1. Exercises in Software Architectures: a Pipes&Filters Architecture

(6 credits)

a) Design and implement an ASCII-based word processing application, with a Pipes&Filters architecture, which generates a simple e-book index, called for convenience INDSYS, and implement and test it:
The index system INDSYS accepts a list of lines (the e-book does not know the concept of a page); each line is a list of words, and each word is a list of characters. Any line may be circularly shifted by repeatedly removing the first word and appending it to the end of the line. INDSYS outputs a list of all circular shifts of all lines in alphabetical order, together with the corresponding line number. Take the following example:

Kiwis are good. (line 12) Apples are better. (line 13)

The first line, for example, implies three circular shifts: "kiwis are good", "are good kiwis", "good kiwis are". INDSys would generate the following entries in the index out of these two lines:
apples are better 13
are good kiwis 12
are better apples 13
better apples are 13
good kiwis are 12
kiwis are good 12
The following material is already available on ILIAS, and you are supposed to start design and implementation of INDSYS from this starting point:
- the framework "PimpMyPipe" for a general Pipe&Filters architecture: pimpmypipe.zip

The application starts with a source filter reading in lines as a whole. The next filter transforms a line into a sequence of words. The next filter creates the circular shifts. The next filter then orders the circular shifts alphabetically. Finally, the last filter writes the index to file specified by an attribute of that filter.
Try to find a solution to the problem, that some words are useless in an index, such as "are", "the", "and" etc., that means words which cannot be considered as a keyword worth looking for: we do not want to have these words in our index (as first words of a circular shift). You may use the file with frequent words on the Ilias. Implement this solution.
Test your application with some ASCII-formated book (for example Alice in Wonderland, see the textfile on the Ilias platform).
b) extend INDSYS such that you start with a character stream, then a filter constructs words, then the next filter generates lines of a desired length and of a desired alignment (left, centered, right), then the pipeline splits with the first sub-pipeline running like that of a) generating an index, and the second sub-pipeline writes the lines to a file defined by an attribute of that sink filter.

Due date: 30.10.2017, 9h50
Deliverables:

1. Design documentation (UML class diagram, package diagram, sequence diagram).
2. Description of how you solved the problem of suppressing uninteresting words in the index.
3. source code
4. an executable that can be started from the console