

XML i nowoczesne technologie zarządzania treścią, rok akademicki 2008/09

Zadanie 3 (termin 2, poprawkowy)

Bajka

Firma zatrudnia informatyków w systemie godzinowym i opłaca ich z budżetów projektów przy których pracują. W tym celu co jakiś czas zbierane są, godzinki (dane mówiące o tym ile godzin dany pracownik przepracował w danym projekcie danego dnia). Firma chce otrzymać program który mając na wejściu dane o pracownikach, projektach, godzinkach (godzinka oznacza czwórkę: kod projektu, kod pracownika, datę, liczbę przepracowanych godzin), stopę procentową oraz datę wygeneruje dwa pliki:

- listę wypłat dla pracowników
- listę godzinek (z zapisanym statusem wskazującym na to czy godzinka została opłacona, a jeżeli nie to z jakiej przyczyny)

Istotne dla firmy jest to, że są dwa rodzaje godzinek:

- godzinki które powinny być opłacone w dniu pracy (umieszczone w przestrzeni nazw „http://www.mimuw.edu.pl/~kedar/xml08/instant”).
- godzinki które mogą być opłacone w dowolnym terminie (umieszczone w przestrzeni nazw „http://www.mimuw.edu.pl/~kedar/xml08”).

Oczywiście opłacenie tych pierwszych jest dla firmy bardziej priorytetowe. Sposób wyliczenia wypłat zostanie opisany dokładniej w dalszej części zadania.

Parametry wywołania programu

Kolejne argumenty oznaczają ścieżki do plików lub inne napisy (stopę procentową, datę).

Wszystkie dokumenty XML mają być zgodne ze schematem **schemat2008Zad3Poprawka.xsd**.

Kolejne parametry wywołania programu oznaczają:

0. plik zawierający dane pracowników firmy (element główny: pracownicy).
1. plik zawierający opisy projektów realizowanych przez firmę (element główny: projekty).
2. plik z godzinkami, godzinka zawiera informację o pracy pracownika nad projektem w danym dniu (element główny: godzinki).
3. Stopa procentowa (odsetki za jeden dzień opóźnienia płatności) wyrażone w promilach (liczba całkowita)
4. datę wypłaty w formacie rrrr-mm-dd (np. 2009-01-10).
5. plik wynikowy (wygenerowany przez program) z pracownikami firmy (element główny: pracownicyOut)
6. plik wynikowy (wygenerowany przez program) z godzinkami (element główny: godzinkiOut)

Klasa zawierająca metodę main ma znajdować się w domyślnym pakiecie i nosić nazwę Wypłacacz.

Algorytm ☺

Program ma opracować listę wypłat dla pracowników (dla podanej daty wypłaty) oraz przetworzyć każdą godzinkę (nadać jej status).

Budżety projektów (budżet podany jest w podelemencie „budżet” elementu „projekt”) wydawane są w następujący sposób:

