先下载tinyxml解析库, 解压之后在VC工程的文件夹目录添加现有项，选择tinystr.cpp tinyxml.cpp tinyxmlerror.cpp tinyxmlparser.cpp tingystr.h tinyxml.h这六项进行添加

然后进行用C++代码读取xml的文件内容

代码：

#include <iostream>

#include "tinystr.h"

#include "tinyxml.h"

#include <string>

using namespace std;

void readSchoolXml() {

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

TiXmlDocument doc;

if (doc.LoadFile(xmlFile))//打开Library.xml文件

{

doc.Print();

}

else

{

cout << "can not parse Library.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

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

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

{ //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

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

{//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()

{

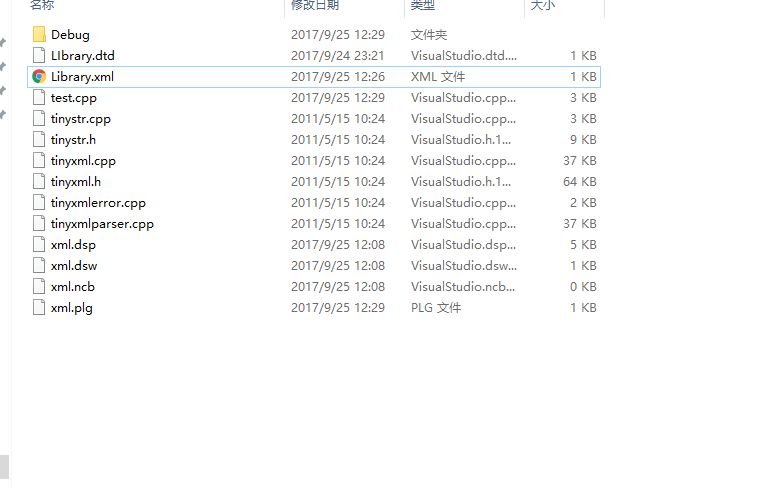
readSchoolXml();

system("pause");

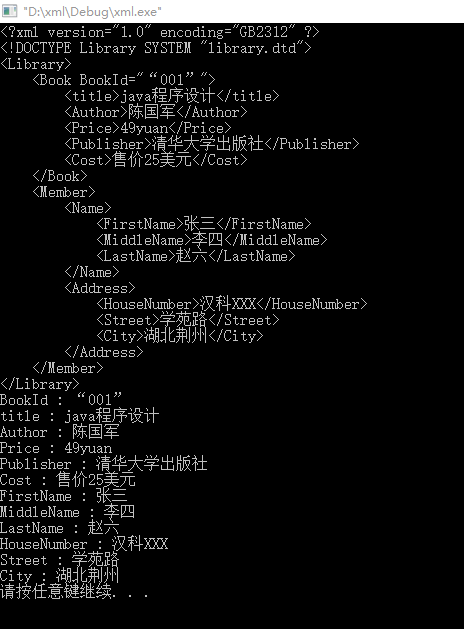
return 1;

}

Xml工程目录：



运行结果：



DTD 是一种保证XML文档格式正确的有效方法