МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

**Отчет по лабораторной работе №6**

по дисциплине

Управление данными

«Знакомство с XML»

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Балашова Т. И.

(подпись)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Спиридонов Д. С.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

17-АС

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород, 2020

**Цель лабораторной работы**:

Изучить работу с языком разметки XML.

**Задание**:

Часть1.

Составить свой корректный XML-документ, описывающий иерархическую базу данных.

Часть 2.

Ознакомится с языком запросов XPath. Написать пару запросов к своему XML документу созданном ранее.

**Выполнение работы:**

**Часть 1**

Описание документа:

<!DOCTYPE maternity\_hospital

[

<!ELEMENT maternity\_hospital (mother+)>

<!ELEMENT mother (full\_name,birthday,number\_hospital\_room,childs)>

<!ATTLIST mother id CDATA #REQUIRED>

<!ELEMENT full\_name (surname,name)>

<!ELEMENT surname (#PCDATA)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT birthday (#PCDATA)>

<!ELEMENT number\_hospital\_room (#PCDATA)>

<!ELEMENT childs (nameChild|birthdayChild)+>

<!ELEMENT nameChild (#PCDATA)>

<!ELEMENT birthdayChild (#PCDATA)>

]>

Корневой элемент *maternity\_hospital*, содержащий вложенный элемент *mother*.  
Атрибутом mother является id, кроме того, он будет содержать вложенные элементы: *full\_name*(имеет вложенные элементы: *surname*, *name*), *birthday*, *number\_hospital\_room*, *childs*(имеет вложенные элементы: *name*, *birthday*).

Составим XML – документ:

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

<!DOCTYPE maternity\_hospital

[

<!ELEMENT maternity\_hospital (mother+)>

<!ELEMENT mother (full\_name,birthday,number\_hospital\_room,childs)>

<!ATTLIST mother id CDATA #REQUIRED>

<!ELEMENT full\_name (surname,name)>

<!ELEMENT surname (#PCDATA)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT birthday (#PCDATA)>

<!ELEMENT number\_hospital\_room (#PCDATA)>

<!ELEMENT childs (nameChild|birthdayChild)+>

<!ELEMENT nameChild (#PCDATA)>

<!ELEMENT birthdayChild (#PCDATA)>

]>

<maternity\_hospital>

<mother id="1">

<full\_name>

<surname>Ivanova</surname>

<name>Ekaterina</name>

</full\_name>

<birthday>12.05.1990</birthday>

<number\_hospital\_room>1</number\_hospital\_room>

<childs>

<nameChild>Ivan</nameChild>

<birthdayChild>29.02.2008</birthdayChild>

<nameChild>Nikita</nameChild>

<birthdayChild>29.02.2008</birthdayChild>

<nameChild>Oleg</nameChild>

<birthdayChild>29.02.2008</birthdayChild>

</childs>

</mother>

<mother id="2">

<full\_name>

<surname>Sidorova</surname>

<name>Anna</name>

</full\_name>

<birthday>06.07.1994</birthday>

<number\_hospital\_room>1</number\_hospital\_room>

<childs>

<nameChild>Alex</nameChild>

<birthdayChild>31.01.2015</birthdayChild>

</childs>

</mother>

<mother id="3">

<full\_name>

<surname>Naumova</surname>

<name>Svetlana</name>

</full\_name>

<birthday>06.07.1991</birthday>

<number\_hospital\_room>2</number\_hospital\_room>

<childs>

<nameChild>Kirill</nameChild>

<birthdayChild>22.08.2016</birthdayChild>

</childs>

</mother>

</maternity\_hospital>

Часть 2

**Код:**

import org.w3c.dom.Document;

import org.w3c.dom.NodeList;

import org.xml.sax.SAXException;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import java.io.IOException;

import java.util.ArrayList;

import javax.xml.xpath.XPath;

import javax.xml.xpath.XPathFactory;

import javax.xml.xpath.XPathConstants;

import javax.xml.xpath.XPathExpression;

import javax.xml.xpath.XPathExpressionException;

public class Main {

public static void main(String[] args) {

DocumentBuilderFactory builderFactory = DocumentBuilderFactory.newInstance();

builderFactory.setNamespaceAware(true);

