**3 ЛАБОРАТОРНАЯ РАБОТА №3**

ТЕХНОЛОГИЯ СВЯЗЫВАНИЯ XML ДАННЫХ

**3.1 Цель работы**

Изучить технологию XML связывания данных с помощью API связывания – JAXB.

**3.2 Краткие теоретические сведения**

Расширяемый язык разметки (Extensible Markup Language, аббревиатура - XML) описывает класс объектов XML document, а также частично описывает работу компьютерных программ, обрабатывающих объекты с данными, реализующими этот класс. XML - это прикладной уровень или усеченная форма SGML, Стандартного обобщенного языка разметки. По своему построению, XML документ является полноценным SGML документом. XML документы состоят из единиц размещения, называемых сущностями, которые содержат разобранные или неразобранные данные.

Разобранные данные состоят из набора символов, часть которых образуют символьные данные, часть–разметку. Разметка образует описание схемы размещения и логической структуры документа. Язык XML дает механизм создания ограничений для указанной схемы размещения и логической структуры.

Для того, чтобы обеспечить проверку корректности XML-документов, необходимо использовать анализаторы, производящие такую проверку и называемые верифицирующими. На сегодняшний день существует два способа контроля правильности XML- документа: DTD - определения(Document Type Definition) и схемы данных(Semantic Schema). В отличии от SGML, определение DTD-правил в XML не является необходимостью, и это обстоятельство позволяет нам создавать любые XML-документы, не ломая пока голову над весьма непростым синтаксисом DTD и XSD.

DTD позволяет задать основную структуру XML-документа. Есть два вида DTD: внешний и внутренний. Внешний файл DTD просто содержит элементы. Для внутреннего DTD эти определения являются частью самого XML-файла. Внешней DTD может определять содержимое многих разных документов, делая их в чем-то более легкими для управления. Однако иногда правильный документ нуждается в том, чтобы подтверждать сам себя. В этом случае нужно включать информацию DTD в сам документ.

**3.3 Выполнение работы**

package ua.cn.stu.model;

import java.util.ArrayList;

import java.util.List;

import javax.xml.bind.annotation.XmlElement;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement

public class TestData {

public TestData() {

super();

}

@XmlElement(name = "TEST")

private List<Test> testList;

public List<Test> getTestList() {

return testList;

}

public void add(Test obj) {

if (this.testList == null)

this.testList = new ArrayList<Test>();

this.testList.add(obj);

}

}

**protected** **void** saveToXML() {

JFileChooser fileCh = **new** JFileChooser("Save xml");

fileCh.showSaveDialog(**this**);

File f = fileCh.getSelectedFile();

List<Test> list = dataCntr.getTestList();

TestData tdata = **new** TestData();

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

tdata.add(list.get(i));

}

JAXBContext jaxbContext;

**try** {

jaxbContext = JAXBContext.*newInstance*(TestData.**class**);

Marshaller jaxbMarshaller = jaxbContext.createMarshaller();

jaxbMarshaller.setProperty(Marshaller.***JAXB\_FORMATTED\_OUTPUT***, **true**);

jaxbMarshaller.marshal(tdata, f);

jaxbMarshaller.marshal(tdata, System.***out***);

} **catch** (JAXBException e) {

e.printStackTrace();

}

}

}

**3.4 Выводы**

В ходе данной лабораторной работы изучена технология XML связывания данных с помощью API связывания – JAXB.