

# XML i nowoczesne technologie zarządzania treścią

Egzamin, 2 lutego 2012 r.

Imię i nazwisko: \_\_\_\_\_

25 pytań, 85 minut. Pytania testowe są wielokrotnego wyboru, możliwy jest dowolny układ odpowiedzi.

Każde pytanie – 1 punkt. Aby zdobyć punkt z pytania testowego, należy poprawnie odpowiedzieć na wszystkie podpunkty (3 lub 4 w zależności od pytania). W przypadku pytań opisowych możliwe są także oceny ułamkowe.

Prefiksy `xsd` i `xsl` zgodnie z konwencjami wskazują na przestrzenie nazw, odpowiednio, XML Schema i XSLT.

1. Zaznacz prawdziwe stwierdzenia dotyczące XML:

- a) Każdy dokument XML musi być poprawny składniowo (ang. *well-formed*).
- b) Pojęcie poprawności strukturalnej (ang. *validity*) ma sens jedynie dla dokumentów XML o formalnie zdefiniowanej strukturze.
- c) Przestrzenie nazw mogą być używane jedynie w dokumentach XML o formalnie zdefiniowanej strukturze.

2. Które z poniższych zapisów zaburzają poprawność składniową (ang. *well-formedness*) zawierającego je dokumentu XML-owego?

- a) `<Gosia _Hanka- = "2*Izka"/>`
- b) `<Kaśka>Lenka <łucja łucja=' łucja '/> </Kaśka>`
- c) `<Tosia Ułka=Zenka/>`

3. W DTD można wyrazić następujące modele zawartości:

- a) Element `super` zawiera dowolnie wiele (ale co najmniej jeden) elementów `sub`.
- b) Element `super` zawiera od 3 do 5 elementów `sub`.
- c) Element `super` zawiera tekst, a w nim zanurzone 3 elementy `sub`.
- d) Element `super` zawiera tekst albo 3 elementy `sub`.

4. Encje parametryczne (ang. *parameter entities*) to:

- a) konstrukcje umożliwiające zastępowanie tekstu w DTD
- b) instrukcje umożliwiające przekazanie dodatkowych informacji parserowi XML-owemu
- c) inaczej encje znakowe (ang. *character entities*), umożliwiające zapis odwołań do symboli specjalnych za pomocą nazw i kodów znaków

5. Typ prosty w XML Schema może być zdefiniowany jako:

- a) ograniczenie (*restriction*)
- b) lista (*list*)
- c) sekwencja (*sequence*)

6. Które stwierdzenia na temat definicji `<xsd:attribute name="alien" default="no"/>` są prawdziwe? (Dotyczą oczywiście parsera walidującego dokument w oparciu o schemat XML Schema.)
- a) Podanie w dokumencie wartości atrybutu `alien` jest opcjonalne.
  - b) Podanie w dokumencie wartości atrybutu `alien` jest wymagane.
  - c) Jeśli wartość atrybutu nie zostanie podana w dokumencie, zostanie mu przypisana wartość `no`.
  - d) Jeśli wartość atrybutu zostanie podana w dokumencie, wartość z dokumentu zostanie przysłonięta wartością `no` (obowiązującą wartością atrybutu będzie zawsze `no`).
7. Zaznacz prawdziwe stwierdzenia na temat XML Schema i przestrzeni nazw. Przez „pojedynczy schemat” rozumiemy tu pojedynczy dokument XML zawierający schemat XML Schema (z elementem głównym `xsd:schema`).
- a) Pojedynczy schemat może definiować elementy w wielu różnych przestrzeniach nazw.
  - b) Pojedynczy schemat może definiować modele zawartości zawierające elementy z wielu różnych przestrzeni nazw.
  - c) Wiele schematów może definiować elementy z tej samej przestrzeni nazw.
  - d) Ta sama definicja elementu, po włączeniu schematu ją zawierającego do różnych schematów poleceniem `xsd:include`, może definiować elementy w różnych przestrzeniach nazw.
8. Jakie są dwa alternatywne w stosunku do DTD i XML Schema sposoby definiowania struktury dokumentu? Podaj (może być słownie) dla każdego z nich po jednym przykładzie ograniczenia lub modelu zawartości, którego nie da się wyrazić w XML Schema, a da się w tym formacie.

Chodziło o

- **RelaxNG** – m.in. niedeterministyczne modele zawartości, włączenie atrybutów do modelu zawartości (np. wybór atrybut `tytuł` lub `podelement tytuł`), wybór między zawartością prostą a złożoną,
- **Schematron** – praktycznie dowolne warunki wyrażalne w XPath, w szczególności warunki dla wartości liczbowych (`cena_brutto >= cena_netto`), zależność struktury od wartości (jeśli `sposób_płatności = 'karta'` to występuje `numer_karty`), kontrola występowania potomków / przodków, a nie tylko dzieci, ...

Natomiast akceptowałem też **Examplotron** i (częściowo) propozycje rozwiązań ad-hoc, jak XSLT czy program w Javie.

