Homework 1

Submission format will be a single zip file, containing 4 maven project directories. Project directories must be **cleaned** before zipping, e.g. remove target folder from submission, etc. Each source file must be formatted using IntelliJ code formatting. All naming must follow the naming convention taught in the lecture. You lose points for any failure to meet the instruction. The one of the goals is to give you a practice on the best practice in large scale application development.

1. [100 points] In this problem, you have to download a Java SE Development Kit 8u211 Documentation distribution (zip file) and extract it to a directory. You **MUST NOT** include this folder in your submission file. [http://www.oracle.com/technetwork/java/javase/documentation/jdk8-doc-downloads-2133158.html]

   Create a maven project to walk into this directory. You may hardcode the path to documentation directory you extracted in your source code. The program should print out to console the following:
   a. Total number of files.
   b. Total number of directory
   c. Total number of unique file extensions.
   d. List all unique file extensions. Do not list duplicates.
   e. Total number of files for each extension.

   Hint: use DirectoryWalker and FileFilter in Apache Common IO.

2. [100 points] In this problem, you will make a duplicate of the previous project, extend the program to take the following command line arguments:
   a. **-a, --total-num-files** The total number of files
   b. **-b, --total-num-dirs** Total number of directory
   c. **-c, --total-unique-exts** Total number of unique file extensions.
   d. **-d, --list-exts** List all unique file extensions. Do not list duplicates.
   e. **--num-ext=EXT** List total number of file for specified extension EXT.
   f. **-f=path-to-folder** Path to the documentation folder. This is a required argument.

   Your application should support multiple arguments and in any order, e.g. **-a -b** should does the same thing as **-b -a**. Hint: use Apache Commons CLI.

3. [100 points] Create a maven project for downloading a URL content and write to file using 3 unique methods. Hint: use URLConnection in standard library, and looks up in Apache Common. Apache Http Component is a recommended library because you can keep the connection alive and reuse it in the problem 4.

4. [200 points] In this problem, you will download an entire Java docs by crawling the URL here: https://cs.muic.mahidol.ac.th/courses/ooc/docs/

   Make sure you do not crawl any links outside this domain. Complete downloaded Java docs should contain the same number of files as in the first problem. **Do not use crawler library**. The final output of this program should print the total number words in all html files, excluding all html tags, attributes and html comments. Hint: There are Html parser called jsoup.