

Name:

Matr.Nr.:

Gruppe: 08:30 10:15

Abgabefrist: 10.05.2010

Abzugeben: Quellcode elektronisch.

Tutor:

Punkte:

Maximal: 24 Punkte

Networking, Databases

Implement a minimalistic search engine (*Mist*). *Mist* should consist of a crawler to find web pages and an indexer to put the found web pages into a database.

Crawler

The crawler should be configurable with an initial url, a maximum recursion depth, and an url filter where the crawler should only follow links containing the given pattern. Hints: a regular expression can be used to find the urls on a web page; plain *Sockets*, or *URL* and *URLConnection* can be used to load the pages.

Configuration of the crawler:

- Initial url: a String, e.g. <http://ssw.jku.at>
- Depth: an int, e.g. 0 only the initial site, 1 also the sites linked from the initial site, and so on
- Url filter: a regular expression Pattern, e.g. `Pattern.compile("jku.at")`

Indexer

The indexer should find all words on a page, enter these words in a database, add the url of the page into the database, and add a relation between the word and the url into the database.

Configuration of the indexer:

- Database: a Connection to the target database

Database

As database JavaDB (Derby) should be used. Derby can be used in an embedded mode and a server mode. Feel free the use it as you please; info: if the database runs in embedded mode only one connection can be established.

The database for *Mist* must contain the tables: *urls*, *words*, and *relations*. The table *urls* has the columns: ID as INTEGER and URL as VARCHAR(1000). The table *words* has the columns: ID as INTEGER and WORD as VARCHAR(1000). The table *relations* has the columns: URLID as INTEGER and WORDID as INTEGER.

The tables can be created with the statements:

```
create table urls (id INTEGER PRIMARY KEY GENERATED ALWAYS AS IDENTITY, url VARCHAR(1000));
create table words (id INTEGER PRIMARY KEY GENERATED ALWAYS AS IDENTITY, word VARCHAR(1000));
create table relations (urlid INTEGER, wordid INTEGER);
```

Every url and every word should only be once in the database. A relation between a word and url should as well be only once in the database, even if the word is many times on the page.