

NLLR

Testausdokumentti

Leo Leppänen

24. helmikuuta 2014

1 Automaattinen testaus

Ohjelmalle on luotu kattavat JUnit testit, jotka kattavat lähes kaikki tavalliset (ei-rajapinta) luokat. Testejä on arvioitu jatkuvasti erilaisia metriikoita (Kts. alla) hyväksikäyttäen ja niitä on pyritty parantamaan aktiivisesti mahdollisimman hyvän rivi-, polku- ja mutaatiokattavuuden saavuttamiseksi.

Algoritmien osalta algoritmeille on luotu käsin syötteet, joiden oikeelliset tulokset on laskettu muilla menetelmin. Näiden toisin menetelmin hankittujen tulosten vertaaminen algoritmien tuottamiin tuloksiin sopivan pienellä ϵ -arvolla ($\epsilon = 0.00000001$) on merkittävin osa testeistä.

Tietorakenteiden testauksessa on keskitytty kahteen päähaaraan: ulkoisen toiminnallisuuden oikeellisuus sekä sisäisen toiminnallisuuden oikeellisuus. Ulkoisesta toiminnallisuudesta on testattu, että kaikki tietorakenteiden operaatiota kuten *add*, *remove*, *get* ja niin edelleen tuottavat halutun tuloksen.

Sisäisestä toiminnallisuudesta on tarkasteltu erityisesti *HashMap* ja *ArrayList* -tietorakenteiden taustalla olevien taulukoiden koon muuttamista. *HashSet*-tietorakenteen testaaminen on jätetty vähemmälle sen luonteen takia: toteutus on käytännössä pelkkä naamioitu *HashMap*, joten käytännössä samojen

toimintojen tarkistaminen useaan otteeseen koettiin epämielekkääksi ajankäytöksi.

1.1 Metriikat

Projektin testien arviointiin on käytetty PIT:iä ja Coberturaa. Alla on esitetty PIT-testikattavuus 24.02.2014 klo 10:15.

Cobertura-testikattavuuteen voi tutustua seuraavassa osoitteessa:
<https://coveralls.io/r/ljleppan/nllr?branch=master>

Luokan TimeSpan testikattavuus on muiden luokkien testien sivuvaikutusta, sille ei ollut vielä tämän raportin kirjoitushetkellä kirjoitettu testejä. Luokan CommandLineInterface testikattavuus on heikko, sillä pääosa sen testaamisesta on suoritettu manuaalisesti. Luokkan Main testaamista automaattisesti ei pidetty mielekkäänä, vaan se testautuu käyttöliittymän manuaalisen testaamisen oheistuotteena.

Class/Package	Line Coverage		Mutation Coverage	
loez.nllr	0%	0/14	0%	0/8
Main	0%	0/14	0%	0/8
loez.nllr.algorithm	100%	84/84	90%	53/59
Argmax	100%	43/43	84%	31/37
Nllr	100%	26/26	100%	13/13
Tfidf	100%	15/15	100%	9/9
loez.nllr.datastructure	100%	247/248	91%	161/177
ArrayList	100%	73/73	94%	60/64
HashMap	99%	128/129	86%	70/81
HashSet	100%	46/46	97%	31/32
loez.nllr.domain	99%	156/158	71%	105/114
Corpus	99%	76/77	96%	49/51
Document	97%	32/33	77%	17/22
TimeSpan	100%	48/48	95%	39/41
loez.nllr.preprocessor	96%	49/51	88%	15/17
SimplePreprocessor	100%	10/10	100%	2/2
SnowballPreprocessor	95%	39/41	87%	13/15
loez.preprocessor.util	100%	6/6	100%	4/4
Numeral	100%	3/3	100%	2/2
Punctuation	100%	3/3	100%	2/2
loez.nllr.reader	89%	25/28	50%	12/24
CorpusReader	92%	11/12	33%	5/15
DocumentConverter	88%	14/16	78%	7/9
loez.nllr.userinterface	73%	142/195	50%	51/103
CommandLineInterface	73%	142/195	50%	51/103
Class/Package	Line Coverage		Mutation Coverage	

2 Manuaalinen testaus

Käyttöliittymän testaus on suoritettu pääosin käsin. Laajaa ja kattavaa testausta ei ole tehty, sillä käyttöliittymä ei ole osa projektin ydinaluetta. Kaikki tavallisen käytön yhteydessä löydetyt bugit on kuitenkin korjattu. On kuitenkin varsin mahdollista, että käyttöliittymään on jäänyt rajatapauksia, joissa toiminta ei ole toivottavaa.