

Laborator de Structuri de Date – Lucrarea nr. 14

Stocarea datelor/structurilor în fișiere/format XML

1. Formatul XML.....	1
2. Reprezentarea documentelor XML cu ajutorul unui arbore	2
3. Modalități de citire a fișierelor XML	3
4. Exerciții.....	4

1. Formatul XML

Un document XML (Extensible Markup Language) permite stocarea unor de structuri arborescente de date prin intermediul unor câmpuri și valori textuale grupate sub formă de tag-uri. XML se aseamănă cu limbajul HTML (HyperText Markup Language) doar că în cazul XML tag-urile nu sunt predefinite (nu au semnificații bine definite). Limbajul este considerat a fi extensibil deoarece permite utilizatorilor să își definească propriile elemente.

În exemplul de mai jos este furnizat un exemplu de document XML ce conține o parte dintre studenții unei facultăți. Tag-urile pot fi identificate ca fiind incluse între caracterele < ... >, ca de exemplu Prenume, Nume, Nota. Acestea se mai numesc și **elemente**. De asemenea, un tag poate conține și o serie de **attribute**, de exemplu Grupa. Fiecare tag trebuie să aibă un tag corespondent de final, pentru a ști unde se termina. Acest tag de final are același nume ca și tag-ul inițial, dar este între caracterele </ ... >.

Considerăm fișierul SampleData.xml furnizat ca anexă a laboratorului:

```
<?xml version="1.0" encoding="utf-8"?>
<Studenti>
  <Student Grupa="1101">
    <Prenume>Andrei</Prenume>
    <Nume>POPESCU</Nume>
    <Nota>10</Nota>
  </Student>
  <Student Grupa="1101B">
    <Prenume>Mihai</Prenume>
    <Nume>IONESCU</Nume>
    <Nota>9</Nota>
  </Student>
</Studenti>
```

Prima linie a fișierelor XML conține de obicei declararea XML. Aceasta conține versiunea XML folosită precum și codarea folosită. În funcție de necesități, poate să conțină și alte informații. Dat fiind faptul că reperele element (tagurile) sunt textuale, pentru o gestionare adecvată a fișierelor xml trebuie ales un tip de codare (encoding="..") în acord cu lexicul vizat, pentru a fi interpretate corect diacriticele etc.

Aceste fișiere XML pot fi cel mai ușor încărcate și inspectate prin deschiderea lor în browserele uzuale de internet.

2. Reprezentarea documentelor XML cu ajutorul unui arbore

Unul din avantajele folosirii fișierelor XML este acela că organizarea datelor este descrisă în interiorul fișierelor. Structura de date ce corespunde cel mai bine datelor conținute în fișierele XML este un arbore de grad oarecare.

Toate documentele XML trebuie să conțină un element rădăcină. Pentru exemplul considerat anterior acesta este <Studenți>. Elementele unui document XML formează un arbore. Toate elementele pot avea unele sub-elemente. Acestea vor fi reprezentate în arbore ca și copiii elementului respectiv.

O structură generală a arborelui XML este:

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

Pentru exemplul considerat mai sus, arborele rezultat este prezentat în figura 1.

Elementul rădăcină este <Studenti>. Toate elementele <Student> sunt conținute în <Studenti>. Un element <Student> are 3 copii <Prenume> <Nume> <Nota>.