1. W przypadku gdy data pracy zapisana w elemencie godzinka jest późniejsza niż data wypłaty, w pliku wyjściowym godzince nadawany jest status „WPrzyszlosci”.
 2. W przypadku gdy nie istnieje projekt o podanym w godzince kodzie – godzince nadawany jest status „BlednyKodProjektu”.
 3. W przypadku gdy nie istnieje pracownik o podanym w godzince peselu – godzince nadawany jest status „BlednyPesel”.
 4. W przypadku gdy data pracy nie jest zawarta w okresie realizacji projektu (zapisanym w elemencie projekt) – godzince nadawany jest w pliku wyjściowym status „BlednaDataPracy”.
- Uwaga: Proszę zadbać by w przypadku występowania kilku faktów (np. brak pracownika o podanym peselu oraz brak projektu o podanym kodzie) nadany status w pełni odwzorowywał wszystkie fakty (w tym przypadku zawierał „BlednyKodProjektu” oraz „BlednyPesel”), najdłuższy możliwy ciąg literałów ma długość 3 (np. „BlednyKodProjektu”, „BlednyPesel”, „WPrzyszlosci”).
5. Godzinki o wymienionych wyżej statusach: „BlednaDataPracy”, „BlednyKodProjektu”, „BlednyPesel”, „WPrzyszlosci” nie są opłacane.
 6. W pierwszej kolejności opłacane są **godziny** za które trzeba płacić odsetki (tzn. te znajdujące się w przestrzeni nazw „http://www.mimuw.edu.pl/~kedar/xml08/instant”), godzinki z tej grupy opłacane są w kolejności w jakiej pojawiają się w dokumencie.
 7. W drugiej kolejności opłacane są **godziny** za które nie płaci się odsetek (tzn. te znajdujące się w przestrzeni nazw „http://www.mimuw.edu.pl/~kedar/xml08”), godzinki z tej grupy opłacane są w kolejności w jakiej pojawiają się w dokumencie.
 8. Za jedną godzinę pracy pracownik otrzymuje kwotę określoną przez element stawka odpowiedniego pracownika.
 9. Godziny pracy są opłacane z budżetów odpowiednich projektów.
 10. Każdej godzince która została opłacona z funduszu swego projektu zostaje nadany status „Oplacono”, każdej godzince która z powodu braku wystarczających funduszy w swym projekcie nie została opłacona zostaje nadany status „BrakFunduszy”.
 11. W przypadku gdy część godzin z wpisu godzinka (w pliku z godzinkami) może zostać opłacona, a część nie może (z powodu przekroczenia budżetu), wpis w pliku wynikowym powinien być podzielony na dwie części (status „Oplacono”, „BrakFunduszy”). Ewentualne odsetki oblicza się tylko dla opłaconej części.
 12. Odsetki są obliczane za każdy dzień opóźnienia, tzn. jeżeli godzinka była przepracowana 10 stycznia 2009(i znajduje się w przestrzeni nazw „http://www.mimuw.edu.pl/~kedar/xml08/instant”), a program wywołana z parametrami 12 stycznia 2009 oraz oprocentowaniem 100, to do należności za pracę trzeba doliczyć 20% odsetek.
 13. Odsetki (za wszystkie opłacane w danej godzince godzin) należy zaokrąglić **w górę** do pełnych groszy.
 14. Godzinki w pliku wynikowym (czyli godzinkiOut) powinny znajdować się w tej samej przestrzeni nazw co odpowiadające im godzinki w pliku wejściowym.

15. Program:

- a. wypisuje na standardowe wyjście następującą informację „
Budżet łączny projektów wynosi: <kwota>,
wydano na opłacenie godzinek łącznie: <kwota>,
w tym na opłacenie odsetek wydano: <kwota>,
nie opłacono z powodu braku funduszy godzinek na łączną kwotę: <kwota>,
w tym nie opłacono należnych odsetek na łączną kwotę: <kwota>,
liczba godzinek o statusie WPrzyszłości: <liczba>,
liczba godzinek o statusie BlednaDataPracy: <liczba>,
liczba godzinek o statusie BlednyKodProjektu: <liczba>,
liczba godzinek o statusie BlednyPesel: <liczba>,
liczba pracowników którzy otrzymają wypłaty: <liczba>
”
- b. W wygenerowanym pliku wynikowym z pracownikami firmy(element główny pracownicyOut) powinny znaleźć się informacje o należnych wypłatach (w elementach „doWypłaty”).
- c. Pracownicy w pliku wynikowym z pracownikami powinni być posortowani po numerach PESEL.
- d. Data wypłaty dla wszystkich pracowników powinna być taka sama i być równa parametrowi wywołania programu.
- e. W pliku wynikowym z pracownikami powinny znaleźć się informacje jedynie o pracownikach, którzy uzyskają niezerową wypłatę z któregośkolwiek projektu.
- f. Pracownik w pliku wynikowym może pojawić się co najwyżej raz, a dla każdego projektu w elemencie pracownik może istnieć co najwyżej jeden element „doWypłaty” zawierający sumaryczną kwotę za opłacone godziny (wraz z odsetkami). Elementy „doWypłaty” powinny być posortowane po kodzie
- g. w pliku wynikowym z godzinkami powinny znaleźć się informacje o opłaceniu pracy. Konkretnie dopuszczalna wartość atrybutu status to: „BlednaDataPracy”, „BlednyKodProjektu”, „BlednyPesel”, „WPrzyszlosci”, „Oplacony”, „”. Kolejność wpisów w dokumentach wynikowych powinna być następująca:
 - i. zgodna z kolejnością wpisów w dokumencie wejściowym.
 - ii. w przypadku podziału wpisu, powstałe wpisy są uporządkowane w następującej kolejności (wg statusu): „Oplacony”, „BrakFunduszy”.

