Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

Отчёт

по лабораторной работе №3

на тему:

XML, РАБОТА С SAX И DOM ПАРСЕРАМИ

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Станкевич А.Д.

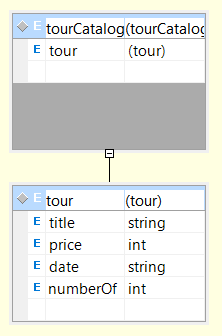
(подпись)

Выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Прихожая А.В., группа 114302

(подпись)

Минск 2023

Схема валидации XML-файла представлена ниже.



Файл xml.

<?xml version="1.0" encoding="windows-1251"?>  
<tourCatalog>  
 <tour>  
 <title>Обзорная экскурсия по Минску</title>  
 <price>50</price>  
 <date>25.09.2023</date>  
 <numberOfPeople>8</numberOfPeople>  
 </tour>  
 <tour>  
 <title>Экскурсия Замки Мир - Несвиж</title>  
 <price>135</price>  
 <date>26.09.2023</date>  
 <numberOfPeople>12</numberOfPeople>  
 </tour>  
 <tour>  
 <title>Экскурсия Хатынь - Курган Славы</title>  
 <price>60</price>  
 <date>26.09.2023</date>  
 <numberOfPeople>2</numberOfPeople>  
 </tour>  
 <tour>  
 <title>Экскурсия Брест - Брестская Крепость - Беловежская пуща</title>  
 <price>160</price>  
 <date>27.09.2023</date>  
 <numberOfPeople>26</numberOfPeople>  
 </tour>  
 <tour>  
 <title>Экскурсия на завод БелАЗ</title>  
 <price>75</price>  
 <date>29.09.2023</date>  
 <numberOfPeople>4</numberOfPeople>  
 </tour>  
</tourCatalog>

Файл xml был проверен на валидность, файл валидный.

Класс MyClass.

import org.w3c.dom.Document;  
import org.w3c.dom.Element;  
import org.w3c.dom.NodeList;  
import org.xml.sax.SAXException;  
  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
import javax.xml.parsers.ParserConfigurationException;  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.List;  
  
public class MyClass {  
 public static void main(String[] args) {  
 List<Tour> tourCatalog = new ArrayList<>();  
  
 try {  
 DocumentBuilder builder = DocumentBuilderFactory.*newInstance*().newDocumentBuilder();  
 Document document = builder.parse("scheme.xml");  
  
 Element tourcat = document.getDocumentElement();  
  
 if ("tourCatalog".equals(tourcat.getTagName())) {  
  
 NodeList tours = tourcat.getElementsByTagName("tour");  
 for (int i = 0; i < tours.getLength(); i++) {  
  
 Element tour = (Element) tours.item(i);  
 String title = null;  
 int price = 0;  
 String date = null;  
 int numberOfPeople = 0;  
  
 NodeList properties = tour.getElementsByTagName("\*");  
 for (int j = 0; j < properties.getLength(); j++) {  
  
 Element property = (Element) properties.item(j);  
 if ("title".equals(property.getTagName()))  
 title = property.getTextContent();  
 else if ("price".equals(property.getTagName()))  
 price = Integer.*parseInt*(property.getTextContent());  
 else if ("date".equals(property.getTagName()))  
 date = property.getTextContent();  
 else if ("numberOfPeople".equals(property.getTagName()))  
 numberOfPeople = Integer.*parseInt*(property.getTextContent());  
 }  
  
 Tour tourItem = new Tour(title, price, date, numberOfPeople);  
 tourCatalog.add(tourItem);  
 }  
 }  
  
 for (Tour tour:tourCatalog) {  
 System.*out*.println("Название: " + tour.getTitle() + "\nЦена: " + tour.getPrice() + " руб." +  
 "\nДата: " + tour.getDate() + "\nКоличество людей: " + tour.getNumberOfPeople());  
 System.*out*.println("===========>>>>");  
 }  
  
 } catch (ParserConfigurationException e) {  
 System.*out*.println("Ошибка при создании построителя");  
 } catch (IOException e) {  
 System.*out*.println("Файл формата xml не найден в папке проекта");  
 } catch (SAXException e) {  
 System.*out*.println("Ошибка при парсинге");  
 }  
 }  
}

Класс Tour.

public class Tour {  
 private String title;  
 private int price;  
 private String date;  
 private int numberOfPeople;  
  
 public Tour(String title, int price, String date, int numberOfPeople) {  
 this.title = title;  
 this.price = price;  
 this.date = date;  
 this.numberOfPeople = numberOfPeople;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public int getPrice() {  
 return price;  
 }  
  
 public String getDate() {  
 return date;  
 }  
  
 public int getNumberOfPeople() {  
 return numberOfPeople;  
 }  
}

Для парсинга была был выбран Dom Parser. Он анализирует XML-документ, загружая все содержимое документа и создавая его полное иерархическое дерево в памяти.

Объектная модель документа (DOM) является официальной рекомендацией Консорциума World Wide Web (W3C). Он определяет интерфейс, который позволяет программам получать доступ и обновлять стиль, структуру и содержимое документов XML. XML-парсеры, поддерживающие DOM, реализуют этот интерфейс.

DOM – это общий интерфейс для управления структурами документов. Одна из целей разработки заключается в том, чтобы код Java, написанный для одного DOM-совместимого синтаксического анализатора, работал на любом другом DOM-совместимом синтаксическом анализаторе без каких-либо изменений.