**Xml文档创建外部DTD**

**DTD文档要求：**

（1）根元素名称为Library。

（2）Library必须包含2个元素：Book和Member，分别表示书的详细信息和发行人。

（3）Book必须包含一个BookId属性。所有其它的元素，例如Title、Author、Price、Publisher和Cost都只能作为Book的子元素。

（4）元素Member包含Name和Address两个子元素。元素Name包含FirstName、MiddleName和LastName三个子元素。元素Address包含HouseNumber、Street、City三个子元素。

（5）用C++的XML解析库解析所做XML文档，附运行结果图。

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)>

<!ELEMENT Title (#PCDATA)>

<!ELEMENT Author (#PCDATA)>

<!ELEMENT Price (#PCDATA)>

<!ELEMENT Publisher (#PCDATA)>

<!ELEMENT Cost (#PCDATA)>

<!ELEMENT FirstName (#PCDATA)>

<!ELEMENT MiddleName (#PCDATA)>

<!ELEMENT LastName (#PCDATA)>

<!ELEMENT HouseNumber (#PCDATA)>

<!ELEMENT Street (#PCDATA)>

<!ELEMENT City (#PCDATA)>

<!ATTLIST Book

BookId CDATA "#REQUIRED">

**XML文档：**

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

<!DOCTYPE Library SYSTEM "Library.dtd">

<Library>

<Book BookId="23">

<Title>Java</Title>

<Author>韩顺平</Author>

<Price>35</Price>

<Publisher>韩顺平</Publisher>

<Cost>35</Cost>

</Book>

<Member>

<Name>

<FirstName>韩</FirstName>

<MiddleName>顺</MiddleName>

<LastName>平</LastName>

</Name>

<Address>

<HouseNumber>温习之家</HouseNumber>

<Street>和平路23</Street>

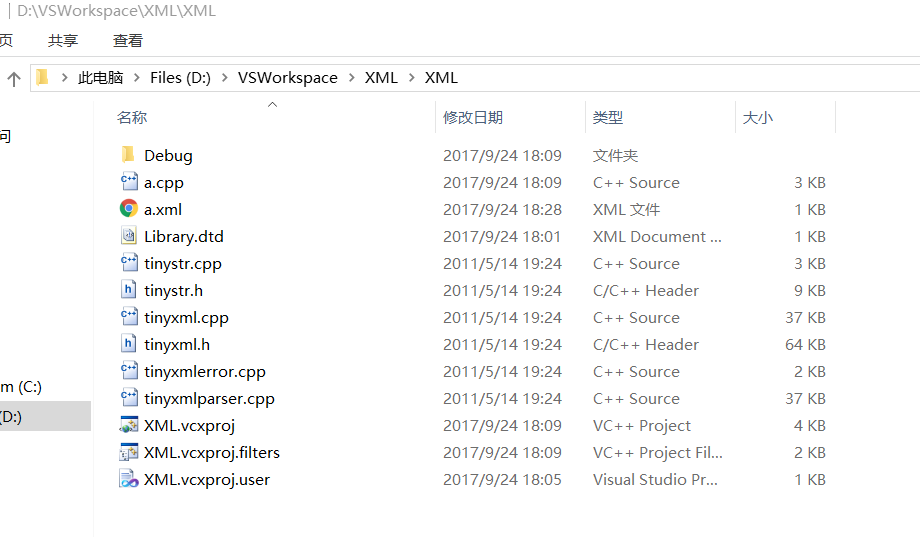
<City>上海</City>

</Address>

</Member>

</Library>

**项目XML目录：**



**先下载tinyxml, 解压之后在VS2010->项目XML->添加现有项，选择tinystr.cpp  tinyxml.cpp  tinyxmlerror.cpp  tinyxmlparser.cpp  tingystr.h  tinyxml.h这六个文件，在a.cpp中写解析代码，如下：**

#include <iostream>

#include "tinystr.h"

#include "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;

}

**运行结果图：**

