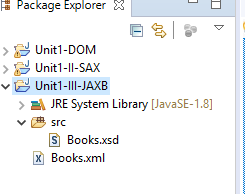
PRACTICE 6. ACCESS XML USING JAXB:Unmarshalling, adding, marshalling

OBJECTIVE: Use JAXB to access XML files

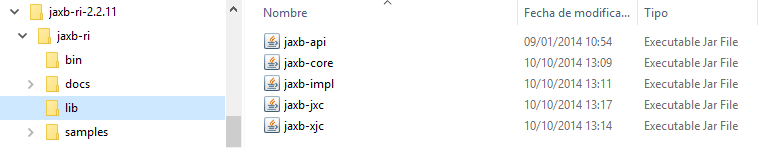
# PRACTICE:

Before we start: class ArrayList. <http://puntocomnoesunlenguaje.blogspot.com/2012/12/arraylist-en-java.html>

1. We need to have **Books.xml in the proyect root folder**, and **Books.xsd in the src folder**



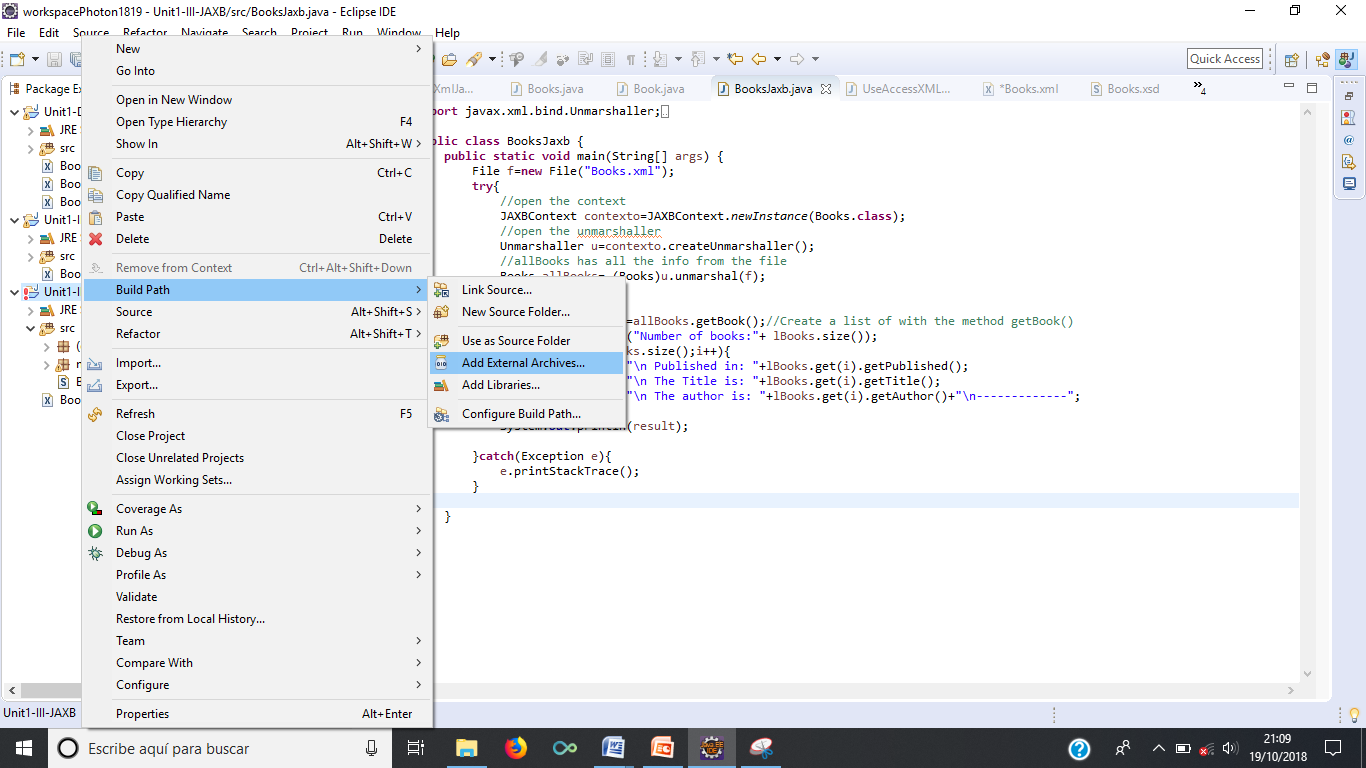
1. To generate JAXB classes, download the file jaxb.ri-2-2-11.zip from the Internet. Unzip it. Inside the forlder lib we find the necessary files for our JAXB programs



