ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Факультет кибербезопасности и управления

Кафедра программной инженерии

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №** 1

по дисциплине Прикладное программирование

XML

|  |  |  |
| --- | --- | --- |
| **ВЫПОЛНИЛ** | | |
| студент | РПИС-11  (группа) | Яковлев П. Н.  (ФИО) |
|  | **ПРОВЕРИЛ** |  |
| Ассистент  (должность) | | (ФИО) |

Самара

2023

**Задание на практическую работу**

В процессе написания тестовых заданий ознакомиться с элементами языка XML и средствами Java для работы с XML-документами.

**Задание 1**

Изучить предлагаемый файл описания типа документа сформировать документ согласно этим правилам.

--- group.dtd --------------------------------------------------

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

**<!ELEMENT group (student\*)>**

**<!ELEMENT student (subject\*,average?)>**

**<!ATTLIST student**

**firstname CDATA #REQUIRED**

**lastname CDATA #REQUIRED**

**groupnumber CDATA #REQUIRED**

**>**

**<!ELEMENT subject EMPTY>**

**<!ATTLIST subject**

**title CDATA #REQUIRED**

**mark (1|2|3|4|5) #REQUIRED**

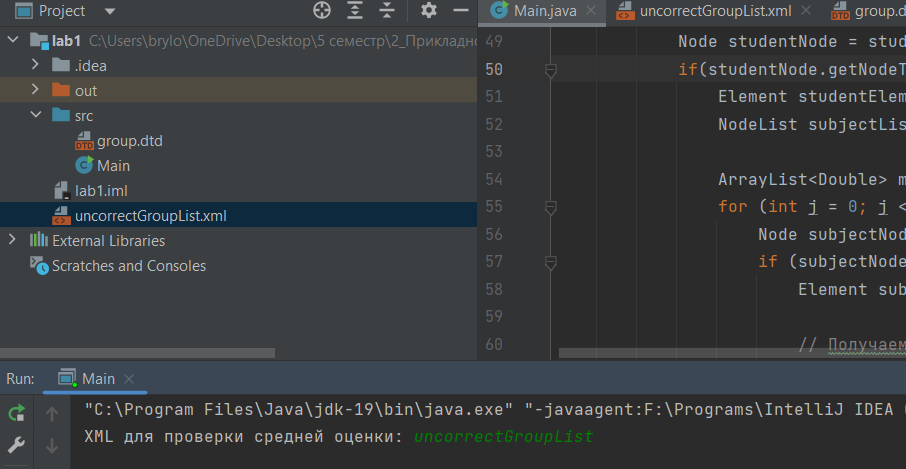
**>**

**<!ELEMENT average (#PCDATA)>**

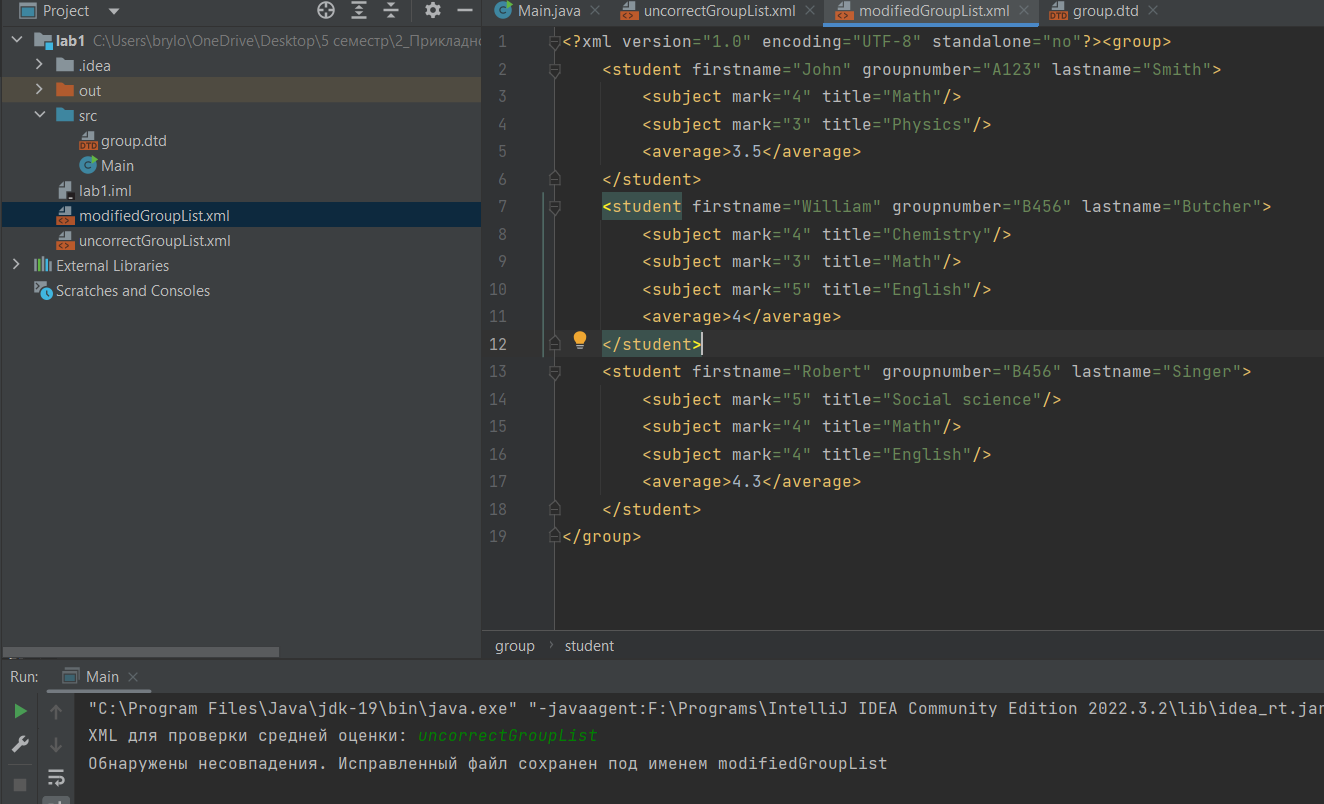
----------------------------------------------------------------

