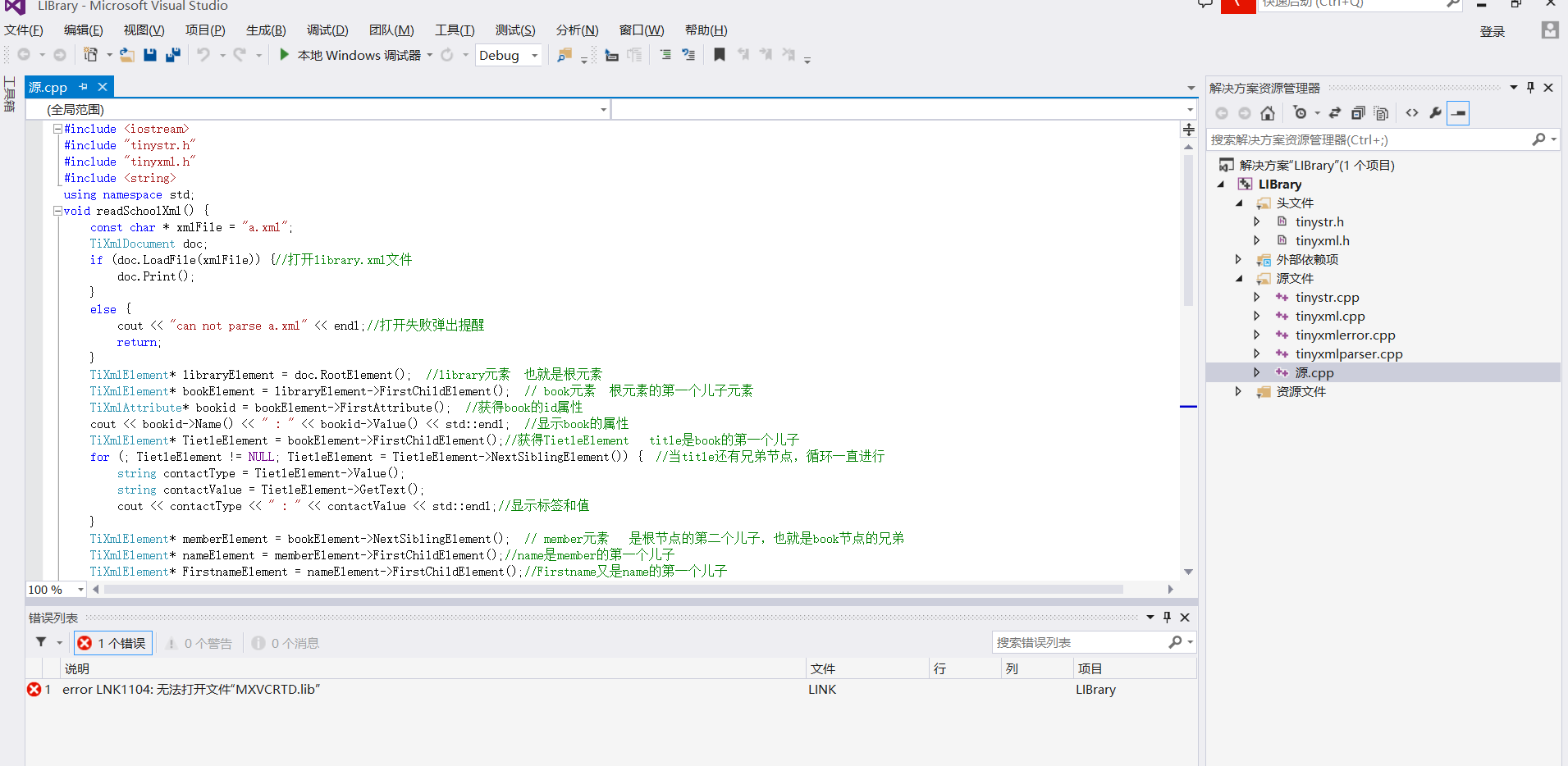
开始使用tinyxml .把两个头文件和四个源文件添加到工程里面，解析简单的xml文档是可以使用，后来第二天把我自己的文档添加进去后就出现了问题，后来一直解决不了就打算先用Java完成作业问题再看以后能不能解决。