Uwagi i wymagania w stosunku do implementacji

1. Program powinien być napisany w Javie, kompilować się i działać w środowisku Sun Java SE 6 (tak będzie testowany).
2. Dokumenty z pracownikami oraz z projektami muszą być wczytywane/zapisywane w standardowy sposób **za pomocą DOM lub JAXB**.
3. Dokumenty z godzinkami nie mogą być wczytywane w całości do pamięci (w szczególności program będzie testowany z dużym dokumentem wejściowym i ograniczoną pamięcią), ma zostać użyty **SAX lub StAX**.
4. Dopuszczalne jest dwukrotne czytanie dokumentów wejściowych.
5. Sposób przetwarzania dokumentów zależy od autora rozwiązania (to znaczy, że np. można stosować kolekcje dostępne w Javie, zamiast operować na drzewie DOM).

6. Należy obsługiwać przestrzenie nazw, pliki wynikowe powinny być zgodne ze stanowiącym załącznik do niniejszego zadania schematem.
7. Plik(i) źródłowe kodu programu mają używać kodowania UTF-8.

Uwagi końcowe:

1. Rozwiązanie powinno zostać przesłane do dnia **28 lutego 2009** (23:59) z serwera students na adres kedar@mimuw.edu.pl, tytuł emaila powinien brzmieć „XML 2008, Z3, 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).
2. 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 powinna znajdować się informacja o jego autorze (imię, nazwisko, login na students).
3. Bardzo proszę by, w miarę możliwości, rozwiązanie było zawarte w pojedynczym pliku lub został dostarczony skrypt umożliwiający szybką kompilację (np. ant).
4. Za przesłanie rozwiązania w terminie późniejszym naliczone zostaną punkty karne (1 dzień – 1 punkt, 2 dni – 2 punkty), rozwiązania nadesłane później sprawdzane nie będą.
5. Proszę sprawdzać [FAQ](#), mogą się tam pojawić informacje np. o modyfikacji schemy (mam jednak nadzieję, że nie będzie to konieczne).
6. Proszę zapoznać się z plikami przykładowymi (pracownicy.xml, projekty.xml, godzinki.xml) oraz zmodyfikować je w taki sposób by przetestować wszystkie możliwe typy sytuacji np.:
 - a. brak pracownika
 - b. brak projektu
 - c. inna kolejność elementów, np. podelementów w elemencie projekt (typ all!)
 - d. konieczność podziału wpisu godzinka na dwie części.
 - e. występowanie komentarzy, instrukcji przetwarzania etc.
7. Lokalizacja schematu w dokumentach wyjściowych może być (w zależności od decyzji autora rozwiązania)
 - a. taka sama jak w odpowiadającym dokumencie wejściowym
 - b. ustalona na plik **schemat2008Zad3Poprawka.xsd**. (w katalogu dokumentu, tzn. bez ścieżki).
8. Można założyć, że dokumenty wejściowe są zgodne ze schematem.
9. Proszę sprawdzić, czy dokumenty wynikowe się walidują (czy mają deklarację schemy!)