Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра экономической информатики

ОТЧЕТ

ЛАБОРАТОРНАЯ РАБОТА №4

«XML, работа с SAX и DOM парсерами»

Выполнил:

Студент ФКП, гр.814301

Мацаль Н.Ю.

Проверил:

Лыщик А.П.

Минск 2020

**Цель:** Является освоение навыков работы с данными в формате XML средствами языка Java.

**Задание:**

Разработать программу и вывести на экран информацию о списке фильмов.

Требуется:

* определить схему XML-файла;
* определить структуры данных для представления в памяти содержимого XML-файла
* после выполнения разбора (если такое представление необходимо по заданию);
* выбрать способ разбора (с создание документа или без такового);
* реализовать программу на языке программирования Java.

**Ход работы**

В данной лабораторной работе XML-файл хранит информацию о компаниях: название, основатель, дата основания, страна и капитал.

XML схема

*<?*xml version="1.0" encoding="UTF-8"*?>*<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="CompanyList" type="CompanyListType"/>  
 <xs:complexType name="CompanyType">  
 <xs:sequence>  
 <xs:element name="Title">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:enumeration value="Itra"/>  
 <xs:enumeration value="Apple"/>  
 <xs:enumeration value="Лаванда"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="Director">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:enumeration value="Антон Вареник"/>  
 <xs:enumeration value="Стив Джобс"/>  
 <xs:enumeration value="Михаил Васильевич"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="Date" minOccurs="0">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:enumeration value="2000"/>  
 <xs:enumeration value="1967"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="Country">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:enumeration value="РБ"/>  
 <xs:enumeration value="США"/>  
 <xs:enumeration value="Румыния"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 <xs:element name="Money">  
 <xs:simpleType>  
 <xs:restriction base="xs:string">  
 <xs:enumeration value="1200000"/>  
 <xs:enumeration value="1000000000"/>  
 <xs:enumeration value="120"/>  
 </xs:restriction>  
 </xs:simpleType>  
 </xs:element>  
 </xs:sequence>  
 <xs:attribute type="xs:string" name="number" use="optional"/>  
 </xs:complexType>  
 <xs:complexType name="CompanyListType">  
 <xs:sequence>  
 <xs:element type="CompanyType" name="Film" maxOccurs="unbounded" minOccurs="0"/>  
 </xs:sequence>  
 </xs:complexType>

XSD — это язык описания структуры XML документа. Его также называют XML Schema. При использовании XML Schema XML парсер может проверить не только правильность синтаксиса XML документа, но также его структуру, модель содержания и типы данных.

Элемент complexType представляет собой сложный тип, который хранит в себе более простые элементы и атрибуты. Элемент sequence задает последовательность. Элемент restriction задает только какой-то определенный тип данных или промежуток значений. Например, в коде выше задается только значение типа string для title.

Код программы

Мы работаем с DOM парсерами, инициализируем новые

переменные для вызова из XML файла тегов с значениями.

import java.io.IOException;  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
import javax.xml.parsers.ParserConfigurationException;  
import org.w3c.dom.Document;  
import org.w3c.dom.Node;  
import org.w3c.dom.NodeList;  
import org.xml.sax.SAXException;  
  
public class DOM {  
 public static void main(String[] args) {  
 try {  
 *// Создается построитель документа* DocumentBuilder documentBuilder = DocumentBuilderFactory.*newInstance*().newDocumentBuilder();  
  
 *// Создается дерево DOM документа из файла* Document document = documentBuilder.parse("src/lab2/CompanyList.xml");  
  
 *// Получаем корневой элемент* Node root = document.getDocumentElement();  
  
 System.*out*.println("List of companes:");  
 System.*out*.println();  
 *// Просматриваем все подэлементы корневого* NodeList films = root.getChildNodes();  
 for (int i = 0; i < films.getLength(); i++) {  
 Node film = films.item(i);  
 *// Смотрим, если нода не текст, то это компания* if (film.getNodeType() != Node.*TEXT\_NODE*) {  
 NodeList filmsProps = film.getChildNodes();  
 for(int j = 0; j < filmsProps.getLength(); j++) {  
 Node filmsProp = filmsProps.item(j);  
 *// Если нода не текст, то это одна из параметров компании, выводим это* if (filmsProp.getNodeType() != Node.*TEXT\_NODE*) {  
 System.*out*.println(filmsProp.getNodeName() + ":" + filmsProp.getChildNodes().item(0).getTextContent());  
 }  
 }  
 System.*out*.println("====================");  
 }  
 }  
  
 } catch (ParserConfigurationException ex) {  
 ex.printStackTrace(System.*out*);  
 } catch (SAXException ex) {  
 ex.printStackTrace(System.*out*);  
 } catch (IOException ex) {  
 ex.printStackTrace(System.*out*);  
 }  
 }  
}

XML-файл:

*<?*xml version="1.0" encoding="UTF-8"*?>*<CompanyList>  
 <Company number= "1">  
 <Title>Itra</Title>  
 <Director>Антон Вареник</Director>  
 <FoundationDate>2000</FoundationDate>  
 <Country>РБ</Country>  
 <Money>1200000</Money>  
 </Company>  
 <Company number= "2">  
 <Title>Apple</Title>  
 <Director>Стив Джобс</Director>  
 <FoundationDate>1967</FoundationDate>  
 <Country>США</Country>  
 <Money>1000000000</Money>  
 </Company>  
 <Company number= "3">  
 <Title>Лаванда</Title>  
 <Director>Михаил Васильевич</Director>  
 <Country>Румыния</Country>  
 <Money>120</Money>  
 </Company>  
</CompanyList>

Результаты выполнения:

