

# Käyttöohje

## Tietorakenteiden harjoitustyö

Joosua Laakso

11. tammikuuta 2014

## 1 Tiedostot

Ohjelman suorittamiseen tarvittava .jar-tiedosto löytyy sijainnista "tiralabra/dist/tiralabra.jar". Ohjelmaan liittyvät testit löytyvät kansioista "tiralabra/test". Sijainnista "tiralabra/src/regex/evaluate/Evaluator.java" löytyvä tiedosto sisältää Evaluator-luokan, jonka avulla käyttäjä voi käyttää tätä säännöllisten lausekkeiden toteutusta omassa Java-projektissa.

## 2 Ohjelman käyttäminen

### 2.1 Säännöllisten lausekkeiden kielioppi

Tämä ohje on suunnattu lukijalle joka tuntee säännöllisten lausekkeiden perustoiminnan entuudestaan. Tämä toteutus tukee seuraavia ominaisuuksia:

**Tavalliset merkit:** Esimerkiksi "a", "b", "c".

**Katenointi:** " $re_1re_2$ ". Esimerkiksi "ab", "abc", "xyz".

**Jokerimerkki:** Piste, ".", hyväksyy kohdallaan minkä tahansa merkin.

**Unionioperaattori:** " $re_1|re_2$ ". "|" -operaattorin sisältävä lauseke hyväksyy merkkijonon, jos joko operaattorin oikeanpuoleinen lauseke tai vasemmanpuoleinen lauseke hyväksyy merkkijonon. Esimerkiksi " $a|b$ ", hyväksyy merkkijonot "a" ja "b", mutta ei esimerkiksi merkkijonoa "ab".

**Tähtioperaattori:** " $re_1^*$ ". Hyväksyy merkkijonon, jossa tähden vasemmalla puolella oleva lauseke toistuu usean kerran tai puuttuu kokonaan.

**Plus-operaattori:** " $re_1^+$ ". Sama kuin tähti, mutta vasemmalla oleva lauseke ei saa puuttua.

**Kysymysmerkki-operaattori:** " $re_1^?$ ". Hyväksyy merkkijonon, vaikka vasemmalla puolella oleva lauseke puuttuisi.

**Hakasulut:** Hakasulut, “[ ]”, sisältävät merkkejä, ja hyväksyvät merkkijonon jos hakasulun kohdalla on merkkijonossa jokin hakasulkujen sisällä olevista merkeistä. Hakasulkujen sisällä voi käyttää kahden merkin välillä viivaa, “-”, merkitsemään sitä, että hakasulkujen hyväksymä merkki voi olla jokin merkki viivan vasemmalla puolella olevan merkin ja oikealla puolella olevan merkin väliltä. Esimerkiksi lauseke “a[b-dx]y” hyväksyy merkkijonot “aby”, “acy”, “ady” ja “axy”.

**Aaltosulut:** Aaltosulut, “re<sub>1</sub>{n}”, hyväksyvät merkkijonon, jossa vasemmalla puolella oleva lauseke toistuu hakasulkujen sisällä olevan luvun verran. Hakasulkujen sisällä voi käyttää viivaa, “-”, erottamaan kaksi lukua, jolloin vasemmalla puolella oleva lauseke voi toistua viivan vasemmalla puolella olevan luvun verran, viivan oikealla puolella olevan luvun verran tai jonkin näiden kahden luvun välillä olevan määrän verran. Esimerkiksi säännöllinen lauseke “a{3-5}b” hyväksyy merkkijonot “aaab”, “aaaab” ja “aaaaab”.

## 2.2 Evaluator-luokan käyttäminen

Evaluator-luokkaa käytetään kahden metodin avulla. Evaluator-luokasta luodulle oliolle voi ladata säännöllisen lausekkeen loadRegex-metodin avulla, joka suorittaa myös lausekkeen validoinnin, eli tarkistaa, onko lauseke syntaktisesti korrekti säännöllinen lauseke. Jos käyttäjä haluaa, että validointia ei suoriteta, käyttäjä voi joko antaa arvioitavan säännöllisen lausekkeen konstruktoria parametrina, tai antaa konstruktoria parametrina null-validaattoria, jolloin validointia ei myöskään suoriteta edes loadRegex-metodin suorituksen yhteydessä. Toinen metodi, jota tarvitaan on evaluateString, joka palauttaa boolean-arvon, joka kertoo, hyväksyykö vai hylkääkö säännöllinen lauseke parametrina annetun merkkijonon.

## 2.3 .jar-tiedoston käyttäminen

.jar-tiedostoa voi käyttää komentoriviltä kirjoittamalla komentoriville “java -jar <.jar-tiedoston sijainti> <parametrit>” ja kirjoittamalla kohdan <.jar-tiedoston sijainti> tilalle .jar tiedoston sijainnin ja kohdan <parametrit> tilalle ohjelman parametrit.

### 2.3.1 Komentoriviparametrit

Parametrit annetaan ohjelmalle välilyönnein eroteltuina. Jos haluat käyttää parametrissa itsessään välilyöntiä tai muuta varattua merkkiä kuten pystyviivaa, parametri täytyy antaa lainausmerkeissä. Tässä ovat lueteltuina ohjelman hyväksymät parametrit:

<lauseke> <syöte> Ohjelma tulostaa “true” tai “false”, riippuen siitä, hyväksyykö lauseke syötteen.

-i <lauseke> Interaktiivinen tila; ohjelmalle voi antaa arvioitavaksi merkkijonoja rivinvaihdolla eroteltuina, ohjelma tulostaa kunkin syötteen kohdalla “true” tai “false”, riippuen siitä, hyväksyykö lauseke syötteen.

**-f** <lauseke> <tiedoston sijainti> Ohjelma lukee syötteen annetusta tiedostosijainnista ja tulostaa "true" tai "false", riippuen siitä, hyväksyykö lauseke tiedoston sisältämän syötteen.

**-g** <lauseke> <tiedoston sijainti> "Grep"-tila; ohjelma lukee syötteen annetusta tiedostosijainnista rivi riviltä ja tulostaa jokaisen rivin, jonka annettu lauseke hyväksyy.