PCD1718 - Assignment #2

Design and implement a tool for searching matches of a regexp in a tree or graph structure of files such as a filesystem or a website.

The tool must exhibit the following UI:

- Inputs: the base path of the search (e.g., a filesystem path or a webpage URL); properties of the traversal (e.g., max depth).
- Outputs: the list of matching files, the % of files with at least one matching, and the mean number of matches among files with matches.

The outputs must be updated in real-time as processing proceeds.

Exercise 1: solve the problem using tasks and executors
- The files should be read and analysed concurrently.
- One may also opt for parallelising the analysis of big files.

Exercise 2: solve the problem using asynchronous programming in the event loop
- Try to reuse as much code as possible from exercise 1, but also rethink at the solution according to the new viewpoint.

Exercise 3: complement your solution to use reactive streams
- E.g. the processing results can be reified into an event stream that can be manipulated through reactive programming techniques.

Further notes:
- The UI may be command line-, graphical-, or web-based.
- You are free to use any event-driven framework (e.g.,Vertx, NodeJS) and reactive stream library (e.g., RxJava, Sodium).

References:
- NodeJS filesystem API: https://nodejs.org/api/fs.html
- NodeJS HTTP API: https://nodejs.org/api/http.html
- NodeJS child processes: https://nodejs.org/api/child_process.html
- Vertx blocking code how-to: https://vertx.io/docs/vertx-core/java/#blocking_code
- Vertx filesystem: https://vertx.io/docs/vertx-core/java/#_using_the_file_system_with_vert_x
- Vertx HTTP how-to https://vertx.io/docs/vertx-core/java/#_making_requests