**ЛАБОРАТОРНАЯ РАБОТА №9**

**XML: ОСНОВЫ СОЗДАНИЯ**

**И ОБРАБОТКИ СТРУКТУРИРОВАННЫХ ДАННЫХ**

**Цель работы:** освоить принципы создания, валидации и обработки XML

документов, развить навыки анализа данных и работы с инструментами для

обработки структурированных форматов.

**Задание:**

1. Создание XML-документа:

* Напишите XML-документ, описывающий библиотеку (не менее 3 книг).
* Сохраните его в файл с именем library.xml.
* Убедитесь, что документ соответствует синтаксису XML (правильное закрытие тегов, корректные атрибуты).

2. Создание XSD-схемы:

* Разработайте XSD-схему для валидации library.xml.
* Схема должна проверять:
* Обязательное наличие всех указанных элементов (title, author,
* year, genre, price).
* Атрибут id — уникальный и обязательный.
* year — целое число от 1800 до текущего года.
* price — положительное число с двумя знаками после запятой.
* Сохраните схему в файл library.xsd.

3. Валидация XML-документа:

* Используйте любой инструмент или библиотеку (например, онлайн валидатор, Python с lxml, Java с JAXB) для проверки library.xml на соответствие library.xsd.
* Исправьте ошибки, если они есть, чтобы документ стал валидным.

4. Обработка XML-документа:

* Напишите программу на выбранном языке программирования,

которая:

* Читает library.xml.
* Выводит список всех книг с их данными в консоль (название, автор, год, жанр, цена).
* Вычисляет среднюю цену книг и выводит её.
* Фильтрует книги по заданному критерию (например, жанр или год) и выводит результат.

5. Исследование и сравнение

Найдите в интернете или библиотеке пример реального XML документа (например, экспорт данных из программы, RSS-ленты, конфигурационного файла). Сохраните его как sample.xml.

Напишите краткий анализ (5-7 предложений) структуры этого документа: какие теги используются, какова их вложенность, есть ли атрибуты. Ответьте, для чего, по вашему мнению, этот документ используется.

Library.xml:

<?xml version="1.0" encoding="UTF-8"?>  
<library>  
 <book id="1">  
 <title>Зеленая миля</title>  
 <author>Стивен Кинг</author>  
 <year>1996</year>  
 <genre>Драма</genre>  
 <price>15.99</price>  
 </book>  
 <book id="2">  
 <title>Убийство на поле для гольфа</title>  
 <author>Агата Кристи</author>  
 <year>1923</year>  
 <genre>Детектив</genre>  
 <price>10.49</price>  
 </book>  
 <book id="3">  
 <title>Золотая рыбка</title>  
 <author>Пушкин</author>  
 <year>1833</year>  
 <genre>Сказка</genre>  
 <price>8.99</price>  
 </book>  
</library>

Library.xcd

<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="library">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="book" maxOccurs="unbounded">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element name="title" type="xs:string"/>  
 <xs:element name="author" type="xs:string"/>  
 <xs:element name="year">  
 <xs:simpleType>  
 <xs:restriction base="xs:int">  
 <xs:minInclusive value="1800"/>  
 <xs:maxInclusive value="2025"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="genre" type="xs:string"/>  
 <xs:element name="price">  
 <xs:simpleType>  
 <xs:restriction base="xs:decimal">  
 <xs:minInclusive value="0.01"/>  
 <xs:fractionDigits value="2"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute name="id" type="xs:string" use="required"/>  
 </xs:complexType>  
 </xs:element>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
</xs:schema>

