**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Владимирский государственный университет**

**имени Александра Григорьевича и Николая Григорьевича Столетовых»**

**(ВлГУ)**

Кафедра информационных систем и программной инженерии

**Лабораторная работа №5**

**ОПРЕДЕЛЕНИЕ СТРУКТУРЫ JSON-ДОКУМЕНТА СРЕДСТВАМИ JSON SCHEMA. ПРОГРАММНАЯ ОБРАБОТКА JSON**

Выполнил: студент группы ПРИм-121

Евстигнеев Андрей Васильевич

Проверил: преподаватель кафедры ИСПИ

Спирин Иван Вячеславович

Владимир 2022

**ЦЕЛЬ РАБОТЫ**

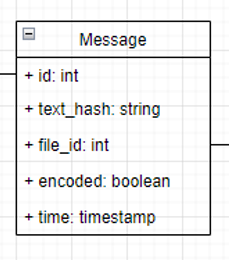
Разработать программу, использующую один из рассмотренных инструментов для обработки XML-документа. Обработка должна заключаться в формировании JSON-представления для XML-документа, поданного на вход программы. Структура JSON-документа должна быть предварительно определена с помощью JSON Schema. В процессе обработки должно изменяться содержимое и структура полученного при анализе графа объектов, JSON-представление должно быть сформировано с учётом изменений.

**ХОД РАБОТЫ**

Была разработана программа на основе ЛР №4.

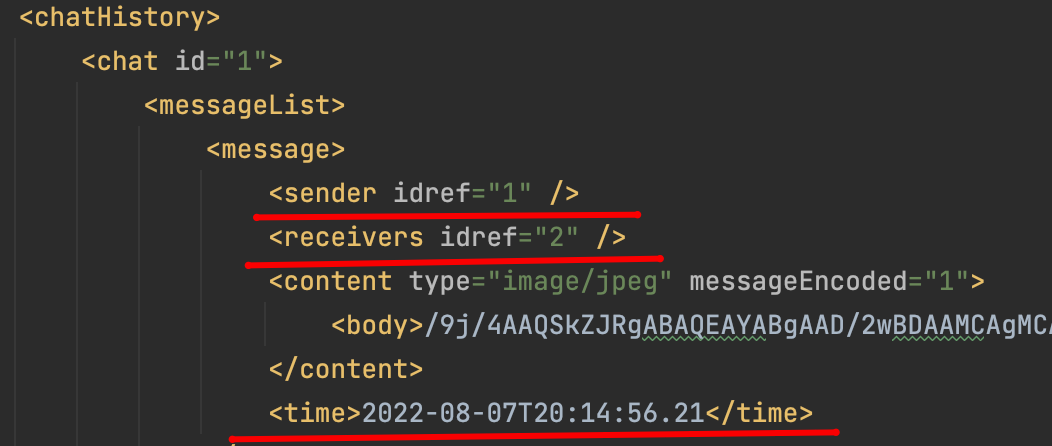
Соответствие требованиям:

1) предметная область должна предусматривать наличие данных трех типов: строкового, числового и типа дата



На примере сущности «Message» есть тип int – id, тип String – текст сообщения или изображение в формате Base64, тип Date – дата сообщения.

2) структура и наполнение XML-документа должны быть такими, чтобы часть данных можно было представить в табличной форме, и при этом в каждой строке таблицы было как минимум два числовых поля и как минимум одно поле типа дата



3) результат формирования JSON-представления должен содержать поле с как минимум одним итогом – результат агрегации строк табличной части; итоги могут вычисляться любым способом, но осмысленно в терминах предметной области



4) ввод каждая из программ должна брать со стандартного ввода, результат преобразования передавать на стандартный вывод

XMLStreamReader reader = XMLInputFactory.*newInstance*().createXMLStreamReader(Files.*newInputStream*(Paths.*get*("./src/main/resources/example2.xml")));

5) каждая из программ должна обеспечить обработку ошибок анализа документа соответствующими средствами, при этом в случае ошибки уровня error и fatal error анализ должен быть прекращен без выдачи результата преобразования на стандартный вывод, а при возникновении ошибки уровня warning анализ должен быть продолжен, а преобразование - выполнено; в любом случае при возникновении ошибки любого уровня необходимо на стандартный вывод об ошибках выдать информацию об уровне ошибки, системное сообщение об ошибке и место возникновения ошибки с указанием строки и столбца документа

try (FileWriter writer = new FileWriter("myjson.json", false)) {  
 writer.write(new Gson().toJson(systemXML));  
 writer.flush();  
} catch (IOException ex) {  
 System.*out*.println(ex.getMessage());  
}

json-схема:

