

JAXB: средство связывания данных XML для Java

М.Ватник
vtnk@yandex.ru

Java Architecture for XML Binding (JAXB)

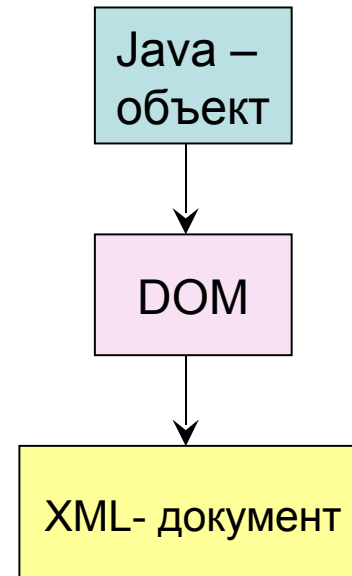
Позволяет связать классы Java с их XML-представлением непосредственно, без преобразований через промежуточные интерфейсы.

Сохранение и передача данных выглядят как сохранение и передача объектов целиком.

Альтернативы

1. Формирование выходной последовательности "вручную"	2. Преобразование через API (например, DOM, JDOM) <ul style="list-style-type: none">•Преимущества: работа с XML-файлом на низком уровне позволяет вручную контролировать вывод в файл•Недостатки: сохраняется большая вероятность ошибки программиста, например неверного перевода строк, неправильной кодировки, и т.д.
--	---

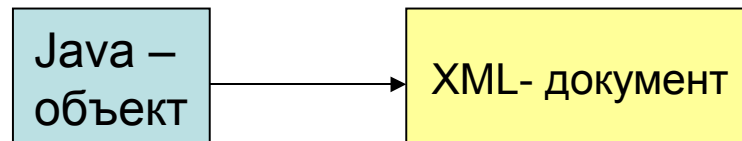
```
String xmlString = setupXMLBuffer(  
    new StringBuffer("<firstName>")  
        .append(customer.firstName)  
        .append("</firstName>")  
        .append("<lastName>")  
        .append(customer.lastName)  
        .append("</lastName>")  
    // etc...  
    .toString()  
);  
bufferedWriter.write(xmlString);  
// other file I/O code
```



