The **RecSPL input file**, which your Lexer software must analyse, will be given as a plain ***.txt** file; for the purpose of experimenting and testing you can easily create such ***.txt** files (containing some RecSPL program code) by yourself.

How you implement your Lexer - in which programming language, and with which methods and techniques - is *entirely up to your own choice,* as long as only the following **requirements** are met:

- IF the input ***.txt** file contains any **lexical errors** (which corresponds, in theory, to the underlying DFA getting 'stuck' in a non-accepting state), then your **Lexer** software must "throw" a reasonably understandable **Error-Message** back to the User.
- IF the input ***.txt** file does <u>not</u> contain any lexical errors, then your **Lexer** software must **create**, **write**, and **store** (as its output) an **XLM file** which the Parser can later use as its input.

In this manner, you can keep your implementations of Lexer and Parser completely separated from each other in different "phases" of your semester-project, because **the persistent XML file will serve as an *"offline bridge"* between Lexer and Parser.** This design approach is very "convenient" and will make your "project life" much easier (in comparison against having to implement one huge all-in-all Compiler software). *Thus you'll even be able to switch your computer off and on again between Lexing and Parsing* **:)**

Thereby, the "tokenized" **contents of the XML** file shall be structured as follows:

\<TOKENSTREAM\>

\<TOK\>

\<ID\>**1**\</ID\> *// comment: Each token has its own unique ID number*

\<CLASS\>*the token's class*\</CLASS\> *// comment: See an example for illustration below*

\<WORD\>**<u>terminal characters for the Parser</u>**\</WORD\> *// comment: See example below*

\</TOK\>

\<TOK\>

\<ID\>**2**\</ID\>

\<CLASS\>*the token's class*\</CLASS\>

\<WORD\>**<u>terminal characters for the Parser</u>**\</WORD\>

\</TOK\>

... etc ...

<TOK>

... etc ...

</TOK>

</TOKENSTREAM>

EXAMPLES:

<TOK>

<ID>**126**</ID>

<CLASS>**reserved_keyword**<CLASS> *// comment: The class corresponds to some **Accept-State** of the **DFA***

<WORD>**else**</WORD>

</TOK>

...

<TOK>

<ID>**496**</ID>

<CLASS>**N**<CLASS> *// comment: That is the token-class for **N**umbers, as per given Specification Sheet*

<WORD>**56.7**</WORD>

</TOK>