

Programming Exercise

The goal of this programming exercise is to give us insight in your coding level and style. Try to make your code well structured. Pay attention to:

- SOLID design principles
- Software design patterns
- Clean, readable code
- Unit tests for the critical sections of code

Be able to explain and defend design decisions.

Use case

We regularly receive documents from an external system: weekly status reports, monthly overviews, detailed reports. Sometimes, these documents are plain text files, and sometimes they are xml files. We want to upload these documents to our database. However, before we upload them, we want to search and replace certain text phrases in the documents.

Currently, we only want to search for specific text phrases, but we expect that we will want to be able to use a more flexible search in the future; patterns or wildcards or variables – we have not decided yet. We might also want to be able to perform multiple search / replace actions at once.

Exercise

Design and code a program that will allow us to search and replace text phrases in regular text documents and xml documents.

- The program should be easy to maintain and extend, for instance to add a more flexible search method, or other file types.
- Use a modern programming language, like Java or C#.
- The program should be able to handle large files – think several gigabytes of data.
- The program should use a predictable, limited amount of memory – think a few hundred megabytes of memory.
- The input xml is always well formed. The output xml should also always be well formed.
- In case of xml, at least the following tokens should be subject to search and replace: element names, attribute values, text nodes.
- The program is not required to have any graphical user interface.
- It is okay to handle errors (e.g. in user input) by throwing exceptions.
- It is okay to determine file type by extension (txt or xml).
- You may assume the data is plain ASCII.