**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.

ЗАДАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ

1. Создание XML-документа:

o Напишите XML-документ, описывающий библиотеку (не менее 3

книг).

o Сохраните его в файл с именем library.xml.

o Убедитесь, что документ соответствует синтаксису XML

(правильное закрытие тегов, корректные атрибуты).

2. Создание XSD-схемы:

o Разработайте XSD-схему для валидации library.xml.

o 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.

o Сохраните схему в файл library.xsd.

3. Валидация XML-документа:

o Используйте любой инструмент или библиотеку (например, онлайн-

валидатор, Python с lxml, Java с JAXB) для проверки library.xml на

соответствие library.xsd.

o Исправьте ошибки, если они есть, чтобы документ стал валидным.

4. Обработка XML-документа:

o Напишите программу на выбранном языке программирования,

которая:

▪ Читает library.xml.

▪ Выводит список всех книг с их данными в консоль (название,

автор, год, жанр, цена).

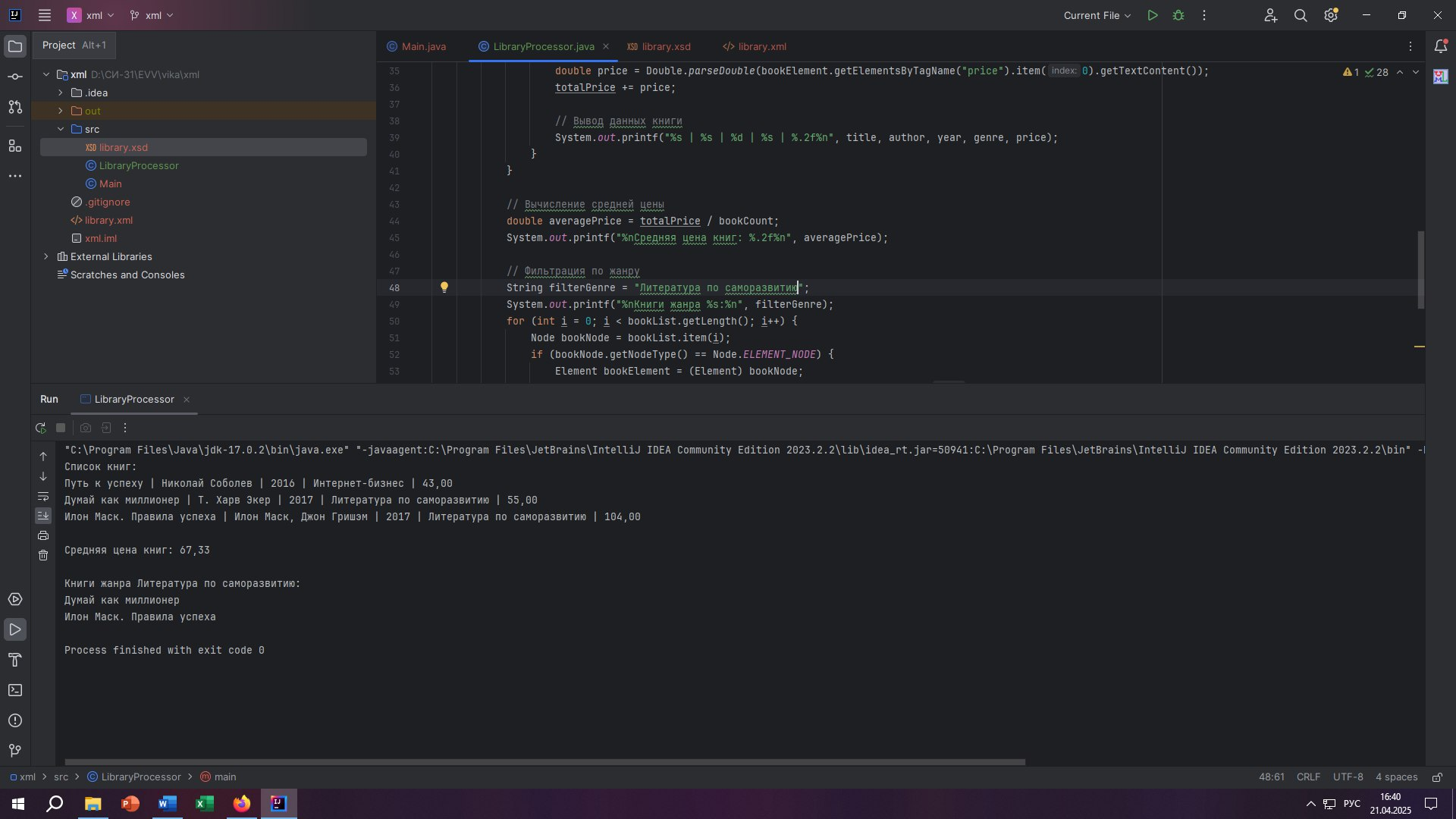
▪ Вычисляет среднюю цену книг и выводит её.

▪ Фильтрует книги по заданному критерию (например, жанр или

год) и выводит результат.

После выполнения получается программа которая :

* Читает XML-документ library.xml с помощью DocumentBuilder.
* Выводит список всех книг с их данными (название, автор, год, жанр, цена).
* Вычисляет среднюю цену книг.
* Фильтрует книги по жанру (Литература по саморазвитию)



**Library.xsd**

<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" type="xs:integer"/>

<xs:element name="genre" type="xs:string"/>

<xs:element name="price" type="xs:decimal"/>

</xs:sequence>

<xs:attribute name="id" type="xs:string" use="required"/>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

**Library.xml**

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

<library>

<book id="1">

<title>Путь к успеху</title>

<author>Николай Соболев</author>

<year>2016</year>

<genre>Интернет-бизнес</genre>

<price>43</price>

</book>

<book id="2">

<title>Думай как миллионер</title>

<author>Т. Харв Экер</author>

<year>2017</year>

<genre>Литература по саморазвитию</genre>

<price>55</price>

</book>

<book id="3">

<title>Илон Маск. Правила успеха</title>

<author>Илон Маск, Джон Гришэм</author>

<year>2017</year>

<genre>Литература по саморазвитию</genre>

<price>104</price>

</book>

</library>

XML-документа — RSS-лента, который можно найти, например, на W3Schools.

Пример XML-документа sample.xml:

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

<rss version="2.0">

<channel>

<title>W3Schools Home Page</title>

<link>https://www.w3schools.com</link>

<description>Free web building tutorials</description>

<item>

<title>RSS Tutorial</title>

<link>https://www.w3schools.com/xml/xml\_rss.asp</link>

<description>New RSS tutorial on W3Schools</description>

</item>

<item>

<title>XML Tutorial</title>

<link>https://www.w3schools.com/xml</link>

<description>New XML tutorial on W3Schools</description>

</item>

</channel>

</rss>

Анализ структуры:

В этом документе корневым элементом является <rss>, который содержит атрибут version="2.0", определяющий версию используемого стандарта RSS. Внутри <rss> располагается элемент <channel>, являющийся контейнером для основных данных канала, таких как <title>, <link> и <description>. Элемент <channel> дополнительно включает несколько элементов <item>, каждый из которых описывает отдельную новостную запись или статью — внутри <item> используются теги <title>, <link> и <description>, что создаёт четкую вложенность данных. Структура документа демонстрирует иерархию, где информационные блоки сгруппированы от общего к частному. Атрибуты в данном примере встречаются только у корневого элемента, что характерно для многих XML-документов, где основное внимание уделяется содержимому, а не параметрам тегов.