XML文档调用外部DTD文档：

XML文档：

<?xml version="1.0" encoding="GB2312"?>

<!DOCTYPE Library SYSTEM "Library">

<Library>

<Book BookId="24">

<Title>巴黎圣母院</Title>

<Author>维克多·雨果</Author>

<Price>80</Price>

<Publisher>译林出版社</Publisher>

<Cost>85</Cost>

</Book>

<Member>

<Name>

<FirstName>Adenauer</FirstName>

<MiddleName>noname</MiddleName>

<LastName>Konrad</LastName>

</Name>

<Address>

<HouseNumber>504</HouseNumber>

<Street>Knigsallee</Street>

<City>杜塞尔多夫</City>

</Address>

</Member>

</Library>

DTD文档：

<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT Library (Book,Member+)>

<!ELEMENT Book (Title,Author,Price,Publisher,Cost)>

<!ELEMENT Member (Name,Address)>

<!ELEMENT Name (FirstName,MiddleName,LastName)>

<!ELEMENT Address (HouseNumber,Street,City)>

<!ATTLIST Book

BookId CDATA #REQUIRED>

XML解析库解析的代码和结果：

#include <iostream>

#include "D:\Coursework\软工课程作业\XML\外部DTD（XML）\tinyxml\_2\_6\_2\tinyxml\tinystr.h"

#include "D:\Coursework\软工课程作业\XML\外部DTD（XML）\tinyxml\_2\_6\_2\tinyxml\tinyxml.h"

#include <string>

using namespace std;

void readSchoolXml() {

const char \* xmlFile = "a.xml";

TiXmlDocument doc;

if (doc.LoadFile(xmlFile)) {//打开a.xml文件

doc.Print();

}

else {

cout << "can not parse a.xml" << endl;//打开失败弹出提醒

return;

}

TiXmlElement\* libraryElement = doc.RootElement(); //library元素 也就是根元素

TiXmlElement\* bookElement = libraryElement->FirstChildElement(); // book元素 根元素的第一个儿子元素

TiXmlAttribute\* bookid = bookElement->FirstAttribute(); //获得book的id属性

cout << bookid->Name() << " : " << bookid->Value() << std::endl; //显示book的属性

TiXmlElement\* TietleElement = bookElement->FirstChildElement();//获得TietleElement title是book的第一个儿子

for (; TietleElement != NULL; TietleElement = TietleElement->NextSiblingElement()) { //当title还有兄弟节点，循环一直进行

string contactType = TietleElement->Value();

string contactValue = TietleElement->GetText();

cout << contactType << " : " << contactValue << std::endl;//显示标签和值

}

TiXmlElement\* memberElement = bookElement->NextSiblingElement(); // member元素 是根节点的第二个儿子，也就是book节点的兄弟

TiXmlElement\* nameElement = memberElement->FirstChildElement();//name是member的第一个儿子

TiXmlElement\* FirstnameElement = nameElement->FirstChildElement();//Firstname又是name的第一个儿子

for (; FirstnameElement != NULL; FirstnameElement = FirstnameElement->NextSiblingElement()) { //当Firstname还有兄弟节点，循环一直进行

string contactType = FirstnameElement->Value();

string contactValue = FirstnameElement->GetText();

cout << contactType << " : " << contactValue << std::endl;//显示标签和值

}

TiXmlElement\* addressElement = nameElement->NextSiblingElement();//address元素 是member的第二个儿子，也就是name节点的兄弟

TiXmlElement\* HouseNumberElement = addressElement->FirstChildElement();//Housenumber是address的第一个儿子

for (; HouseNumberElement != NULL; HouseNumberElement = HouseNumberElement->NextSiblingElement()) {//当Housenumber还有兄弟节点，循环一直进行

string contactType = HouseNumberElement->Value();

string contactValue = HouseNumberElement->GetText();

cout << contactType << " : " << contactValue << std::endl;//显示标签和值

}

}

int main() {

readSchoolXml();

system("pause");

return 1;

}