9. Selektory w Cascading Style Sheets pozwalają wybrać:
- a) elementy A będące dziećmi elementów B
  - b) elementy A posiadające atrybut T o wartości V
  - c) elementy A zawierające (jako dziecko) co najmniej jeden element B
10. W standardzie XSL Formatting Objects:
- a) Formatowanie (np. rozmiar czcionki) opisane jest w elemencie preambuły opisującym styl poszczególnych obiektów formatujących (bloków, tabel itp.).
  - b) Formatowanie opisane jest w atrybutach formatowanych elementów.
  - c) Na szablonie strony (ang. *page master*) można zdefiniować obszary nagłówek i stopki.

11. Napisz wyrażenie XPath 1.0, które wybiera z dokumentu wszystkie elementy A, których bezpośrednim poprzednikiem (na tym samym poziomie drzewa) jest element B, a wśród jego (elementu A) dzieci znajdują się co najmniej 2 elementy C.

```
//A[following-sibling::*[1][name()='B']] [count(C)>=2]
```

```
//B/preceding-sibling::*[1]/self::A [C/following-sibling::C]
```

Bardzo podobne zadanie było na ćwiczeniach. Oceniałem pół punktu za pierwszy warunek, pół za drugi.

12. Wynikiem poniższego zapytania XQuery jest (nie uwzględniając dodatkowych białych znaków dodanych tu dla czytelności)...

```
<doc>{for $i in (1,2,3) return <a>{not($i != (1,2))}</a>}</doc>
```

- a) <doc>  
    <a>true</a>  
    <a>true</a>  
    <a>false</a>  
    </doc>
- b) <doc>  
    <a>false</a>  
    <a>false</a>  
    <a>false</a>  
    </doc>
- c) <doc>  
    <a>true true false</a>  
    </doc>
- d) <doc>  
    <a>false</a>  
    </doc>

13. Wiele prostych szablonów, stosowanie instrukcji `apply-templates`, brak skomplikowanych wyrażeń XPath – to cechy charakterystyczne arkusza XSLT zgodnego z:

- a) modelem przetwarzania opartego o strukturę dokumentu źródłowego
- b) modelem przetwarzania opartego o strukturę dokumentu wynikowego
- c) złymi praktykami tworzenia przekształceń XSLT

14. Poniższy fragment arkusza XSLT

```
<xsl:template match="/">
  <xsl:for-each select="towary/towar">
    <xsl:copy-of select="kolor"/>
  </xsl:for-each>
</xsl:template>
```

przy dokumencie wejściowym

```
<towary>
  <towar nazwa="czajnik">
    <kolor>czerwony</kolor> <kolor>srebrny</kolor>
  </towar>
  <towar nazwa="garnek">
    <kolor>srebrny</kolor> <kolor>niebieski</kolor>
  </towar>
</towary>
```

może (w XSLT 1.0 lub 2.0) spowodować wypisanie na wyjście (przy metodzie serializacji xml, nie uwzględniając białych znaków):

- a) czerwony srebrny
- b) <kolor>czerwony</kolor> <kolor>srebrny</kolor>
- c) czerwony srebrny srebrny niebieski
- d) <kolor>czerwony</kolor> <kolor>srebrny</kolor>  
<kolor>srebrny</kolor> <kolor>niebieski</kolor>

15. Wybieramy model programistyczny dostępu do dokumentów XML spośród parsowania strumieniowego i DOM. Które cechy problemu przemawiają za wyborem parsera strumieniowego?

- a) Odczyt dużych dokumentów w środowisku o ograniczonej ilości pamięci operacyjnej.
- b) Skomplikowane operacje na drzewie dokumentu wymagające jednoczesnego dostępu do wielu węzłów.
- c) Jedyne dane, które potrzebujemy odczytać, znajdują się z dużym prawdopodobieństwem na początku dokumentu.
- d) Znamy schemat dokumentów, które będziemy czytać.  
To jest zawsze pomocne, ale nie jest tu argumentem za/przeciw.

16. Zaznacz prawdziwe stwierdzenia dotyczące technologii SAX.

- a) Kod napisany przez programistę jest wywoływany przez parser w miarę czytania dokumentu.
- b) W oparciu o schemat dokumentu typy zawartości elementów i atrybutów są odzwierciedlane na odpowiednie typy w języku programowania.
- c) Filtry mogą przekazywać do dalszego przetwarzania zmodyfikowane zdarzenia (np. ze zmienioną nazwą elementu).

