



Софийски университет „Св. Кл. Охридски“

Факултет по математика и информатика

*Бакалавърска програма
„Софтуерно инженерство“*



Предмет: XML технологии за семантичен Уеб

Зимен семестър, 2020/2021 год.

Тема №32: B2B XML transformation by SAX

Курсов проект

Автори:

Станислав Димов, фак. номер 62352

Георги Кацарски, фак. номер 62353

януари, 2021

София

Съдържание

1	Error! Bookmark not defined.
2	Error! Bookmark not defined.
2.1	3
2.2	Error! Bookmark not defined.
2.3	4
3	Error! Bookmark not defined.
4	5
5	6
6	7
7	Error! Bookmark not defined.



1 Въведение

Настоящият документ представлява документация на курсовия проект - B2B XML transformation by SAX.

XML не дефинира строго таговете, които ще бъдат използвани в даден документ и това позволява на потребителите да създават тагове, които са нужни в съответната приложна област. Това поражда и проблеми свързани с разликите, които са породени от различните начини за представяне на информация от различни потребители. Един обект може да бъде представен чрез различни имена на тагове и чрез различна йерархичност на таговете, но неговата семантика да не бъде променена.

Реализацията на курсовия проект решава конкретен проблем - една книжарница, предлагаща само хартиени издания на книги, и една онлайн книжарница, предлагаща само книги в електронен вариант, решават да направят общ проект за книжарница, която съдържа както хартиени книги, така и книги в електронен вариант. Проблемът е, че двете книжарници представят сходна информация чрез XML по различен начин. Нашата задача е като използваме предварително зададените DTD валидационни документи да направим трансформация от каталог с хартиени издания на книги в каталог за дигитални книги.

За решаването на проблема използваме интерфейса на SAX в JAVA, който ни предоставя възможност за императивна обработка на XML документ. По този начин съхраняваме информацията от даден XML каталог в контейнер от реализиран от нас клас представляващ съответно дигитална книга или хартиена книга. В тези класове са създадени методи, създаващи съответно XML представяне за данни, които съдържа.

В следващите страници може да откриете по-задълбочено представяне на проблема и решението.

2 Анализ на решението

2.1 Работен процес

Нужна ни е средна за програмиране, в която да се заредят класовете и XML файл, който да бъде обработен. Чрез main методите на двата основни класа SAXParserDigitalToPhysical и SAXParserPhysicalToDigital стартираме програмата. Нужно е да укажем къде се намират съответните XML документи, а когато програмата обработи файла, ще бъдат създадени нови XML файлове, съдържащи трансформираното съдържание

2.2 Структура на съдържанието

Структурата на JAVA програмата е представена чрез използването на packages и обособяването на отделните смислови единици в отделни такива.

2.3 Тип и представяне на съдържанието

SAXParserDigitalToPhysical.java - Имплементация на SAX Default Handler

SAXParserPhycialToDigital .java - Имплементация на SAX Default Handler

Author.java - клас представящ автор

Publisher.java - клас представляващ издател

DigitalBook.java - клас представляващ дигитална книга

DigitalBookTags.java - клас, съдържащ всички тагове на дигитална книга

PhysicalBook.java - клас представляващ хартиена книга

PhysicalBookTags.java - клас, съдържащ всички тагове на хартиена книга

digital_book.xml - съдържа дигиталните книги, представени с помощта на XML

physical_book.xml - съдържа хартиените книги, представени с помощта на XML

digital_book_validation.dtd - представлява валидиращ документ за digital_book.xml

physical_book_validation.dtd - представлява валидиращ документ за physical_book.xml

3 Дизайн

Чрез override-натите методи на основните методи на SAX интерфейса:

startDocument(),startElement(),characters(),endElement(),endDocument() ние обработваме събитията, които се случват по време на четенето на XML файла. Съответно информацията която се прочита се съхранява в класовете DigitalBook или PhysicalBook, които от своя страна имат методи, парсващи тази информация към противоположния тип книга във формата на String.

Методът startDocument() създава файл, в който ще пишем и създава основните средства за работа - книгата,която се обработва в момента и контейнер, съдържащ всички обработени книги.

Методът startElement() записва типа на елемента, който обработваме и ако елемента има атрибути, които са ни нужни те се записват в класовете за съхраняване.

Методът characters() служи за прочитане на съдържанието на елемента и съхранението му. Ключови са таблицата, която указва дали обработваме даден елемент и методът правещ проверката.

Методът endElement() служи да укажем, че вече не обработваме даден елемент.