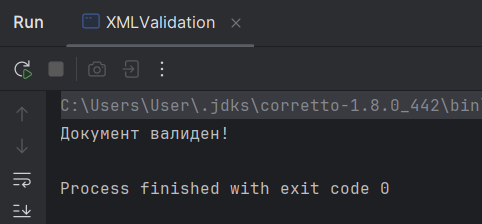


Рисунок 1 – работа документа XMLValidation

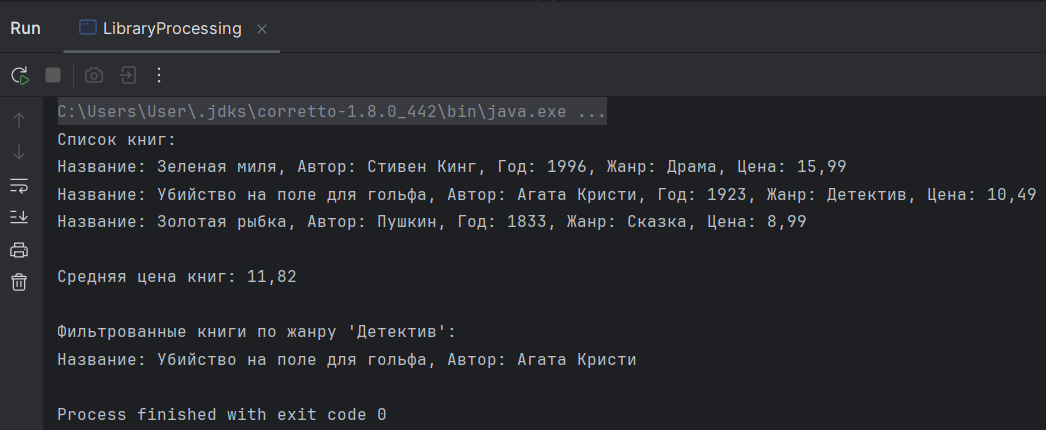


Рисунок 2 - работа документа LibraryProcessing

Пример XML-документа от службы прогноза погоды из Интернета:

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

<current\_observation>

<credit>Национальная служба прогноза погоды NOAA</credit>

<credit\_URL>http://weather.gov/</credit\_URL>

<image>

<url>http://weather.gov/images/xml\_logo.gif</url>

<title>Национальная служба прогноза погоды NOAA</title>

<link>http://weather.gov</link>

</image>

<location>Нью Йорк/Аэропорт им. Джона Кеннеди, NY</location>

<station\_id>KJFK</station\_id>

<latitude>40.66</latitude>

<longitude>-73.78</longitude>

<observation\_time\_rfc822>Mon, 11 Feb 2008 06:51:00 -0500 EST</observation\_time\_rfc822>

<weather>Небольшая облачность</weather>

<temp\_f>11</temp\_f>

<temp\_c>-12</temp\_c>

<relative\_humidity>36</relative\_humidity>

<wind\_dir>Западный</wind\_dir>

<wind\_degrees>280</wind\_degrees>

<wind\_mph>18.4</wind\_mph>

<wind\_gust\_mph>29</wind\_gust\_mph>

<pressure\_mb>1023.6</pressure\_mb>

<pressure\_in>30.23</pressure\_in>

<dewpoint\_f>-11</dewpoint\_f>

<dewpoint\_c>-24</dewpoint\_c>

<windchill\_f>-7</windchill\_f>

<windchill\_c>-22</windchill\_c>

<visibility\_mi>10.00</visibility\_mi>

<icon\_url\_base>http://weather.gov/weather/images/fcicons/</icon\_url\_base>

<icon\_url\_name>nfew.jpg</icon\_url\_name>

<disclaimer\_url>http://weather.gov/disclaimer.html</disclaimer\_url>

<copyright\_url>http://weather.gov/disclaimer.html</copyright\_url>

</current\_observation>

**Анализ структуры XML-документа:**

1. **Корневой элемент**: <current\_observation> содержит все данные о текущих погодных условиях. Вложенность преимущественно одноуровневая, за исключением блока <image> (3 тега внутри).
2. **Основные теги**: Документ использует специализированные метеорологические теги:
   * Географические (location, station\_id, latitude, longitude)
   * Временные (observation\_time\_rfc822)
   * Погодные параметры (weather, temp\_f/c, wind\_dir, pressure\_mb и др.)
3. **Атрибуты**: Отсутствуют (все данные переданы через содержимое тегов).
4. **Ссылочные элементы**: Содержит URL-ссылки на иконки погоды (icon\_url\_base), логотипы и юридические документы (disclaimer\_url).

**Предназначение документа**: Этот XML-файл представляет собой стандартизированный формат передачи текущих метеоданных от Национальной службы погоды NOAA. Он используется для:

* Публикации актуальных наблюдений с метеостанции KJFK (Нью-Йорк)
* Интеграции с веб-сайтами и мобильными приложениями
* Автоматизированной обработки погодных данных системами
* Визуализации данных через привязанные иконки (icon\_url\_name)