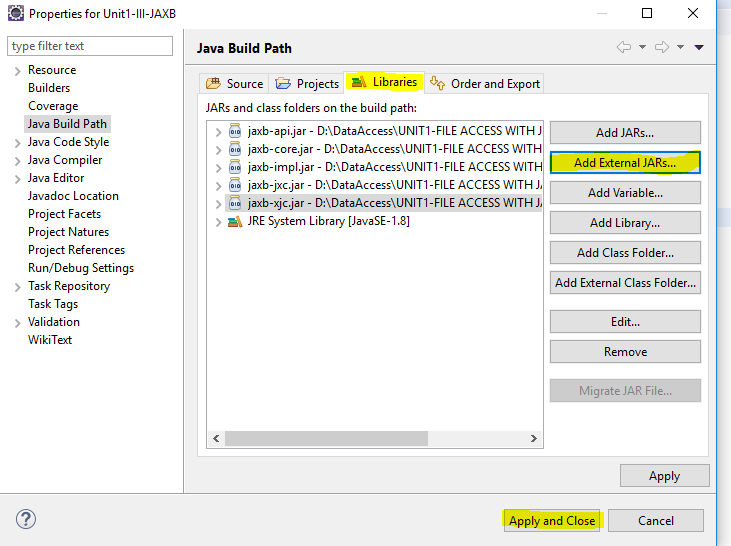
NOTE: COPY THE FOLDER JAXB-RI-2.2.11 IN YOUR PROJECT FOLDER TO AVOID PROBLEMS IF YOU CHANGE YOUR WORKSPACE.

1. To use the jar files, we go to Eclipse, right-click on the project and select

Build Path>>Configure Build Path



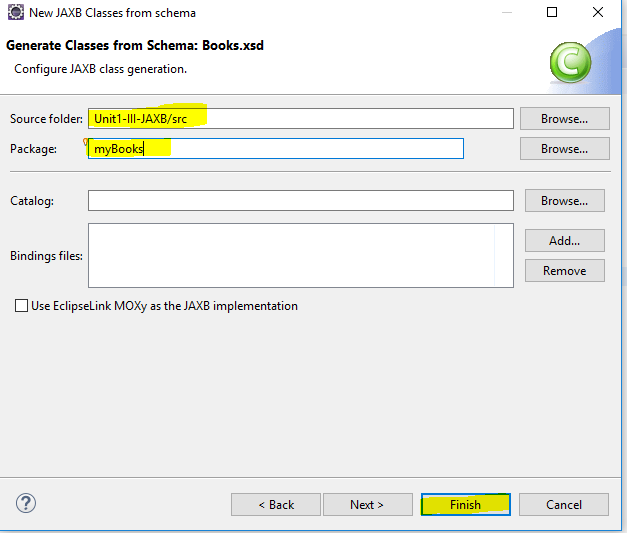
Click on “Add External JARs…” and look for the path of your jaxb Jar files. Select the 5 jar files one by one. Then click on “Apply and close”



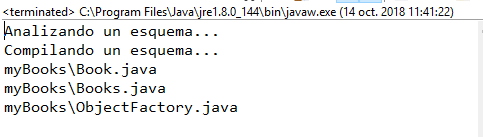
Ahora podemos ir a generar los archivos de conexión.

Sobre el archivo XSD con botón derecho le damos a Generate>>JAXB files

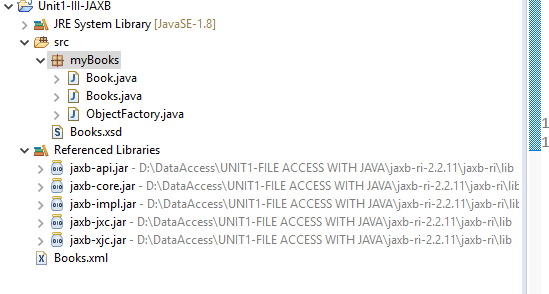
1. Now we can generate the connection files: Right-click on Books.xsd and select Generate>>JAXB files



1. You’ll see the message that the classe Book.java, Books.java and ObjectFactory.java are created



At this point, we must have something like this:



1. Now we must write our code: First we will open an XML file, create a list from the file, and read the information from the list with the methods of the class Book (Unmarshall).

