

Programowanie pod Windows

Zestaw 8

eXtensible Markup Language

ogłoszenie listy: 10-05-2005

ważność listy: 31-05-2005

Streszczenie

Poniższe problemy skomponowano w sposób maksymalnie atomowy, nic nie stoi jednak na przeszkodzie aby kilka kolejnych powiązanych zadań połączyć w jednej większej aplikacji.

1. XML

Zaprojektować prostą strukturę XML do przechowywania danych o studentach. Każdy student reprezentowany jest **co najmniej** przez podstawowy zbiór atrybutów osobowych, ma dwa adresy (stały i tymczasowy) oraz listę zajęć na które uczęszcza wraz z ocenami.

[1p]

2. XSD

Schemat struktury z poprzedniego zadania wyrazić w postaci XSD. Zadbać o poprawne opisane reguł walidacji zakresu danych (pewne dane mogą być opcjonalne) i ich zawartości (pewne dane mogą przyjmować wartości o konkretnym formacie).

[1p]

3. XML + XSD

Napisać program, który używa zaprojektowanego w poprzednim zadaniu schematu XSD do walidacji wskazanych przez użytkownika plików XML i raportuje ewentualne niezgodności.

[1p]

4. XML - serializacja

Napisać prostego klienta struktury XML z zadania 3, który pliki XML czyta i zapisuje mechanizmem serializacji do struktur danych zamodelowanych odpowiednimi atrybutami.

[1p]

5. XML - DOM

Napisać prostego klienta struktury XML z zadania 3, który pliki XML czyta i zapisuje za pomocą modelu DOM (`XmlDocument`).

[1p]

6. XML - strumienie

Napisać prostego klienta struktury XML z zadania 3, który pliki XML czyta i zapisuje za pomocą mechanizmów strumieniowych (`XmlTextReader`, `XmlTextWriter`).

[1p]

7. XML - analiza

Porównać trzy metody obsługi XML z poprzednich zadań. Porównanie powinno uwzględniać:

- (a) czas odczytu/zapisu

- (b) łatwość implementacji odczytu/zapisu
- (c) podatność na konserwację (np. ewolucję struktury)

Ponadto rozważyć model aplikacji, w której **nie ma** żadnych pośrednich struktur danych w których przechowywane byłyby dane z XML, a warstwa logiki biznesowej korzysta bezpośrednio ze struktury XML przechowywanej w pamięci na przykład w obiekcie DOM.

Rozwiązanie zadania powinno mieć formę pisemną i nie powinno przekraczać 150 słów.
[1p]

8. XML jako protokół komunikacyjny

Napisać prostego okienkowego klienta protokołu RSS (w dowolnej wersji).

Klient powinien nawiązywać połączenie sieciowe do wskazanego źródła danych i udostępniać listę publikowanych informacji. Wybór linka przez użytkownika powinien powodować pobranie zawartości wskazanego artykułu i zaprezentowanie go użytkownikowi w uruchomionej z boku nowej instancji przeglądarki internetowej lub ([+1p]) w kontrolce ActiveX Internet Explorera hostowanej w obrębie aplikacji.

Uwaga! Do połączeń HTTP użyć gotowych klas z przestrzeni nazw System.Net.

Uwaga! Mechanizm hostowania kontrolki IE omówiony był na wykładzie.

[3+1p]