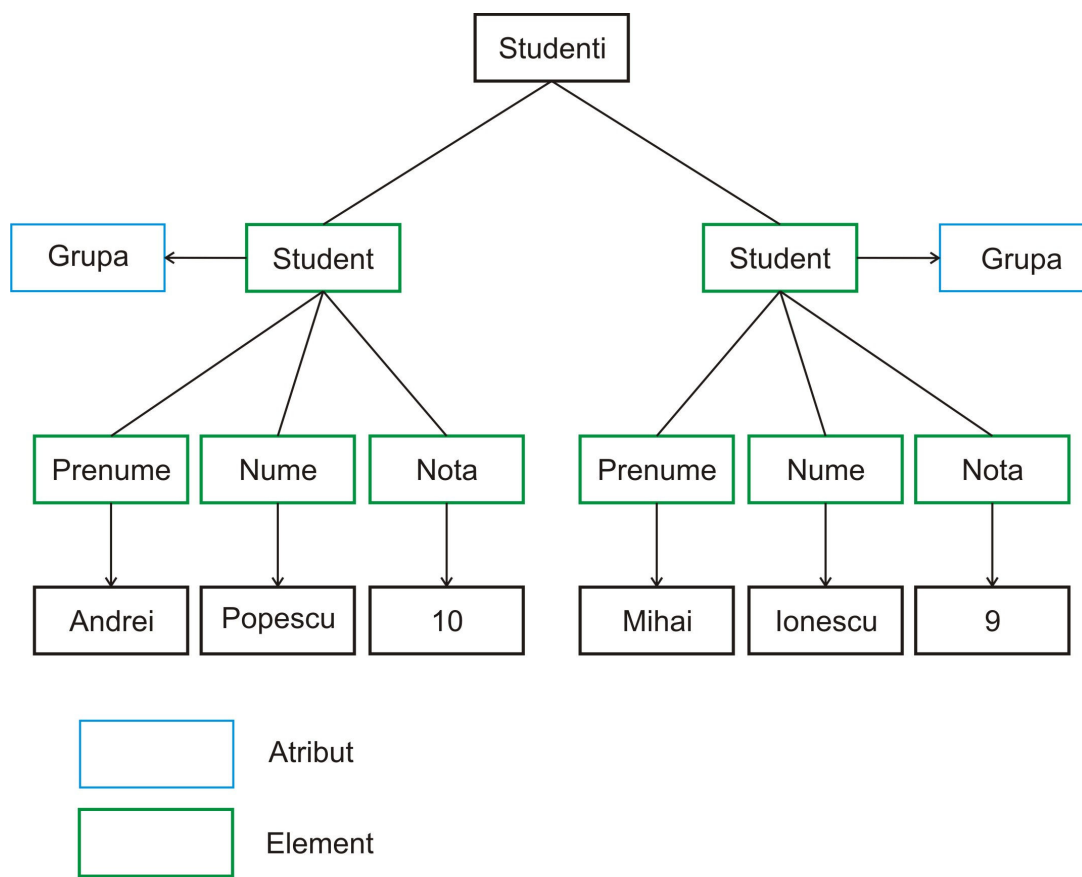


Figura 1. Structura arborescenta a fișierului SampleData.xml

3. Modalități de citire a fișierelor XML

Pentru a putea citi un fișier XML și pentru a putea folosi elementele sale constitutive în cadrul unui program C/C++, soluția recomandată este aceea de folosire a unei biblioteci ce oferă funcționalitățile necesare.

În mediul on-line se pot găsi nenumărate astfel de biblioteci și codul sursă aferent, fiecare prezentând atât avantaje cât și dezavantaje. Pentru exemplificarea modalităților de lucru cu aceste librării, s-a ales librăria tinyXML. Aceasta poate fi descărcată de la adresa: <http://sourceforge.net/projects/tinyxml/>

Această librărie este o librărie simplă ce poate fi ușor integrată în orice program, fie în C cât și în C++. Documentația aferentă acestei librării poate fi consultată la adresa:

<http://www.grinninglizard.com/tinyxmldocs/index.html>

Pentru a putea folosi librăria, se descarcă fișierul zip ce conține fișierele sursă, exemple, documentație etc., de la adresa:

<http://sourceforge.net/projects/tinyxml/files/tinyxml/2.6.2/>

Din această arhivă se copiază în directorul de lucru fișierele:

- tinyxml.cpp
- tinyxml.h
- tinyxmlerror.cpp
- tinyxmlparser.cpp
- tinystl.cpp
- tinystl.h

Aceste fişiere vor trebui incluse în proiectul în care se doreşte prelucrarea fişierelor XML. Pentru folosirea bibliotecii standard C++, în fişierul tinyxml.h se adaugă pe prima linie `#define TIXML_USE_STL`

Codul sursă folosit pentru afişarea element cu element al fişierului XML SampleData.xml este furnizat ca anexă la acest laborator.

4. Exerciţii

1.
 - a. Deschideţi fişierul SampleData.xml cu un browser de internet şi verificaţi corectitudinea dpdv sintactic al acestuia.
 - b. Utilizaţi programul test_xml_cpp_clr.exe furnizat în arhiva laboratorului pentru a vizualiza şi inspecta structura arborescentă a fişierului SampleData.xml. (ambele fişiere trebuie să se afle în acelaşi director)
2. Pe baza fişierelor furnizate în arhiva laboratorului (fişierele librăriei tinyXML precum şi fişierul xml.cpp în care este dat un exemplu de parcurgere a fişierului SampleData.xml), realizaţi o aplicaţie care deschide fişierul XML SampleData.xml şi afişează elementele acestuia.
3. Creaţi o clasă Student care să conţină ca membri informaţiile unui element <Student> din fişierul XML.
4. Creaţi un arbore de grad oarecare în care nodul rădăcină să fie un nod de tip Student şi fără informaţie utilă (fără membru atom). Copii acestui nod vor fi noduri în care Atomul va fi constituit din obiecte de tip Student.
5. Afişaţi arborele creat.
6. Converteţi un fişier xml (cel creat anterior sau altul) într-un arbore, efectuaţi unele operaţii cu acel arbore (adăugaţi un nou element, ştergeţi un element etc.) şi salvaţi din nou arborele în fişierul xml.