**import** javax.xml.bind.Unmarshaller;

**import** javax.xml.bind.JAXBContext;

**import** javax.xml.bind.Marshaller;

**import** myBooks.\*;

**import** java.io.\*;

**import** java.util.List;

**public** **class** BooksJaxb {

JAXBContext contexto;

Unmarshaller u;

Books allBooks;

List <Book> lBooks;

**public** **void** openXMLwithJAXBandShow(File f) {

//UNMARSHALLING: from the XML file, we fill the object allBooks and the list lBooks

**try**{

//open the context

contexto=JAXBContext.*newInstance*(Books.**class**);

//open the unmarshaller

u=contexto.createUnmarshaller();

//allBooks has all the info from the file

allBooks= (Books)u.unmarshal(f);

String result="";//in result, we will concat the information from the XML

lBooks=allBooks.getBook();//Create a list of Books with the method getBook()

System.***out***.println("Number of books:"+ lBooks.size());

//Extract the information with the methods of the class Book traversing the list lBooks

**for**(**int** i=0;i<lBooks.size();i++){

result=result+"\n Published in: "+lBooks.get(i).getPublished();

result=result+"\n The Title is: "+lBooks.get(i).getTitle();

result=result+"\n The author is: "+lBooks.get(i).getAuthor()+"\n-------------";

}

System.***out***.println(result);

}**catch**(Exception e) {

e.printStackTrace();

}

}

We will try the code in another class “UseBooksJAXB”:

**import** java.io.File;

**public** **class** UseBooksJAXB {

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

BooksJaxb b=**new** BooksJaxb();

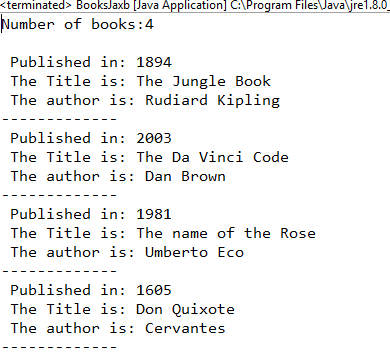
File f=**new** File("Books.xml");

b.openXMLwithJAXBandShow(f);

}

}

The console will show:



Now we can add books to the list. We will write a method to do it.

**public** **void** addBook(String year,String title,String author) {

**try** {

Book book1=**new** Book() ;//First we create an object Book

book1.setAuthor(author);

book1.setTitle(title);

book1.setPublished(year);

//add it to the list of books unmarshalled before

lBooks.add(book1);

//book1 is now added to the list

}**catch**(Exception e){

e.printStackTrace();

}

Finally, we create a method to write our list into a file(marshall):

**public** **void** createXMLfromList(String fileName) {

//MARSHALLING: creating an XML file from a list of Books

**try** {

Marshaller m=contexto.createMarshaller();

m.setProperty(Marshaller.***JAXB\_FORMATTED\_OUTPUT***, Boolean.***TRUE***);//to have indentations and lines

m.marshal(allBooks, System.***out***);//with System.out we will see it in the console

m.marshal(allBooks, **new** FileOutputStream(fileName));//we create the new file

}**catch**(Exception e) {

e.printStackTrace();

}

}

To try this code, we go to class UseBooksJAXB and write some more code:

**import** java.io.File;

**public** **class** UseBooksJAXB {

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

BooksJaxb b=**new** BooksJaxb();

File f=**new** File("Books.xml");

b.openXMLwithJAXBandShow(f);

b.addBook("1930", "Poet in New York","Lorca");

b.addBook("1998","Harry Potter","JKRowking");

b.createXMLfromList("BooksAddedJAXB.xml");

}

}

The result must be the following on the console:

Number of books:4

Published in: 1894

The Title is: The Jungle Book

The author is: Rudiard Kipling

-------------

Published in: 2003

The Title is: The Da Vinci Code

The author is: Dan Brown

-------------

Published in: 1981

The Title is: The name of the Rose

The author is: Umberto Eco

-------------

Published in: 1605

The Title is: Don Quixote

The author is: Cervantes

-------------

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

<Books>

<Book published="1894">

<Title>The Jungle Book</Title>

<Author>Rudiard Kipling</Author>

</Book>

<Book published="2003">

<Title>The Da Vinci Code</Title>

<Author>Dan Brown</Author>

</Book>

<Book published="1981">

<Title>The name of the Rose</Title>

<Author>Umberto Eco</Author>

</Book>

<Book published="1605">

<Title>Don Quixote</Title>

<Author>Cervantes</Author>

</Book>

<Book published="1930">

<Title>Poet in New York</Title>

<Author>Lorca</Author>

</Book>

<Book published="1998">

<Title>Harry Potter</Title>

<Author>JKRowking</Author>

</Book>

</Books>

And we will have our file created:



