

# ЛАБОРАТОРНА РОБОТА №2

з дисципліни «Об'єктно-орієнтоване програмування»

**Тема:** Обробка XML-документів (DOM, SAX, StAX) **Варіант:** Інтернет-сторінки

# 1 Завдання

Розробити структуру XML для зберігання даних про інтернет-сторінки, схему XSD для валідації та Java-додаток для парсингу цих даних трьома методами (DOM, SAX, StAX). Реалізувати перетворення XSLT. **Сутності:** Site (корінь), Page (Title, Type, Chars, Authorize). **Характеристики (Chars):** Email, News, Archive, Voting, Paid.

## 2 Реалізація XML та XSD

Створено файл pages.xml з кореневим елементом Site та файл pages.xsd для валідації типів даних (перерахування для Type та Voting).

### 2.1 XSD Схема

```
<xs:complexType name="PageType">
  <xs:sequence>
    <xs:element name="Title" type="xs:string"/>
    <xs:element name="Type" type="PageCategory"/>
    <xs:element name="Chars" type="CharsType"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="required"/>
  <xs:attribute name="authorize" type="xs:boolean"/>
</xs:complexType>
```

## 3 Лістинг Java коду

### 3.1 Модель (Page.java)

Клас відповідає структурі XML та реалізує інтерфейс Comparator для сортування за назвою.

```
1 public class Page {
2     private String id;
3     private String title;
4     private String type;
5     // ... інші поля ...
6
7     public static Comparator<Page> byTitle =
8         Comparator.comparing(Page::getTitle);
9 }
```

### 3.2 Парсер (DOMPageParser.java)

```
1 public List<Page> parse(String filePath) throws Exception {
2     DocumentBuilder builder = factory.newDocumentBuilder();
3     Document doc = builder.parse(new File(filePath));
4     NodeList nodeList = doc.getElementsByTagName("Page");
5     // Ітерація створення об'єктів ...
6 }
```

## 4 Результати роботи

Програма успішно пройшла валідацію XML-файлу, розпарсила дані всіма трьома методами та згенерувала новий XML-файл (pages\_new.xml) за допомогою XSLT трансформації, згрупувавши сторінки за типом (Portal, News, Mirror).