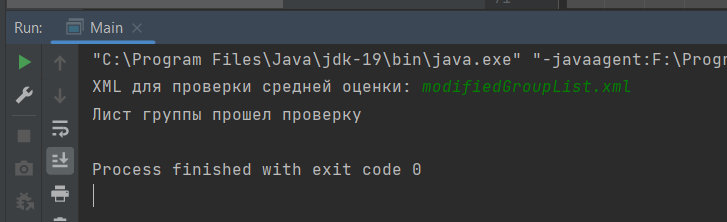
Документ сформированный по правилам:  


Средняя оценка (<average>) в данном документе записана неверно. Для расчета и корректировки написана программа. Результат работы программы:



Если обнаружены несовпадения в оценке, создается файл modifiedGroupList.xml, который содержит исправленный вариант оценки:



Если всё верно, программа выведет соответствующее сообщение:  


Листинг программы:

import org.w3c.dom.\*;

import org.xml.sax.SAXException;

import javax.xml.parsers.\*;

import javax.xml.transform.Transformer;

import javax.xml.transform.TransformerConfigurationException;

import javax.xml.transform.TransformerException;

import javax.xml.transform.TransformerFactory;

import javax.xml.transform.dom.DOMSource;

import javax.xml.transform.stream.StreamResult;

import java.io.File;

import java.io.IOException;

import java.text.DecimalFormat;

import java.text.DecimalFormatSymbols;

import java.util.ArrayList;

import java.util.Locale;

import java.util.Scanner;

public class Main {

static Document xmlToCheck;

public static void main(String[] args) throws ParserConfigurationException, IOException, SAXException {

Scanner scanner = new Scanner(System.in);

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder builder = factory.newDocumentBuilder();

System.out.print("XML для проверки средней оценки: ");

String readed = scanner.nextLine();

if(!readed.endsWith(".xml"))

readed += ".xml";

xmlToCheck = builder.parse(readed);

Checker(xmlToCheck);

}

static void Checker(Document document){

boolean filemistake = false;

NodeList studentList = document.getElementsByTagName("student");

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

Node studentNode = studentList.item(i);

if(studentNode.getNodeType() == Node.ELEMENT\_NODE){

Element studentElement = (Element) studentNode;

NodeList subjectList = studentElement.getElementsByTagName("subject");

ArrayList<Double> marks = new ArrayList<>();

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

Node subjectNode = subjectList.item(j);

if (subjectNode.getNodeType() == Node.ELEMENT\_NODE) {

Element subjectElement = (Element) subjectNode;

// Получаем значение атрибута 'mark' элемента <subject>

String mark = subjectElement.getAttribute("mark");

marks.add(Double.parseDouble(mark));

}

}

Element averageElement = (Element) studentElement.getElementsByTagName("average").item(0);

double average = Double.parseDouble(averageElement.getTextContent());

double mSum = 0;

for (double m: marks) {

mSum+=m;

}

DecimalFormat decimalFormat = new DecimalFormat("#.#", DecimalFormatSymbols.getInstance(Locale.US));

double avgCalc = Double.parseDouble(decimalFormat.format((mSum/marks.size())));

if(average!=avgCalc){

averageElement.setTextContent(String.valueOf(decimalFormat.format(avgCalc)));

}

filemistake = average != avgCalc;

}

}

if(filemistake){

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transformer = null;

try {

transformer = transformerFactory.newTransformer();

DOMSource source = new DOMSource(document);

StreamResult result = new StreamResult(new File("modifiedGroupList.xml"));

transformer.transform(source, result);

} catch (TransformerException e) {

throw new RuntimeException(e);

}

System.out.println("Обнаружены несовпадения. Исправленный файл сохранен под именем modifiedGroupList");

} else System.out.println("Лист группы прошел проверку");

}

}