然后在Java工程写一个解析文件

**package** library;

**import** java.io.IOException;

**import** java.util.List;

**import** javax.xml.parsers.DocumentBuilder;

**import** javax.xml.parsers.DocumentBuilderFactory;

**import** javax.xml.parsers.ParserConfigurationException;

**import** org.w3c.dom.Document;

**import** org.w3c.dom.NamedNodeMap;

**import** org.w3c.dom.Node;

**import** org.w3c.dom.NodeList;

**import** org.xml.sax.SAXException;

**public** **class** zwq {

**public** **static** **void** main(String[] args) {

//创建一个DocumentBuilderFactory的对象

DocumentBuilderFactory dbf = DocumentBuilderFactory.*newInstance*();

//创建一个DocumentBuilder的对象

**try** {

//创建DocumentBuilder对象

DocumentBuilder db = dbf.newDocumentBuilder();

//通过DocumentBuilder对象的parser方法加载books.xml文件到当前项目下

Document document = db.parse("books.xml");

//获取所有book节点的集合

NodeList bookList = document.getElementsByTagName("book");

NodeList MenberList = document.getElementsByTagName("Menber");

//通过nodelist的getLength()方法可以获取bookList的长度

System.***out***.println("Ò»¹²ÓÐ" + bookList.getLength() + "±¾Êé");

//遍历每一个book节点

*xmlReader*(bookList);

System.***out***.println("Ò»¹²ÓÐ" + MenberList.getLength() + "¸ö·¢ÐÐÈË");

*xmlReader*(MenberList);

} **catch** (ParserConfigurationException e) {

e.printStackTrace();

} **catch** (SAXException e) {

e.printStackTrace();

} **catch** (IOException e) {

e.printStackTrace();

}

}

**private** **static** **void** xmlReader(NodeList bookList) {

**for** (**int** i = 0; i < bookList.getLength(); i++) {

System.***out***.println("====¿ªÊ¼±éÀúµÚ" + (i + 1) + "ÌõÄÚÈÝ====");

//通过 item(i)方法 获取一个book节点，nodelist的索引值从0开始

Node book = bookList.item(i);

//获取book节点的所有属性集合

NamedNodeMap attrs = book.getAttributes();

//遍历book的属性

**for** (**int** j = 0; j < attrs.getLength(); j++) {

//通过item(index)方法获取book节点的某一个属性

Node attr = attrs.item(j);

//获取属性名

System.***out***.print(attr.getNodeName());

//获取属性值

System.***out***.println("£º" + attr.getNodeValue());

}

*xml*(book);

System.***out***.println("====½áÊø±éÀúµÚ" + (i + 1) + "ÌõÄÚÈÝ====");

}

}

**private** **static** **void** xml(Node book) {

//解析book节点的子节点

NodeList childNodes = book.getChildNodes();

**for** (**int** k = 0; k < childNodes.getLength(); k++) {

//区分出text类型的node以及element类型的node

**if** (childNodes.item(k).getNodeType() == Node.***ELEMENT\_NODE***) {

//获取了element类型节点的节点名

System.***out***.print(childNodes.item(k).getNodeName());

//获取了element类型节点的节点值

System.***out***.println("--" + childNodes.item(k).getFirstChild().getNodeValue().replaceAll("\\s\*", ""));

}

**if** (childNodes.item(k).hasChildNodes()) {

*xml*(childNodes.item(k));

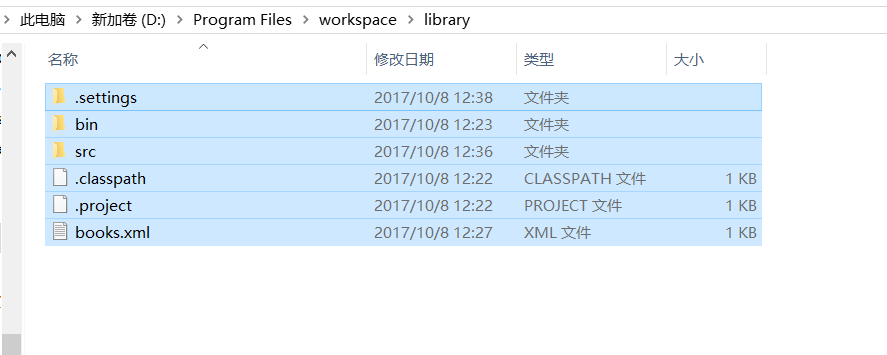
}

}

}

}

之后在项目里面添加自己的xml文档



最后就可以解析出