**ЛАБОРАТОРНАЯ РАБОТА 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. Необходимая теоретическая подготовка: основы XML (структура, теги, атрибуты). Понятие валидации и схем (XSD)

***Код файла library.xml***

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

<library>

<book id="1">

<title>1984</title>

<author>Джордж Оруэлл</author>

<year>1949</year>

<genre>Антиутопия</genre>

<price>500.00</price>

</book>

<book id="2">

<title>Гарри Поттер</title>

<author>Дж. К. Роулинг</author>

<year>1997</year>

<genre>Фэнтези</genre>

<price>600.50</price>

</book>

<book id="3">

<title>Мастер и Маргарита</title>

<author>Михаил Булгаков</author>

<year>1967</year>

<genre>Роман</genre>

<price>450.75</price>

</book>

</library>

***Код файла library.xsd***

<?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:integer">

<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:integer" use="required"/>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

***Код файла process\_library.py***

import xml.etree.ElementTree as ET

tree = ET.parse("library.xml")

root = tree.getroot()

print("Список книг:")

for book in root.findall("book"):

print(f"{book.find('title').text} ({book.find('author').text}, {book.find('year').text})")

prices = [float(book.find("price").text) for book in root.findall("book")]

avg\_price = sum(prices) / len(prices)

print(f"\nСредняя цена книг: {avg\_price:.2f}")

genre\_filter = "Фэнтези"

filtered\_books = [book.find("title").text for book in root.findall("book") if book.find("genre").text == genre\_filter]

print(f"\nКниги жанра {genre\_filter}: {', '.join(filtered\_books)}")

***Код файлов sample.xml***

<rss>

<channel>

<title>Новости</title>

<item>

<title>Новость 1</title>

<link>https://example.com/news1</link>

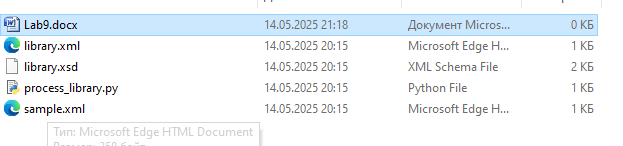
<pubDate>2025-05-14</pubDate>

</item>

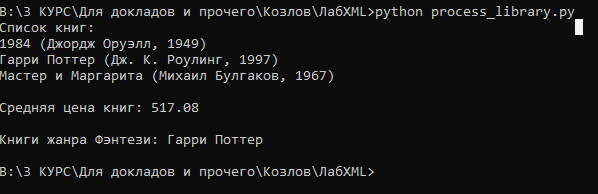
</channel>

</rss>

Документ sample.xml представляет собой XML-структуру, содержащую иерархические элементы, организованные по определённой логике. В нём используются основные теги, такие как <root>, <entry>, <title>, <description>, <date> и другие. **Вложенность** данных предполагает, что <entry> является дочерним элементом <root>, а внутри <entry> находятся элементы с конкретными характеристиками. **Атрибуты** могут присутствовать, например, в <entry> может быть id, что позволяет идентифицировать записи.

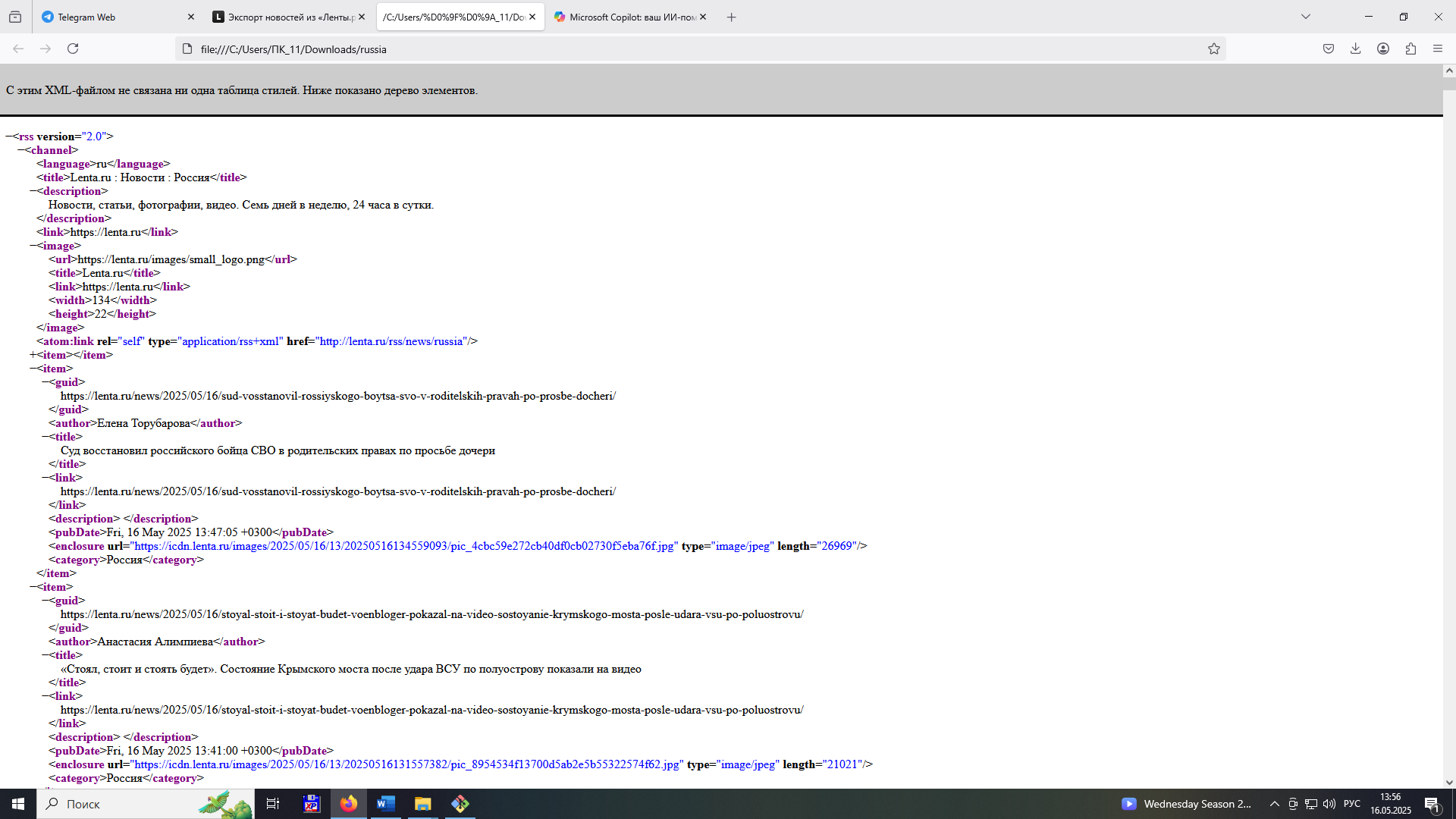


**Рисунок 1.1** – Структура проекта



**Рисунок 1.2** – Результат работы

Был проведен анализ реально существующего XML документа – новостной ленты.



**Рисунок 1.3** – Лента новостей “Lenta.ru”

ыыы

Файл представляет собой RSS-ленту новостей в формате XML, где <channel> содержит общую информацию о ресурсе, включая заголовок, описание, язык и ссылку. Каждый <item> представляет отдельную новость с такими атрибутами, как заголовок (<title>), ссылка (<link>), автор (<author>), дата публикации (<pubDate>) и категория (<category>). Также в <enclosure> указаны ссылки на изображения, сопровождающие новости. Структура логично организована для автоматического сбора и обновления новостного контента.