{  
 "type": "object",  
 "properties": {  
 "personList": {  
 "type": "object",  
 "properties": {  
 "personList": {  
 "type": "object",  
 "properties": [  
 {  
 "firstName": {  
 "type": "string"  
 },  
 "lastName": {  
 "type": "string"  
 },  
 "pnoneNumber": {  
 "type": "string"  
 },  
 "id": {  
 "type": "integer"  
 }  
 }  
 ]  
 }  
 }  
 },  
 "chatHistory": {  
 "type": "object",  
 "properties": {  
 "chatList": {  
 "type": "object",  
 "properties": [  
 {  
 "id": {  
 "type": "integer"  
 },  
 "messageList": {  
 "type": "object",  
 "properties": {  
 "messageList": {  
 "type": "object",  
 "properties": [  
 {  
 "sender": {  
 "type": "object",  
 "properties": {  
 "idref": {  
 "type": "integer"  
 }  
 }  
 },  
 "receivers": {  
 "type": "object",  
 "properties": {  
 "idref": {  
 "type": "integer"  
 }  
 }  
 },  
 "time": {  
 "type": "string",  
 "format": "date"  
 },  
 "content": {  
 "type": "object",  
 "properties": {  
 "type": {  
 "type": "integer"  
 },  
 "messageEncoded": {  
 "type": "boolean"  
 },  
 "body": {  
 "type": "integer"  
 }  
 }  
 }  
 }  
 ]  
 }  
 }  
 },  
 "type": {  
 "type": "object",  
 "properties": {  
 "aPrivate": {  
 "type": "object",  
 "properties": {  
 "dateOfCreation": {  
 "type": "string",  
 "format": "date"  
 }  
 }  
 }  
 }  
 }  
 }  
 ]  
 }  
 }  
 }  
 }  
}

json-представление:

{  
 "personList": {  
 "personList": [  
 {  
 "firstName": "Anton",  
 "lastName": "Antonov",  
 "pnoneNumber": "12345",  
 "id": 1  
 },  
 {  
 "firstName": "Alena",  
 "lastName": "Kuzmina",  
 "pnoneNumber": "89307306455",  
 "id": 2  
 },  
 {  
 "firstName": "Dima",  
 "lastName": "Kovanov",  
 "pnoneNumber": "89304320665",  
 "id": 3  
 }  
 ],  
 "result": 3  
 },  
 "chatHistory": {  
 "chatList": [  
 {  
 "id": 1,  
 "messageList": {  
 "messageList": [  
 {  
 "sender": {  
 "idref": 1  
 },  
 "receivers": {  
 "idref": 2  
 },  
 "time": "Aug 7, 2022, 8:14:56 PM",  
 "content": {  
 "type": "image/jpeg",  
 "messageEncoded": true,  
 "body": "\u003d"  
 }  
 },  
 {  
 "sender": {  
 "idref": 2  
 },  
 "receivers": {  
 "idref": 1  
 },  
 "time": "Aug 7, 2022, 8:15:56 PM",  
 "content": {  
 "type": "image/jpeg",  
 "messageEncoded": false,  
 "body": "\u003d"  
 }  
 }  
 ]  
 },  
 "type": {  
 "aPrivate": {  
 "dateOfCreation": "Oct 14, 2022, 6:33:07 PM"  
 }  
 }  
 },  
 {  
 "id": 2,  
 "messageList": {  
 "messageList": [  
 {  
 "sender": {  
 "idref": 1  
 },  
 "receivers": {  
 "idref": 2  
 },  
 "time": "May 23, 2022, 5:39:10 AM",  
 "content": {  
 "body": "NEW MESSAGE"  
 }  
 }  
 ]  
 },  
 "type": {  
 "group": {  
 "dateOfCreation": "Mar 15, 2022, 6:14:44 AM",  
 "title": "PRIm-121",  
 "admins": {  
 "idref": 1  
 }  
 }  
 }  
 }  
 ]  
 },   
}