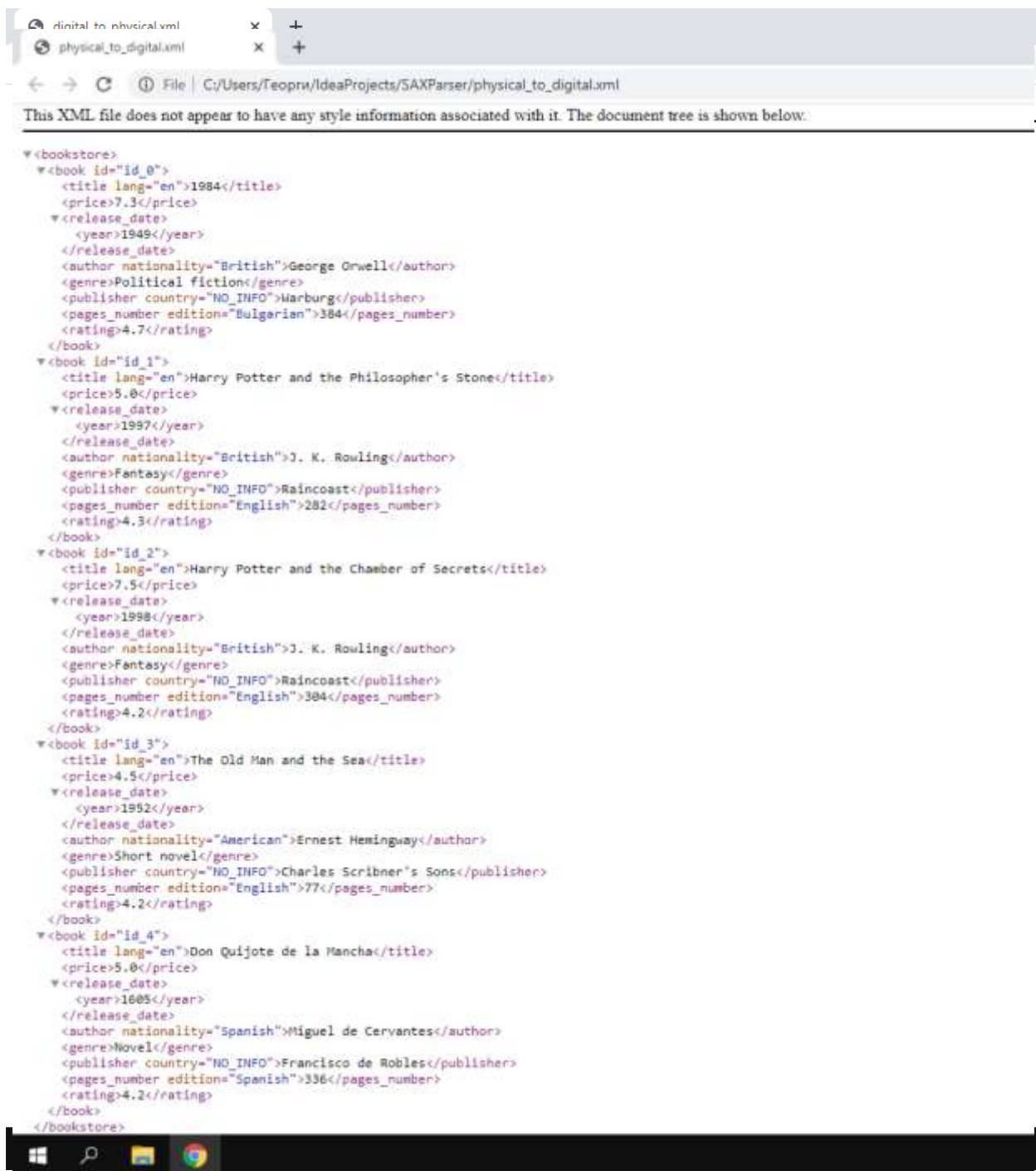
Методът endDocument() служи да запишем обработената информация в нов XML файл.

4 Тестване

Тук обяснете как сте тествали предложеното решение:

- XML документите бяха тествани спрямо съответния им DTD документ чрез сайта xmlvalidation.com

- XML документите бяха проверени в Google Chrome и Opera



5 Заключение и възможно бъдещо развитие

Програмата ще работи само при така въведените валидиращи dtd документи. Възможно е програмата да работи и при корекции, но поведението ѝ не е дефинирано. Бъдещото развитие би било в насока поддръжка на текущия инструмент за работа и по-добрата обработка на грешки.

В заключение, обобщете резултатите от работата ви по проекта, както и предимствата и ограничеността на използваните технологии/езици/методи... Укажете какви алтернативи могат да се използват и техните предимства и недостатъци. Опишете каква е използваемостта на подобни решения в практиката и какво бихте предложили като насоки за бъдещо развитие на вашето решение.

6 Разпределение на работата

Разпределихме работата по xml документите и dtd файловете на две, като всеки направи двойка за съответната книжарница. След обсъждане и корекции и отстраняване на грешки и изготвяне на крайния им вариант, започнахме взаимна работа по SAX парсерът и измисляне на начин за използването на интерфейса му. В точка 4 и в java файловете са описани тези начини. По долу сме описали основните ни източници на информация, служещи за създаването на парсера.

7 Използвани литературни източници и Уеб сайтове

1. https://docs.oracle.com/cd/B28359_01/appdev.111/b28394/adx_j_parser.htm#ADXDK19142
2. <https://www.xmlvalidation.com/>
3. <https://docs.oracle.com/javase/tutorial/jaxp/sax/parsing.html>
4. <https://www.baeldung.com/java-sax-parser>
5. <https://stackoverflow.com>