

# Testausdokumentti

## Labyrintti-ohjelman testaustilanne 12.1.2014 klo: 23:59

Olen käyttänyt Labyrintti-ohjelman testauksessa pääasiassa JUnit -testejä sekä 'käsini testausta' pääohjelmametodista käsin. Yksikkötestien kirjoittaminen on minulla vielä hieman kesken (mikä ei tietysti ole kovin TDD-ihanteiden mukaista), ja tiedostan, että jotkin rajatapaukset on minulla vielä koodissa huomioimatta (esim. Pino-toteutuksessa). Ohjelma ja tietorakenteet toimivat kyllä yhdessä nykyisellään, mutta poikkeustilanteet huomioimalla voisin varmistaa tietorakenteiden toimivuuden myös mahdollisessa muussa myöhemmässä käytössä. Testauksessa on vielä parannettavaa myös esimerkiksi siinä mielessä, että yhdessä testissä tulee nyt monesti käytettyä useampaa kuin yhtä metodia, ja toisaalta jotkut testit ovat ehkä vähän turhiakin. Kaiken kaikkiaan JUnit -testien kirjoittaminen on minulle vielä melko uutta, joten järkevien testien toteuttamisessa on varmasti vielä paljon opittavaa.

Esimerkiksi sellaisia metodeja, jotka vain tulostavat jotakin (esim. keossa olevat alkiot), en ole testannut, enkä myöskään yksityisiä metodeja, joita jokin julkinen metodi käyttää. Luokka Astar, joka hyödyntää Javan valmista kalustoa, on vielä testaamatta.

Tällä hetkellä olen testannut luokkien metodeja seuraavasti:

### Astar:

- searchOmallaKeolla(Labyrintti labyrintti, Solmu lahto, Solmu maali)
  - searchJavanPriorityQueueella(Labyrintti labyrintti, Solmu lahto, Solmu maali)
- Yllä mainittujen testaus sisältää tällä hetkellä lyhimmän polun rekonstruoinnin

### Keko:

- Keko(int koko)
- lisaaKekoon(Solmu solmu)
- vanhempi(int i)
- vasen(int i)
- oikea(int i)
- poistaPienin()
- isEmpty()
- contains()

### Labyrintti:

- getSolmu(int x, int y)
- getLahto()
- getMaali()
- getNaapurit(Solmu s)
- getNaapurit2(Solmu s)
- etaisyysValilla(Solmu nykyinen, Solmu naapuri)

### Pino:

- push(Solmu s)
- pop()
- empty()
- full()

### Solmu:

- setMatkaAlkuun(int alku)
- setKokonaisKustannus(int kustannus)
- setEdellinen(Solmu edellinen)
- getX()
- getY()
- compareTo(Solmu toinen)

## **Suorituskykytestaus**

Suorituskykytestausta olen tehnyt tällä hetkellä lähinnä käynnistämällä kellon, kun olen suorittanut Astarhaun käyttäen kokonaan Javan valmista kalustoa, ainoastaan Javan PriorityQueueeta, tai pelkästään omia tietorakenteita. Tällä hetkellä näyttäisi siltä, että 1000 kierroksella ainakin Astarin toteutus Javan PriorityQueueella ja omalla keolla ovat jollakin tavalla vertailukelpoisia, mutta kuten mainitsin viikkoraportissa, tarvitsen vielä suorituskykytestaukseen ja sen dokumentoimiseen lisää omaa paneutumista ja vinkkejä pajasta.