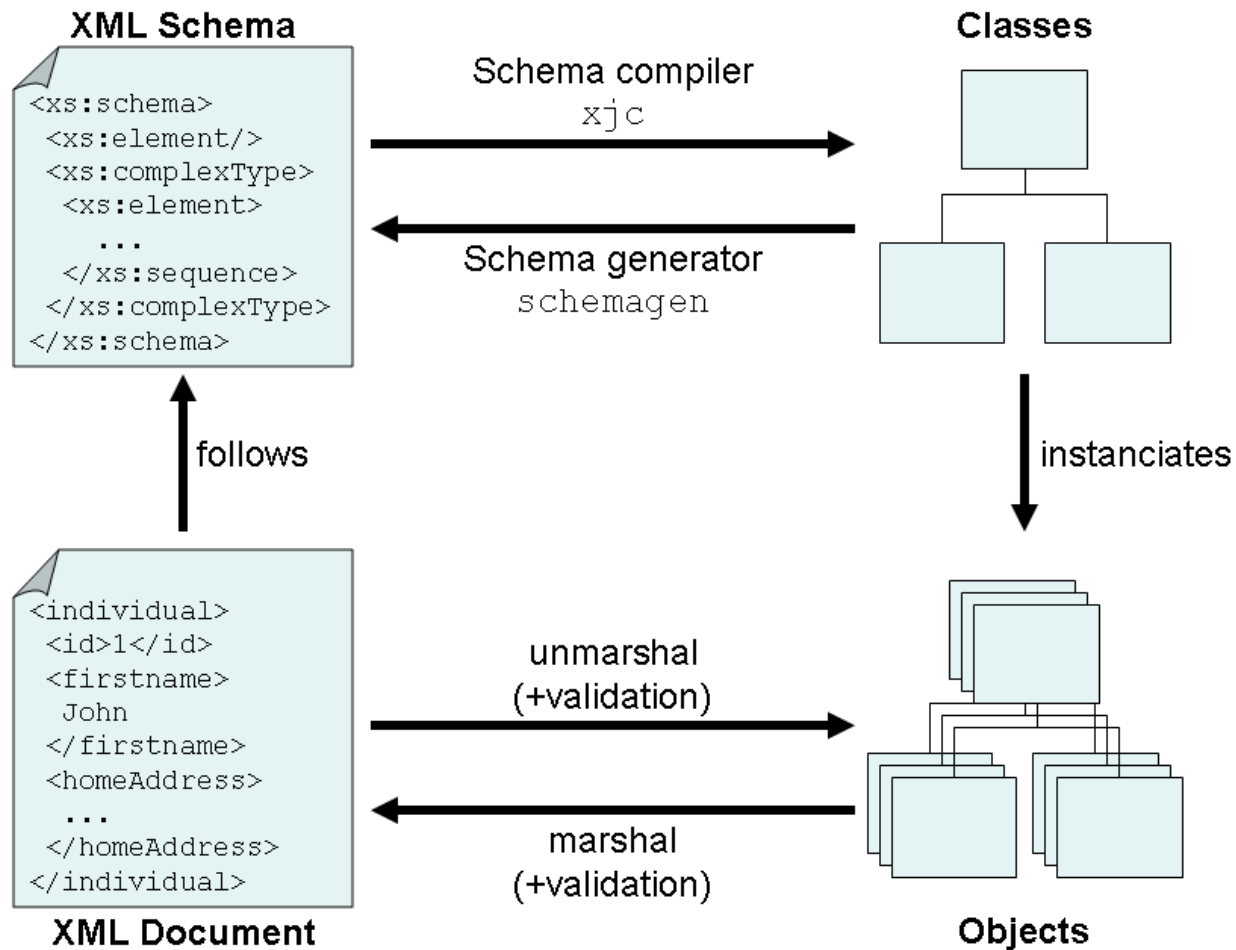
Альтернативы

(продолжение)

- **JAXB.** Позволяет получать XML-формат непосредственно из объекта.
- ❑ **Преимущества:** простота использования, особенно при решении несложных задач. Не требуется глубоких знаний XML.
- ❑ **Недостатки:** разработчик плохо контролирует процесс, все получается "само", поэтому возникает вероятность ошибки, например потери части объектов и полей.

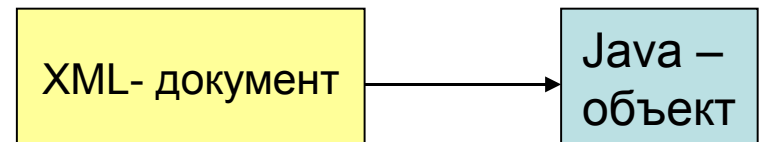
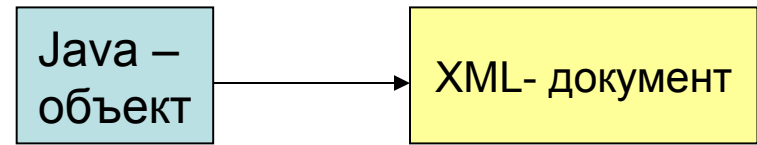


Архитектура JAXB



Маршаллизация и демаршаллизация

- **Маршаллизация** - это процесс преобразования находящихся в памяти данных в формат их хранения.
- **Демаршаллизация** - это процесс преобразования данных из формата среды хранения в память, т.е. процесс, прямо противоположный маршаллизации.



