**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 №2**

**ОСНОВЫ JAXP. ПРОГРАММНЫЙ АНАЛИЗ XML-ДОКУМЕНТОВ**

Выполнил: студент группы ПРИм-121

Евстигнеев Андрей Васильевич

Проверил: преподаватель кафедры ИСПИ

Спирин Иван Вячеславович

Владимир 2022

**ЦЕЛЬ РАБОТЫ**

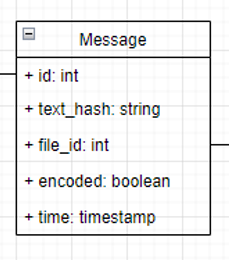
Разработать программу анализа XML-документ, составленного в соответствии со схемой, разработанной в ЛР №1. Суть анализа должна заключаться в формировании HTML-представления XML-документа, поданного программе на вход.

**ХОД РАБОТЫ**

Была разработана программа, использующая средства StAX API.

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 HTML-представления должен содержать в табличной части подвал с как минимум одним итогом; 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

boolean ok = *schemaValidator*();  
if (ok) {  
 File file = new File(Objects.*requireNonNull*(XMLstaxParser.class.getClassLoader().getResource("example2.xml")).getFile());  
 Systema system = *parseXMLFile*(file);  
 *htmlGen*(system);  
}

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

} catch (FileNotFoundException | XMLStreamException exc) {  
 exc.printStackTrace();  
} catch (ParseException e) {  
 e.printStackTrace();  
}

6) при анализе документа необходимо выполнить проверку соответствия документа схеме, при этом предполагается, что файл со схемой поставляется вместе с программой и должен быть расположен в каталоге программы, из которого она и запускается, а анализируемый файл может и вовсе не содержать ссылки на схему

try {  
 SchemaFactory factory = SchemaFactory.*newInstance*(XMLConstants.*W3C\_XML\_SCHEMA\_NS\_URI*);  
 Schema schema = factory.newSchema(new File(".\\src\\main\\resources\\XMLScheme.xsd"));  
 Validator validator = schema.newValidator();  
 XMLStreamReader reader = XMLInputFactory.*newInstance*().createXMLStreamReader(Files.*newInputStream*(Paths.*get*(".\\src\\main\\resources\\example2.xml")));  
 validator.validate(new StAXSource(reader));  
  
 System.*out*.println("XML is valid");  
 return true;  
} catch (Exception e) {  
 System.*out*.println("XML is not valid");  
 return false;  
}