XML and modern techniques of content management 2010/11

Task 3 (make up)

The story is based on the previous task 3. However you should notice some changes in schemas and that the **schemat_B_2010P_Zad3XML.xsd** defines an answer sheet given by a tested person (not a test).

The story

Implement a computer program aimed to help at learning of foreign languages. The input of the program are three XML data files:

1. File A.xml (input), a dictionary which contains set of words: entries from one language and glosses, describing their meanings (possible in various languages). Please refer to the scheme **schemat_A_2010P_Zad3XML.xsd** for more details. Please note that you can assume that the file is small (can fit in memory) and that it does not contain duplicate entries. You cannot assume, however, that the file is sorted.

2. File B.xml (input/output). The total results of previous tests for selected words. The file contains two kinds of information: how many times a word has been tested (TestNumber) and how many answers were correct (SuccessNumber). It can be assumed that the file is small and that word entries are placed in the same order as in the dictionary (in particular, that the corresponding entries in the dictionary exist.). The structure of the file is defined by scheme **schemat_B_2010P_Zad3XML.xsd**. Please note that the languages are set for the whole file. TestNumber and SuccessNumber are updated by the program (they might be increased by 2), the order of the dictionary should be preserved by the program.

3. File C.xml (input). Answers, which is consistent with schema given in file **schemat_C_2010P_Zad3.xsd** (please refer to the schema for more details). The file contains two sets of questions:

   1. The list of answers to multiple choice questions each asking on meaning of one word, there are four possible answers (one or zero indicated by a tested person as the true one).

   2. The list of word descriptions given by a tested person.

The answers to open questions might be long, so the file C.xml cannot be assumed to fit into memory.

The program should count how many answers in the file C.xml are correct and update file B.xml accordingly. If the file B.xml does not exists it should be created

If no answer will be given in a multiple choice question (there is no element trueNumber), then the answer is counted as true if and only if no given option is true (matches a right description in the dictionary). The number of tests is increased in both cases.

If the element trueNumber exists, then the successNumber is increased if and only if the pointed answer is correct.

The answers to the open questions are correct if and only if they match the descriptions in the dictionary. However small and big letters should be treated as equal, same as all white characters, multiple white characters should be treated as one, and punctuation characters should be skipped/substituted by a white character).

Program call parameters

Another program call parameters mean:

1. A path to the dictionary, containing files A.xml, [B.xml], C.xml and schemas.

Remarks and additional requirements

1. The program should be written in Java; It will be compiled and run in the environment Sun Java SE 6.
2. Class containing the main method is located in the default package, and bear the name of TestSprawdzacz.
3. The depth of catalogue tree delivered should not exceed 3.
4. File C.xml cannot be read into memory (in particular, the program will be tested with a large input and limited memory), you should use SAX and STAX.
5. DOM or JAXB API should also be used.
6. The program might read the input files twice.
7. Method of processing files is up to the author's decision (but look above).
8. Program should handle namespaces, output file B.xml should be compatible with the schema . **schemat_B_2010P_Zad3XML.xsd**
9. Source code files should use UTF-8.
10. Errors should be detected and reported on the console (e.g. The program should verify that attributes like languageOfEntry, languageOfDescription i languageOfDescription in files A.xml, B.xml and C.xml are consistent.
11. The program should check existence and validity of A.xml, [B.xml] and C.xml.
12. The program should compile and run correctly at least on students.mimuw.edu.pl server.
13. The solution should be sent by 26 February 2011 (17:59) at the address kedar@mimuw.edu.pl, the title of the email should be "XML 2010, Z3, Solution", attached to the e-mail should be a ZIP- containing a directory named name_surname.
14. Program code should be documented (comprehensively, but without exaggeration), preferably in a way that allows automatic generation of documentation (JavaDoc)). In particular, at the beginning of each file (including README) should contain information about its author (name, surname).
15. An ant script should be provided for quick compilation of the program – a README file must provide appropriate information on the program/script usage.
16. Each started 12 hours of delay means 1 point less.
17. Please check the FAQ.
18. In generated file it should be assumed that the schema is located in the same directory as the generated file.
19. To the solution should be attached a few medium size sample files (A.xml, B.xml, C.xml), a medium size means about 10 entries.