

XML i nowoczesne technologie zarządzania treścią, rok akademicki 2010/11

Zadanie 3 (poprawkowe)

Bajka

Bajka jest kontynuacją zadania 3 w pierwszym terminie, jednak wprowadzono pewne zmiany – w szczególności proszę korzystać ze schematów załączonych do zadania poprawkowego (najważniejsza moim zdaniem zmiana to potraktowanie pliku **C.xml** jako wypełnionych przez studenta odpowiedzi do testu, bez danych o poprawnej odpowiedzi).

Należy zaimplementować program wspomagający naukę języków obcych. Program działa w oparciu o trzy pliki z danymi:

1. Plik wejściowy A.xml (zgodny ze schematem **schemat_A_2010P_Zad3XML.xsd**). Słownik zawierający: hasła (wszystkie z tego samego języka) oraz opisy (z określeniem kategorii, języka opisu). Uwaga: można zakładać, że plik jest mały (zmieści się w pamięci). Nie można zakładać, że jest posortowany. Można zakładać, że nie zawiera duplikatów haseł.
2. Plik B.xml tworzony lub modyfikowany (zgodny ze schematem **schemat_B_2010P_Zad3XML.xsd**): Łączne wyniki użytkownika w konkurencji wprowadzanie **opisu** na podstawie podanego **hasła** oraz w konkurencji wybierania poprawnej definicji spośród czterech podanych. Plik zawiera dla każdego słowa dwie informacje: ile razy słowo było dotychczas testowane (może wzrosnąć o maksymalnie 2), ile razy dotychczas udzielono poprawnej odpowiedzi (może wzrosnąć o maksymalnie 2). Można założyć, że plik jest mały, oraz, że wpisy są w nim umieszczone w takiej samej kolejności jak w słowniku (w szczególności, że odpowiadające wpisy w słowniku istnieją). Wyniki są dla danego słownika i dla opisów w konkretnym języku.
3. Plik wejściowy C.xml (zgodny z **schemat_C_2010P_Zad3.xsd**). Plik zawiera odpowiedzi wprowadzone przez użytkownika programu testującego słówka. W pliku tym na początku ma znajdować się w:
 1. Lista haseł wraz z 4 opisami (w tym jedna wskazana jako poprawna za pomocą trueNumber) – test wyboru.
 2. Lista opisów wprowadzonych przez użytkownika wyżej wymienionych haseł (pytania utrwalające, słowa te same co powyżej) – pytania otwarte.

Odpowiedzi w pytaniach otwartych mogą być długie – nie można zakładać, że plik zmieści się w pamięci. Kolejność słów w obu listach jest dowolna (uwaga na różnicę w stosunku do zadania 3 w pierwszym terminie, gdzie kolejność była zadana przez algorytm).

Stworzony program powinien policzyć ile odpowiedzi w pliku C.xml jest poprawnych i odpowiednio zmodyfikować plik B.xml. W przypadku gdyby plik B.xml nie istniał, ma zostać utworzony nowy plik.

Uwaga: Jeżeli w teście wielokrotnego wyboru nie została zaznaczona żadna odpowiedź (nie ma trueNumber), to odpowiedź zostaje uznana za prawdziwą wtedy i tylko wtedy gdy żadna z opcji nie jest prawdziwa (brak elementu trueNumber). Liczba wskazująca na to ile razy słowo jest testowane jest zwiększana w obu przypadkach.

W przypadku gdy występuje element trueNumber, to sprawdzane jest jedynie czy wskazywana przez ten element odpowiedź jest poprawna.

Odpowiedzi na pytania otwarte są poprawne gdy zgadzają się dosłownie (pomijając wielkość liter, wielokrotne białe znaki traktując jak jeden i ignorując/zastępując białym znakiem znaki interpunkcyjne) z którąś z definicji.

Parametry wywołania programu

Kolejne parametry wywołania programu oznaczają:

- 1) Ścieżka do katalogu w którym znajdują się pliki A.xml, [B.xml], C.xml oraz odpowiednie schematy.

Uwagi i wymagania

1. Program powinien być napisany w Javie, kompilować się i działać w środowisku Sun Java SE 6 (tak będzie testowany).
2. Klasa zawierająca metodę main ma znajdować się w domyślnym pakiecie i nosić nazwę TestSprawdzacz.
3. Głębokość drzewa katalogów dostarczonego przez autora nie powinno przekraczać 3
4. Plik wejściowy C.xml nie może być wczytywane w całości do pamięci (w szczególności program będzie testowany z dużym dokumentem wejściowym i ograniczoną pamięcią), ma zostać użyty **SAX lub StAX**.
5. Interfejs **DOM lub JAXB** ma zostać również użyty.
6. Należy minimalizować liczbę czytań dokumentów wejściowych.
7. Sposób przetwarzania dokumentów zależy od autora rozwiązania (to znaczy, że np. można stosować kolekcje dostępne w Javie, zamiast operować na drzewie DOM).
8. Należy obsługiwać przestrzenie nazw, pliki wynikowe powinny być zgodne ze stanowiącym załącznik do niniejszego zadania schematem **schemat_B_2010P_Zad3.xsd**.
9. Plik(i) źródłowe kodu programu mają używać kodowania UTF-8.
10. Sytuacje błędne powinny być wykryte – a odpowiedni komunikat powinien pojawić się na standardowym wyjściu programu (np. languageOfEntry, languageOfDescription i languageOfDescription w plikach A.xml, B.xml i C.xml powinny być zgodne). Nie trzeba sprawdzać tego o czym powiedziano, że można zakładać.
11. Trzeba sprawdzić czy dokumenty wejściowe A.xml, [B.xml], C.xml istnieją i są zgodne ze odpowiednimi schematami.
12. Rozwiązanie ma się kompilować i działać przynajmniej na serwerze students.
13. Rozwiązanie powinno zostać przesłane do dnia 26 lutego 2011 (17:59) z serwera students na adres kedar@mimuw.edu.pl, tytuł emaila powinien brzmieć „XML 2010, Z3 poprawka, rozwiązanie”, załącznik do emaila powinien stanowić plik-archiwum ZIP o nazwie ab123456.zip zawierające katalog o nazwie ab123456 (ab123456 należy zastąpić loginem na komputerze students).

14. Kod programu powinien być udokumentowany (wyczerpująco, ale oczywiście bez przesady komentarzy ma być znacznie mniej niż kodu– najlepiej w sposób umożliwiający automatyczną generację dokumentacji (JavaDoc)). W szczególności na początku każdego pliku (w tym pliku README) powinna znajdować się informacja o jego autorze (imię, nazwisko, login na students).
15. Ma zostać dostarczony skrypt umożliwiający szybką kompilację (ant) wraz z opisem uruchomienia (README).
16. Za przesłanie rozwiązania w terminie późniejszym naliczone zostaną punkty karne (każde rozpoczęte 12 godzin – 1 punkt).
17. Proszę sprawdzać [FAQ](#), **mogą** się tam pojawić informacje (plik będzie utworzony jeżeli będzie taka potrzeba).
18. Lokalizacja schematu w dokumencie wyjściowym ma być ustalona na katalog bieżący.
19. Do rozwiązania proszę dostarczyć przykładowy zestaw plików (A.xml, B.xml, C.xml) na których program był testowany. Liczba wpisów w plikach nie ma być duża (około 10).