Другие средства связывания данных для Java

- ☐ Eclipse Modeling Framework (EMF)
- ☐ XMLBeans,
- ☐ Hibernate
- ☐ Castor
- ☐ JiBX
- ☐ CookXml
- ☐ XStream

Пример: class Book

```
import java.io.*;
import java.util.logging.*;
import javax.xml.bind.*;
import javax.xml.bind.annotation.XmlRootElement;
```

```
class Author {
    Author();
    Author(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }
    private String firstName;
    private String lastName;
    public String getFirstName() {return firstName;}
    public void setFirstName(String firstName)
        {this.firstName = firstName;}
    public String getLastName() {return lastName;}
    public void setLastName(String lastName)
        {this.lastName = lastName;}
}
```

@XmlRootElement

```
class Book {
    private Author author;
    private String title;
    private int year;
    Book();
    Book(Author author, String title, int year) {
        this.author = author;
        this.title = title;
        this.year = year;
    }
    public Author getAuthor() {return author;}
    public void setAuthor(Author author) {this.author = author;}
    public String getTitle() {return title;}
    public void setTitle(String title){this.title = title;}
    public int getYear() {return year;}
    public void setYear(int year) {this.year = year;}
}
```


Продолжение примера: Маршаллизация

```
public class Main {  
    public static void main(String[] args) {  
        {  
            OutputStream os = null;  
            try {  
                //создаем объект  
                Author pushkin = new Author("Александр", "Пушкин");  
                Book poltava = new Book(pushkin, "Полтава", 2008);  
                //создаем выходной поток  
                File of = new File("book.xml");  
                os = new FileOutputStream(of);  
                // Маршаллизация  
                JAXBContext context = JAXBContext.newInstance(Book.class);  
                Marshaller m = context.createMarshaller();  
                m.marshal(poltava, os);  
                os.close();  
  
                } catch (IOException ex) {  
                    Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);  
                } catch (JAXBException ex) {  
                    Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);  
                }  
            }  
        }  
    }  
}
```

Что получилось?

(файл book.xml)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>  
- <book>  
  - <author>  
    <firstName>Александр</firstName>  
    <lastName>Пушкин</lastName>  
  </author>  
  <title>Полтава</title>  
  <year>2008</year>  
</book>
```

Продолжение примера: Демаршализация

```
public static void main(String[] args) {  
    {  
        try {// Маршаллизация  
            File xmlFile = new File("book.xml");  
            JAXBContext context = JAXBContext.newInstance(Book.class);  
            Unmarshaller u = context.createUnmarshaller();  
            Book book = (Book) u.unmarshal(xmlFile);  
  
            System.out.println(book.getAuthor().getFirstName());  
            System.out.println(book.getAuthor().getLastName());  
            System.out.println(book.getTitle());  
            System.out.println(book.getYear());  
        } catch (JAXBException ex) {  
            Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);  
        }  
    }  
}
```

**Что
получилось?**

Александр
Пушкин
Полтава
2008

Маршаллизация бинарного объекта (изображения GIF 2x2 пикселя)

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

- <book>

- <author>

<firstName>Александр</firstName>

<lastName>Пушкин</lastName>

</author>

<picture>iYBORw0KGqoAAAANSUhEUgAAAAIAAAACCA)

<title>Полтава</title>

7AABYtq0kAAAAEkIEQVR42mNgYC

<year>2008</year>

3D4DwJqEsQBAFa7CfeQt90HAA

</book>

AAAEIFTkSuQmCC</picture>

Литература

- Бретт Маклафлин. Связывание данных на практике: Исследование JAXB, часть 1 (<http://www.ibm.com/developerworks/ru/library/x-pracdb2/index.html>), часть 2 (<http://www.ibm.com/developerworks/ru/library/x-pracdb3/index.html>)
- Antonio Goncalves. Generate an XML Document from an Object Model with JAXB 2 (<http://www.devx.com/Java/Article/34069/0/page/1>)
- Ed Ort and Bhakti Mehta. Java Architecture for XML Binding (JAXB) (<http://java.sun.com/developer/technicalArticles/WebServices/jaxb/>)
- Бретт Маклафлин. XML и Java: Три подхода к сохранению XML (исходники) (<http://www.interface.ru/home.asp?artId=16134>)