

## Ficha de Trabalho nº 8

### XSLT: Transformação de documentos XML

#### 1. Introdução

Nesta ficha de trabalho pretende-se que os alunos explorem o processamento de ficheiros XSLT para geração de outputs. No Moodle encontram-se os ficheiros necessários à resolução desta ficha de trabalho.

#### 2. Transformações com XSLT

##### Funções fornecidas

**Document transformaDocumento(Document XMLdoc, String xmlFile, String xslFile)**

Recebe:

- Document contendo o ficheiro XML a transformar
- String contendo o nome do ficheiro XML a transformar
- String contendo o nome do ficheiro XSL com os templates de transformação

Devolve:

- Document contendo o ficheiro XML/HTML resultante da transformação

>> Este ficheiro deve depois ser gravado em disco com a função *escreveDocumentoParaFicheiro* da classe *XMLJDomFunctions.java*

**void transformaDocumento2(String xmlFile, String xslFile, String sOutFile)**

Recebe:

- String contendo o nome do ficheiro XML a transformar
- String contendo o nome do ficheiro XSL com os templates de transformação
- String: contendo o nome do ficheiro TXT resultante da transformação

## Como fazer uma transformação XSLT em Java?

1. Usando um editor XML crie o ficheiro .xsl contendo os templates de transformação. Grave o ficheiro na raiz do seu projeto Java
2. No código java:
  - i. Criar uma instância DOM do ficheiro XML a transformar (usar a função `lerDocumentoXML`)
  - ii. Transformações para XML ou HTML  
chamar a função `transformaDocumento`  
gravar o resultado da transformação em disco (usar a função `escreverDocumentoParaFicheiro`)
  - iii. Transformações para TXT  
chamar a função `transformaDocumento2`

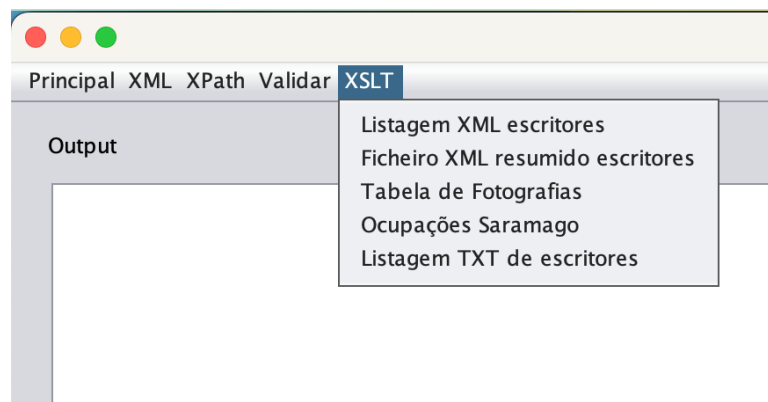
### Exemplo:

Depois de escrito o ficheiro **trf.xsl** com as regras de transformação, este código mostrado abaixo transforma um ficheiro **alunos.xml** noutra ficheiro XML de nome **lista\_alunos.xml**

```
Document doc = XMLJDomFunctions.lerDocumentoXML("alunos.xml");
if(doc!=null){
    Document novo = JDOMFunctions_XSLT.transformaDocumento(doc, "alunos.xml", "trf.xsl");
    XMLJDomFunctions.escreverDocumentoParaFicheiro(novo,"lista_alunos.xml");
}
```

## 3. Tarefas a executar

- a) Abra o Projeto da aula anterior
- b) Copie o ficheiro **JDOMFunctions\_XSLT.java** para a pasta **src** do projeto. Corrija o nome do package se necessário.
- c) Na Frame acrescente mais uma opção no Menu, criando três menu item, como se ilustra na figura abaixo.



- d) Implemente uma transformação de nome **transf1.xsl** que escreva os Templates XSLT que permitam fazer uma transformação do ficheiro **escritores.xml** para outro ficheiro XML (**listagem.xml**) contendo os nomes dos escritores ordenados por ordem alfabética:

```
<?xml version="1.0" encoding="utf-8"?>
<listagem>
  <escritor nome="António Lobo Antunes">António Lobo Antunes</escritor>
  <escritor nome="Eça de Queirós">q</escritor>
  <escritor nome="Fernando Pessoa">Fernando António Nogueira Pessoa</escritor>
  <escritor nome="J. K. Rowling">Joanne Rowling</escritor>
  <escritor nome="José Saramago">José de Sousa Saramago</escritor>
  <escritor nome="Mário Cesariny">Mário Cesariny de Vasconcelos</escritor>
  <escritor nome="Paulo Coelho">Paulo Coelho de Souza</escritor>
</listagem>
```

