

Dossier de réalisation

H4212

Etienne BRODU, Chafik BACHETENE, Adrien BROCHOT,
Johann CHAZELLE, Naby Daouda DIAKITE, Baptiste LECORNU, Thanh PHAN DUC

21 avril 2011

Table des matières

```

1  #include <QtCore/QCoreApplication>
2  #include <iostream>
3  #include "Element.hpp"
4  #include "ElementComplexe.h"
5  #include "ElementTextuel.h"
6  #include "VisitorInterface.hpp"
7  #include "VisitorDisplay.h"
8  #include "visitorbuild.h"
9  #include "XMLDocument.h"
10 #include "contenusimple.h"
11 #include "contenusequence.h"
12 #include "contenuchoix.h"
13 #include "DTD.h"
14 #include "commun.h"
15
16 int main(int argc, char *argv[])
17 {
18     //QCoreApplication a(argc, argv);
19
20
21     DTD* dtd = new DTD();
22
23     XMLDocument *xml = new XMLDocument();
24     ElementName *nom = new ElementName();
25     nom->first = "";
26     nom->second = "auteur";
27
28
29     //ElementComplexe(ElementName *n, AttList *a, list<Element *> *e);
30     list<Element*>* listElt = new list<Element*>;
31     list<Element*>* listElt2 = new list<Element*>;
32     AttList* attList = new AttList();
33     Attribut *att = new Attribut();
34     att->first = "nom";
35     att->second = "Byfall";
36     ElementComplexe* auteurXML = new ElementComplexe(nom, attList, listElt);
37     //auteurXML->addAttribute(att);
38
39     ElementName *nomLivre = new ElementName();
40     nomLivre->first = "";
41     nomLivre->second = "livre";
42
43
44
45     ElementTextuel* text = new ElementTextuel("Chacha");
46
47
48     AttList* attList2 = new AttList();
49     ElementComplexe* livreXML = new ElementComplexe(nom, attList2, listElt2);
50     livreXML->addElement(text);
51     auteurXML->addElement(livreXML);
52     xml->setElement(auteurXML);
53
54     cout<<xml->build(new VisitorBuild());
55
56
57
58     //DTD
59     ContenuSimple *livre = new ContenuSimple("livre",dtd);
60     ContenuSimple *pcdata = new ContenuSimple("#PCDATA",dtd);
61
62
63
64
65     DeclarationElement* elementRacine = new DeclarationElement("auteur", livre);
66     DeclarationElement* eltA = new DeclarationElement("livre", pcdata);
67
68
69     dtd->addDeclarationElement(elementRacine);
70     dtd->addDeclarationElement(eltA);
71
72
73
74     std::cout<<dtd->accept(new VisitorBuild())<<std::endl;
75     std::cout<<"Fin"<<std::endl;
76

```

```
77     return 0;  
78  
79 }
```