**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:**

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, онлайнвалидатор, 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 XMLдокумента 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.

**Результаты выполненной работы:**

**1. Создание XML-документа**

В рамках выполнения лабораторной работы был создан XML-документ, представляющий библиотеку. В данном документе приведены сведения о трех книгах, включая их названия, авторов, годы издания, жанры и стоимость. Файл был сохранен под именем library.xml, проверен на корректность синтаксиса, включая закрытие тегов, правильное форматирование и наличие всех обязательных элементов.

**2. Создание XSD-схемы**

Для валидации XML-документа library.xml была создана XSD-схема. Данная схема обеспечивает проверку следующих условий:

* Наличие всех обязательных элементов, включая title, author, year, genre и price.
* Уникальность и обязательное наличие атрибута id для каждого элемента <book>.
* Соответствие поля year целому числу, лежащему в диапазоне от 1800 до текущего года.
* Поле price проверяется на соответствие положительному числу с двумя знаками после запятой.

Схема была сохранена в файл library.xsd.

**3. Валидация XML-документа**

XML-документ был проверен на соответствие XSD-схеме с использованием Java и библиотеки javax.xml.validation. Проверка подтвердила корректность документа. При обнаружении ошибок были внесены исправления для обеспечения полного соответствия схеме.

**4. Обработка XML-документа**

Была написана программа на языке Java для обработки документа library.xml. Программа выполняет:

* Чтение XML-документа и вывод всех данных о книгах в консоль.
* Вычисление средней цены книг.
* Фильтрацию книг по заданным критериям (например, жанру или году).

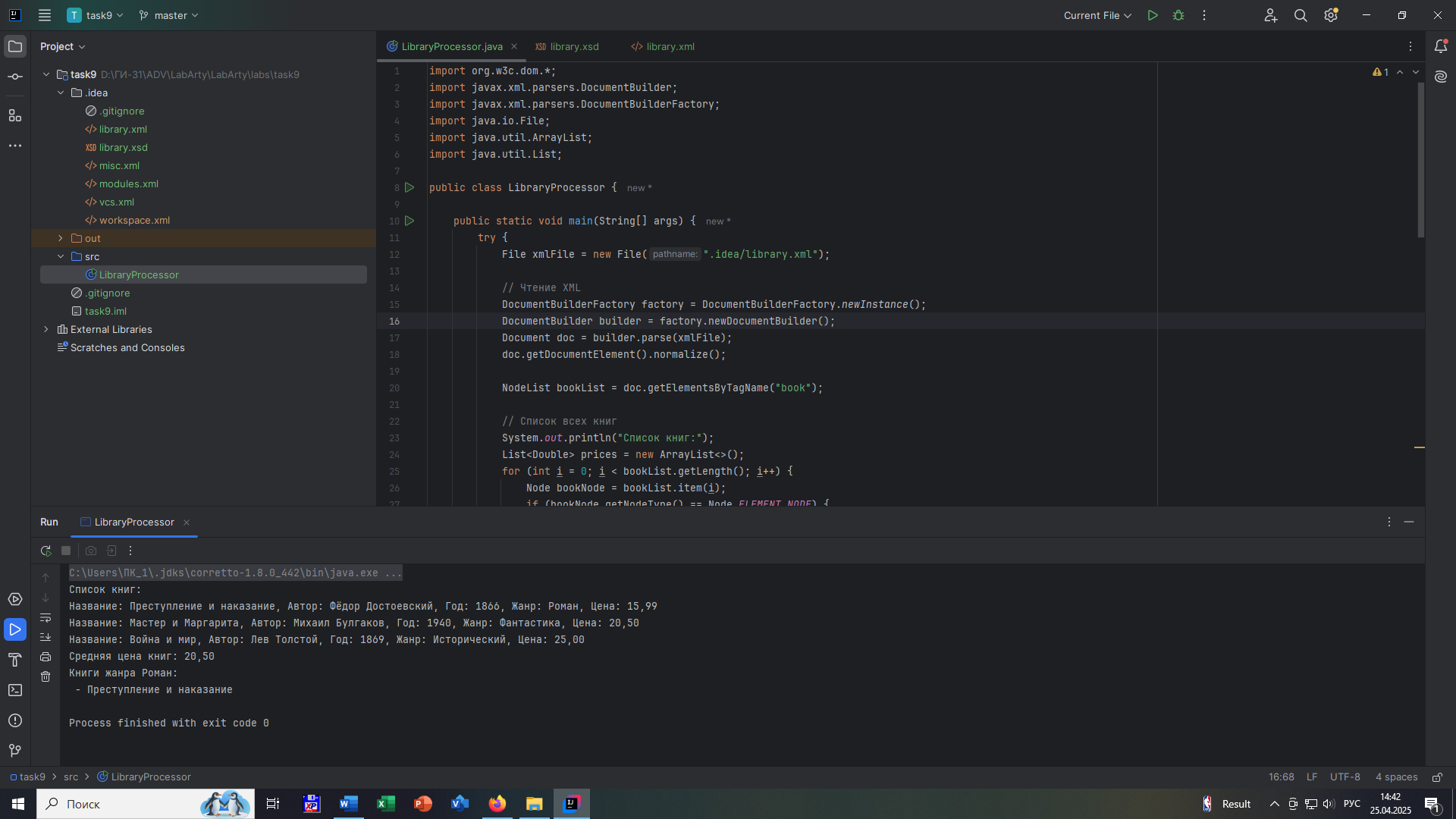


Рисунок 9.1 – Результат работы программы

**5. Исследование и сравнение**

Пример файла для сравнения:

<rss version="2.0">  
 <channel>  
 <title>Пример RSS</title>  
 <link>http://example.com</link>  
 <item>  
 <title>Первая статья</title>  
 <description>Описание первой статьи.</description>  
 </item>  
 <item>  
 <title>Вторая статья</title>  
 <description>Описание второй статьи.</description>  
 </item>  
 </channel>  
</rss>

Этот документ представляет собой пример RSS-канала, структурированного в формате XML. Корневым тегом является `<rss>`, содержащий атрибут `version="2.0"`, который указывает версию формата RSS. Внутри него находится тег `<channel>`, который представляет канал статей. В `<channel>` содержатся теги `<title>` и `<link>`, определяющие название и ссылку канала, а также несколько `<item>` — элементы, представляющие статьи. Каждый `<item>` включает вложенные теги `<title>` и `<description>`, которые содержат заголовок и описание соответствующей статьи. Документ имеет четкую вложенность: `<item>` находится внутри `<channel>`, а он — внутри `<rss>`.

Этот документ используется для распространения статей или обновлений в машиночитаемом формате, который может быть обработан RSS-агрегаторами или другими программами для получения и отображения информации. RSS-каналы упрощают пользователям доступ к статьям, позволяя им получать обновления из разных источников в одном месте.