Hochschule für
Technik und Wirtschaft
des Saarlandes
University of
Applied Sciences

Information Retrieval
Winter 2019/2020

Prof. Dr.-Ing. Klaus Berberich
Telefon: 06 81 58 67-243
klaus.berberich@htwsaar.de

Programming Assignment 1

The programming assignment will be discussed on **November 14**. To obtain bonus points, you have to submit your solution via Moodle by **November 12 at 12:00 (noon)**. Please submit your solution, consisting of source code files and possibly libraries, **as one zip archive**. Teams of up to three students are allowed.

Over the course of the semester you will implement a small search engine through the programming assignments. As a document collection we will use The New York Times Annotated Corpus[*], which contains articles published in the newspaper between 1987 and 2007. In this first programming assignment, you will familiarize yourself with the document collection and write some code to read and tokenize documents.

Please use an IDE such as Eclipse or IntelliJ for writing your code. Organize your project in such a way that there are separate directories for source code (e.g., `src`), libraries (e.g., `lib`), and compiled code (e.g., `build`).

1.1 Document Collection (0.5 Points)

Download the document collection (`nyt.zip`) from Moodle. Unzip the archive and read the documentation available in the directory `docs`. There you will learn about the structure of the documents and which fields are available.

Implement a class `Document` to represent documents. The class should have the following member attributes and provide suitable getter and setter methods for modifying them:

- `long id; // Document identifier`
- `String title; // Document title`
- `String url; // Document URL`
- `String[] content; // Document content`

[*] https://catalog.ldc.upenn.edu/LDC2008T19

1.2 Uncompressing Archives (0.5 Points)

The document collection comes with one `.tar.gz` archive for every month. Use the `tar` command to uncompress these archives. Once you've done that, you should have twelve directories (i.e., `01-12`), each of which contains all articles published in the corresponding month of 2000 as XML files.

Implement a class `Importer`. The class should provide a method

- `void import(File file)`

The method is provided with a directory (e.g., the one where you put the document collection). It should recursively traverse the directory and collect all files with the extension `.xml`. For each file found, please output the complete file name and the file size to the standard output.

1.3 Parsing Documents (1 Point)

The document collection comes with a class `NYTCorpusDocumentParser` that provides a parser for the XML documents. You can find the class in the directory `tools`. Using this parser, implement a class `Parser` that provides a method

- `Document parse(File file)`

The method is provided with a XML file and should return an instance of `Document` (cf. 1.1) with meaningful values for the member attributes `id`, `title`, and `url`.

1.4 Tokenization (1.5 Points)

Next you will implement a simple tokenization strategy to fill the member attribute `content` with meaningful terms. To this end, please use the `Body` field of the XML document and proceed as follows:

- remove HTML commands (e.g., `<P>`) that might still be present
- replace all characters that are neither a letter (e.g., `a-z`), a number (i.e., `0-9`) or a full stop (i.e., `.`) by white spaces
- devise a strategy (e.g., as a regular expression) to decide which full stops should be replaced by white spaces (e.g., if they mark the end of a sentence) and which should not (e.g., because they are part of an abbreviation)
- convert all characters to lower case
- split the resulting character sequence at sequences of white spaces

The member attribute `content` should now contain a sequence of terms, which we can index in the next programming assignment.