Associe ao *menuitem* **Listagem XML escritores** o código que permita executar a transformação anterior e mostrar o resultado na *textArea*.

```
Document doc = XMLJDomFunctions.lerDocumentoXML("escritores.xml");
if (doc != null) {
    Document novo = JDOMFunctions_XSLT.transformaDocumento(doc, "escritores.xml", "transf1.xsl");
    XMLJDomFunctions.escreverDocumentoParaFicheiro(novo, "listagem.xml");

    doc = XMLJDomFunctions.lerDocumentoXML("listagem.xml");
    String t = XMLJDomFunctions.escreverDocumentoString(doc);
    JTextArea1.setText(t);
}
```

Este código mostra o  
ficheiro XML na *textArea*

- e) Implemente uma transformação de nome **transf2.xsl** que escreva os Templates XSLT que permitam fazer uma transformação do ficheiro **escritores.xml** para outro ficheiro XML (**escritores2.xml**) contendo a seguinte estrutura:

```
<escritores>
  <escritor>
    <nome>António Lobo Antunes</nome>
    <nome_completo>António Lobo Antunes</nome_completo>
    <num_ocupacoes>2</num_ocupacoes>
  </escritor>
  <escritor>
    <nome>Eça de Queirós</nome>
    <nome_completo>q</nome_completo>
    <num_ocupacoes>3</num_ocupacoes>
  </escritor>
  ...
</escritores>
```

Como fez no exercício anterior, associe ao *menuitem* **Ficheiro XML resumido** o código que permita executar a transformação anterior e mostrar o resultado na *textArea*.

- f) Implemente uma transformação de nome **transf3.xsl** que escreva os Templates XSLT que permitam fazer uma transformação do ficheiro **escritores.xml** para um ficheiro HTML (**fotos.html**), contendo uma tabela o nome do escritor e a sua fotografia. A tabela deve ficar ordenada ordem alfabética do nome:

### Lista de escritores

NOME	FOTO
António Lobo Antunes	
Eça de Queirós	
Fernando Pessoa	

Associe ao *menu item* **Tabela de fotografias** o código que permita executar a transformação anterior e mostrar o resultado no Browser.

```
Document doc = XMLJDomFunctions.lerDocumentoXML("escritores.xml");
if (doc != null) {
    Document novo = JDOMFunctions_XSLT.transformaDocumento(doc, "escritores.xml", "transf3.xsl");
    XMLJDomFunctions.escreverDocumentoParaFicheiro(novo, "imagens.html");
    doc = XMLJDomFunctions.lerDocumentoXML("fotos.html");
    String t = XMLJDomFunctions.escreverDocumentoString(doc);
    JTextArea1.setText(t);
    JOptionPane.showMessageDialog(this,
        "Transformação feita com sucesso... a abrir browser...",
        "XSLT para HTML", JOptionPane.INFORMATION_MESSAGE);
    String url = "fotos.html";
    File htmlFile = new File(url);
    Desktop.getDesktop().browse(htmlFile.toURI());
}
```

Este código mostra o HTML na textArea

Este código abre o ficheiro HTML no Browser

- g) Implemente uma transformação de nome **transf4.xml** que escreva os Templates XSLT que permitam fazer uma transformação do ficheiro **escritores.xml** para um ficheiro HTML (**listagem.html**), contendo uma listagem de *bullets* com as ocupações do escritor José Saramago, ordenados ordem alfabética:

### Ocupações de José Saramago

- argumentista
- contista
- cronista
- dramaturgo
- ensaísta
- escritor
- funcionário público
- jornalista
- mecânico
- poeta
- romancista
- serralheiro
- teatrólogo
- tradutor
- técnico editorial

Tal como fez no exercício anterior, associe ao *menu item* **Ocupações Saramago** o código que permita executar a transformação anterior e mostrar o resultado no Browser.

- h) Implemente uma transformação de nome **transf5.xml** que escreva os Templates XSLT que permitam fazer uma transformação do ficheiro **escritores.xml** para um ficheiro TXT (**escritores.txt**) contendo uma listagem dos nomes de todos os **escritores do ficheiro** ordenada por ordem alfabética.

Lista de escritores:

António Lobo Antunes  
Eça de Queirós  
Fernando Pessoa  
J. K. Rowling  
José Saramago  
Mário Cesariny  
Paulo Coelho

Associe ao *menuitem* **Listagem TXT de escritores** o código XSLT que permita executar a transformação anterior e mostrar o resultado na *textArea*. O ficheiro TXT resultado da transformação deve ser mostrado na **TextArea** usando a classe **Scanner** para ler o ficheiro linha a linha e concatenar todas as linhas numa variável do tipo **StringBuilder** que deve depois ser mostrada na *textArea*.

```
Document doc = XMLJDomFunctions.lerDocumentoXML("escritores.xml");
if (doc != null) {
    JDOMFunctions_XSLT.transformaDocumento2("escritores.xml", "transf5.xml", "escritores.txt");
    Scanner ler = new Scanner (new FileInputStream("escritores.txt"));
    StringBuilder texto = new StringBuilder();
    String linha;
    while(ler.hasNextLine()){
        linha = ler.nextLine();
        texto = texto.append(linha).append("\n");
    }
    ler.close();
    JTextArea1.setText(texto.toString());
}
```