DocumentBuilder builder;

try {

builder = builderFactory.newDocumentBuilder();

Document doc = builder.parse("lab.xml");

XPathFactory pathFactory = XPathFactory.newInstance();

XPath xpath = pathFactory.newXPath();

ArrayList<ArrayList> name = getMotherNameByNumberHospitalRoom(doc, xpath, "1");

System.out.println("В первой палате лежат:");

for (int i=0;i < name.get(0).size();i++)

System.out.println(name.get(0).get(i)+ " " + name.get(1).get(i));

System.out.println();

ArrayList<ArrayList> child = getChildByMotherName(doc, xpath, "Ivanova");

System.out.println("Дети Ивановой:");

for (int i=0;i < child.get(0).size();i++)

System.out.println(child.get(0).get(i)+ " " + child.get(1).get(i));

System.out.println();

} catch (ParserConfigurationException | IOException | SAXException e) {

e.printStackTrace();

}

}

public static ArrayList<ArrayList> getMotherNameByNumberHospitalRoom(Document doc, XPath xpath, String number) {

ArrayList<String> surname = new ArrayList<>();

ArrayList<String> name = new ArrayList<>();

ArrayList<ArrayList> fullName = new ArrayList<>();

try {

XPathExpression xExp = xpath.compile(

"//mother[number\_hospital\_room='"+number+"']/full\_name/surname/text()");

XPathExpression xExp1 = xpath.compile(

"//mother[number\_hospital\_room='"+number+"']/full\_name/name/text()");

NodeList surnameNode = (NodeList) xExp.evaluate(doc, XPathConstants.NODESET);

NodeList nameNode = (NodeList) xExp1.evaluate(doc, XPathConstants.NODESET);

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

surname.add(surnameNode.item(i).getTextContent());

name.add(nameNode.item(i).getTextContent());

}

fullName.add(surname);

fullName.add(name);

} catch (XPathExpressionException e) {

e.printStackTrace();

}

return fullName;

}

public static ArrayList<ArrayList> getChildByMotherName(Document doc, XPath xpath, String surname) {

ArrayList<String> birthday = new ArrayList<>();

ArrayList<String> name = new ArrayList<>();

ArrayList<ArrayList> fullName = new ArrayList<>();

try {

XPathExpression xExp = xpath.compile(

"//mother[full\_name[surname='"+surname+"']]/childs/nameChild/text()");

XPathExpression xExp1 = xpath.compile(

"//mother[full\_name[surname='"+surname+"']]/childs/birthdayChild/text()");

NodeList nameNode = (NodeList) xExp.evaluate(doc, XPathConstants.NODESET);

NodeList birthdayNode = (NodeList) xExp1.evaluate(doc, XPathConstants.NODESET);

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

name.add(nameNode.item(i).getTextContent());

birthday.add(birthdayNode.item(i).getTextContent());

}

fullName.add(name);

fullName.add(birthday);

} catch (XPathExpressionException e) {

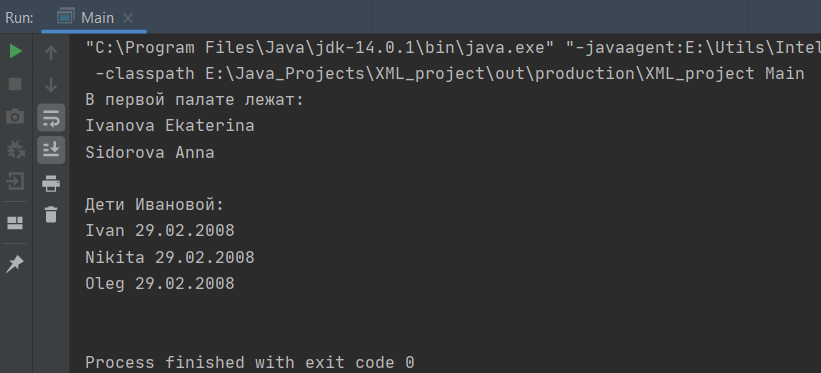
e.printStackTrace();

}

return fullName;

}

}

**Результаты работы:**

**Вывод:**

В процессе выполнения лабораторной работы были изучены основы написания валидных xml-документов, а также получены базовые навыки в работе с языком запросов XPath.