisujen eli juurten määrä (0, 1 tai 2 kpl) riippuu tällöin neliöjuuren alla olevan juurrettavan eli ns. diskriminantin arvosta. Oheinen metodi `juuria(a,b,c)` laskee toisen asteen yhtälön juurten lukumäärän:

```
public int juuria(int a, int b, int c) {
    int juuria;
    if (b*b-4*a*c > 0) {
        juuria = 2;
    }
    else if (b*b-4*a*c == 0) {
        juuria = 1;
    }
    else {
        juuria = 0;
    }
    return juuria;
}
```

1.1 Haaraumakattavuus

Suunnittele metodille haaraumakattavat testitapaukset: testien tuloksena pitää osua kaikille koodiriveille, joten erityisesti kaikista ehtolauseista pitää testata molemmat vaihtoehdot. Yksi testitapaus on yksittäinen syötekombinaatio $\{a, b, c\}$. Montako erillistä testitapausta tässä tilanteessa tarvitaan haaraumakattavuuden aikaansaamiseksi?

1.2 Toiminnallinen testaus ja ositus

Suunnittele metodille toiminnallinen testaus. Koska metodia ei käytännössä ole järkevää testata kaikilla $a:n$, $b:n$ ja $c:n$ arvoilla, pitää kaikkien mahdollisten testisyötteiden joukko osittaa eri kategorioihin, joista valitaan vain muutama esimerkkiarvo.

Ositus pitää tehdä siten, että samankaltaisiin tuloksiin oletetusti johtavat syötteet niputetaan samaan kategoriaan. Tässä esimerkissä ositus kannattaa tehdä 2. asteen yhtälön ratkaisukaavan perusteella. Mieti siis, mitkä kohdat ratkaisukaavassa vaikuttavat 2. asteen yhtälön käyttäytymiseen eniten ("kriittiset pisteet"). Miten syötteet a, b, c voidaan jakaa eri kategorioihin kriittisten pisteiden mukaisesti? Laadi tämän osituksen mukaiset testitapaukset: kustakin kategoriasta eli syötejoukosta yksi edustaja.

2 Henkilötunnuksen syöttö

Oheinen metodi `haeHenkilo()` hakee tietokannasta parametrina annettua henkilötunnusta vastaavan Henkilö-olion, jos sellainen löytyy.

```
public Henkilo haeHenkilo(String hetu);
```

Suunnittele metodille toiminnallinen testaus. Emme tiedä metodin toteutuksen yksityiskohtia, vain sen että se ottaa syötteenä henkilötunnuksen ja palauttaa henkilötunnukseen liittyvän Henkilö-olion.

Mitä asioita ja millaisia tilanteita pitäisi osituksessa ottaa huomioon? Ryhmittele tilanteet sopiviin kategorioihin ja hahmottele kuhunkin kategoriaan testitapaus.

Huom. Henkilötunnus muodostuu merkkijonosta **xxxxxxYzzzQ**, missä

- **xxxxxx** on syntymäaika kuudella merkillä, esim. 291185 == 29.11.1985
- **Y** on sen vuosisadan tunnus, jolloin henkilö on syntynyt:
 - 1800-luku: +
 - 1900-luku: -
 - 2000-luku: A
- **zzz** on yksilönumero väliltä 002-899, jolla samana päivänä syntyneet yksilöidään. Naisilla parillinen, miehillä pariton
- **Q** on tarkistusmerkki, joka lasketaan jakamalla syntymäajan ja yksilönumeron muodostama 9-numeroinen luku 31:llä. Jakojäännös määrittää tarkistusmerkin, esim. jakojäännös 0 -> 0, jakojäännös 10 -> A, jakojäännös 28 -> W... (kaikki aakkoset eivät ole käytössä)