17. Podaj zalety i wady XML jako formatu elektronicznej wymiany danych w stosunku do tradycyjnego EDI (np. EDIFACT).

- + większa elastyczność, choćby dzięki przestrzeniom nazw; możliwość rozszerzania formatów o elementy prywatne, specyficzne dla branży itp.
- + łatwość zmiany formatu z zachowaniem kompatybilności z istniejącymi aplikacjami (które ignorują nieznanne elementy)
- + możliwość wykorzystania istniejącego wsparcia dla XML (parsery, walidacja, XSLT, XPath)
- + prostsza integracja z WebSerwisami i technologiami WWW (np. eksport do HTML przez XSLT)
- większa objętość dokumentów XML w stosunku do np. EDIFACT
- koszty wdrożenia nowej technologii – w przypadku działających już rozwiązań opartych o stare standardy (na tyle oczywisty i ogólny argument, że nie wymagałem jego podania)

18. Dokument WSDL (Web Services Definition Language) może definiować:

- a) strukturę komunikatów i typ zawartości poszczególnych elementów
- b) zestaw operacji dostępnych w ramach usługi sieciowej
- c) adres URL, pod którym usługa sieciowa jest dostępna

19. Ścieżka XPointer `dokument.xml#def-podatnik` wskazuje w dokumencie `dokument.xml`

- a) wszystkie elementy o nazwie `def-podatnik`
- b) element o nazwie `def-podatnik`, o ile jest on elementem głównym
- c) element o ID równym `def-podatnik`

20. Które z poniższych cech pasują bardziej do standardu DITA (Darwin Information Typing Architecture) niż do DocBook?

- a) Dokument ma tradycyjną strukturę rozdziałów i akapitów.
- b) Informacje zgromadzone są w tematach, nad którymi zbudowane mogą być dokumenty-mapy.
- c) Jest wykorzystywany do zapisywania dokumentacji technicznych. [Oba są.](#)

21. W proceso-centricznym podejściu do zarządzania dokumentami („zarządzanie procesem”) typowe jest, że:

- a) Dokumenty są cały czas dostępne dla tych samych uprawnionych użytkowników.
- b) Użytkownik wykonuje zadania wskazane przez system.
- c) Użytkownik przeglądając katalogi lub wyszukując wybiera zasoby, z których w danej chwili korzysta.
- d) System kieruje przepływem dokumentów zgodnie ze zdefiniowanymi regułami biznesowymi.

22. Podaj argumenty za i przeciw łączeniu w jednym systemie zarządzania dokumentami oraz publikacji (po dwa argumenty).

+ Tworzymy dokumenty zwykle po to, aby w jakiejś formie opublikować, więc funkcjonalność publikacji jest bardzo „na miejscu” w takim systemie.

+ System składu może zwracać informacje np. o rozłożeniu tekstu na strony, które możemy chcieć przechowywać – łatwiej to osiągnąć w zintegrowanym systemie.

+ Publikacja może być wpleciona w biznesowy proces obiegu dokumentów. Łatwiej to zautomatyzować przy zintegrowanym systemie.

+ Łatwiej zapewnić dostępność aktualnej wersji dokumentów – gdy jest stale potrzebna ze względów biznesowych lub do podglądu publikowalnej wersji podczas tworzenia treści.

+ Dla specyficznych zastosowań za najlepsze rozwiązanie możemy uznać budowę systemu szytego na miarę (żeby nie powiedzieć 'zkastomizowanego'). W takim przypadku panowanie w pełni nad sposobem publikacji pozwoli nam zrealizować specyficzne dla naszego problemu wymagania, niedostępne w rozwiązaniach dostępnych na rynku.

- Twórcy treści mogą zacząć tworzyć treść pod kątem konkretnej publikacji i wyglądu opublikowanego dokumentu – treść może stać się mniej wartościowa dla przyszłych zastosowań.

- Rozdzielenie tych funkcjonalności może zwiększyć reużywalność oprogramowania i ułatwić pielęgnację.

- Rozdzielając można skorzystać z gotowych rozwiązań dostępnych na rynku.

23. Podaj trzy możliwości (funkcjonalności dla użytkownika lub cechy algorytmu) zaawansowanego wyszukiwania (w porównaniu do prostego wyszukiwania tekstu w tekście).

- Indeksowanie treści i wyszukiwanie w oparciu o indeks.
- Przekształcenia zapytania, np.: usunięcie nieznaczących słów, poprawienie literówek, doprowadzenie wyrazów do podstawowej formy gramatycznej.
- Uwzględnianie języka.
- Uwzględnianie typu dokumentu, metadanych, uprawnień.
- Wyrażenia logiczne w zapytaniach.
- Szukanie tylko we fragmentach o określonym znaczeniu (znaczniki semantyczne treści lub na podstawie kontekstu).
- Podpowiedzi, rankingi.
- ...

24. Wymień 3 powody, dla których wyszukiwanie w intranecie jest łatwiejsze niż w Internecie.

- Mniejsza liczba dokumentów do zindeksowania.
- Większa dostępność dokumentów.
- Często dostępna lista wszystkich dokumentów (niepotrzebne pająki).
- Często dostępne metadane, uprawnienia.
- Często dokumenty są zgodne z konkretną strukturą – można ją uwzględnić w wyszukiwaniu.
- Większa wiarygodność danych i linków, brak dokumentów "śmieciowych", walki o pozycjonowanie itp.

25. W usługach sieciowych w stylu REST (od *Representational State Transfer*):

- a) Komunikaty przesyłane są w formacie SOAP.
- b) Bezpośrednio wykorzystywane są operacje („metody”) protokołu HTTP, co uzależnia korzystanie z takich usług od protokołu HTTP.
- c) Można używać innych niż XML sposobów reprezentacji danych, np. JSON.