

Testausdokumentti

Ohjelman testaus

Itse ohjelmaa testataan [JUnit](#) testausrajapinnalla. Kaikki testit löytyvät tarkasteltavaksi projektista test -kansion alta ja ne voi ajaa Netbeanssissa painamalla pikinäppäinyhdistelmää ALT+F6. Testejä projektissa on yhteensä 136.

Dokumentaatiokansiosta löytyy myös automaattisesti generoitu [Cobertura](#) rivikattavuusanalyysi josta näemme, että projekti saavuttaa yli 90% rivitestikattavuuden.

Coverage Report - domain

Package /	# Classes	Line Coverage	
domain	11	92%	199/214
Classes in this Package /		Line Coverage	
AbstractElement		96%	27/28
AbstractElementHashMap		72%	16/22
Category		91%	11/12
Item		100%	20/20
Quality		90%	9/10
Rating		100%	26/26
RecommendedItem		93%	28/30
SimilarUser		88%	23/26
Star		100%	12/12
User		94%	16/17
Value		100%	11/11

Report generated by [Cobertura](#) 2.0.3 on 3.3.2015 14:22.

Algoritmin testaus

Ohjelma on pelkkä testiohjelma, jonka ainut funktio on testata algoritmia automaattisesti kun se käynnistetään. Katso Example.java tiedosto nähdäksesi miten Algorithm() luokkaa käytetään.

Testidatan määrä

Oletuksena testidataa generoidaan 100000 käyttäjää, 40000 tuotetta, 1000 ominaisuutta, 40 kategoriaa ja 0-65 arvostelua per käyttäjä, eli noin 3.2 miljoonaa arvostelua.

Testaaminen

Testaaminen tapahtuu kahdessa vaiheessa, ensin ohjelma luo testidataa algoritmin käytettäväksi luokassa TestDataGenerator(). Tämä prosessi vie tavallisesti noin 20-27 sekuntia. Tämän jälkeen Example() luokka suorittaa algoritmin testidatalle. Example() luokka ottaa esimerkkitapaukseksi käyttäjän, jonka id on 2000 ja hakee tälle käyttäjälle suositukset. Kahdestuhannes käyttäjä voi olla millainen tahansa, jopa henkilö, jolla ei ole yhtään arvostelua. Itse algoritmin suoritus vie yleensä 400-1500ms (harvoin yli sekunti jos testidataa on oletusmäärä).

Testien tulkinta

Ohessa on leikkaus esimerkkitulosteesta, jonka ohjelma tuottaa ja selityksenä sen eri osat:

```
----- STARTING -----
0ms: Creating 100000 users.
100ms: Creating 40000 items.
153ms: Creating 1000 qualities.
5176ms: Creating 40 categories.
5176ms: Giving items their qualities (5-15 qualities per item).
5308ms: Giving items (every item has a category) their categories.
5643ms: Giving items ratings from users (0-65 per user).
26150ms: Finished giving 3254709 ratings from users.
26150ms: Creation process is now done. Overall it took 26 seconds.
----- ALGORITHM -----
1ms: Interesting users found: 1439
16ms: Interesting items found: 31290
445ms: Similarity between 1439 users defined.
446ms: 145 most similar users users were chosen.
----- TOP 10 ITEMS -----
125.0: Item number: 22185 (created at 129ms) Q:9 R:89 C:1
    1.5: Item number: 1339 (created at 102ms) Q:7 R:77 C:1
    1.5: Item number: 2101 (created at 103ms) Q:9 R:93 C:1
    1.5: Item number: 5575 (created at 104ms) Q:7 R:76 C:1
    1.5: Item number: 7380 (created at 105ms) Q:9 R:104 C:1
    1.5: Item number: 7651 (created at 105ms) Q:6 R:84 C:1
108.0: Item number: 27177 (created at 131ms) Q:9 R:81 C:1
    1.5: Item number: 90 (created at 100ms) Q:8 R:76 C:1
    1.5: Item number: 246 (created at 100ms) Q:7 R:76 C:1
    1.5: Item number: 311 (created at 101ms) Q:8 R:71 C:1
    1.5: Item number: 323 (created at 101ms) Q:9 R:82 C:1
    1.5: Item number: 647 (created at 102ms) Q:7 R:74 C:1
108.0: Item number: 25203 (created at 130ms) Q:9 R:78 C:1
    1.5: Item number: 376 (created at 101ms) Q:9 R:76 C:1
    1.5: Item number: 1738 (created at 102ms) Q:10 R:97 C:1
    1.5: Item number: 1804 (created at 102ms) Q:11 R:83 C:1
    1.5: Item number: 2379 (created at 103ms) Q:8 R:91 C:1
    1.5: Item number: 2462 (created at 103ms) Q:7 R:79 C:1

... (TULOSTE JATKUU)
```

Tuloste on jaettu kolmeen eri osaan: Starting, Algorithm ja Top 10 items -vaiheisiin. Starting vaiheessa luodaan testidataa. Itse algoritmin kannalta tämän suoritustehokkuus ei ole merkityksellistä, mutta yleensä se antaa hyvän kuvan siitä, kuinka paljon dataa algoritmille tuotetaan. Mikäli vaikuttaa siltä, että algoritmi on pysähtynyt vaiheeseen **Giving items ratings from users** voi tämä tarkoittaa, että sovellukseltasi on loppunut käytettävä muisti. Tämä todennäköisesti korjaantuu antamalla [JVM:lle enemmän muistia](#) käytettäväksi.

Toisessa vaiheessa suoritetaan itse algoritmi. Tässä tulosteiden aikaleimat ovat oikeasti merkityksellisiä. Saadaksesi paremmin selville algoritmin tulosteet katso *toteutusdokumentti*, jossa kerrotaan tarkemmin algoritmin toiminnasta.

Viimeisessä vaiheessa tulostetaan TOP-10 lista eniten käyttäjälle suositelluista tuotteista. Lista on kaksiosainen. Ensimmäinen tuloste on suositeltu tuote ja sen alla sarkaimella erotetut 5 tuotetta ovat `Example()` luokan algoritmilta pyytämiä ”samankaltaisia tuotteita”.

Ohjelman tulosteeseen erikseen merkatut värikoodit tarkoittavat seuraavia:

- **Oranssi**: Tuotteen suositteluarvo
- **Punainen**: Tuotteen nimi (esimerkitapauksessa tuotteen id)
- **Keltainen**: Tuotteen kuvaus (esimerkitapauksessa luonnin hetki)
- **Sininen**: Tuotteen sisältämien ominaisuuksien (Q, Quality), arvostelujen (R, Rating) ja kategorioiden (C, Category) määrä.