

## Bài 3. XSLT

### 1. Xpath (XML Path Language)

#### 1.1. Giới thiệu

- Ngôn ngữ Xpath giúp ta đi lại trên các phần tử của tài liệu XML để trích ra những dữ liệu mà chúng ta cần thiết.
- Xpath xem tài liệu XML như là 1 cây.

**Ví dụ:** ta có tài liệu XML như sau:

```
<?xml version="1.0"?>
```

```
<HoaDon MaHD="1044">
```

```
  <NgayBan>2009-03-26</NgayBan>
```

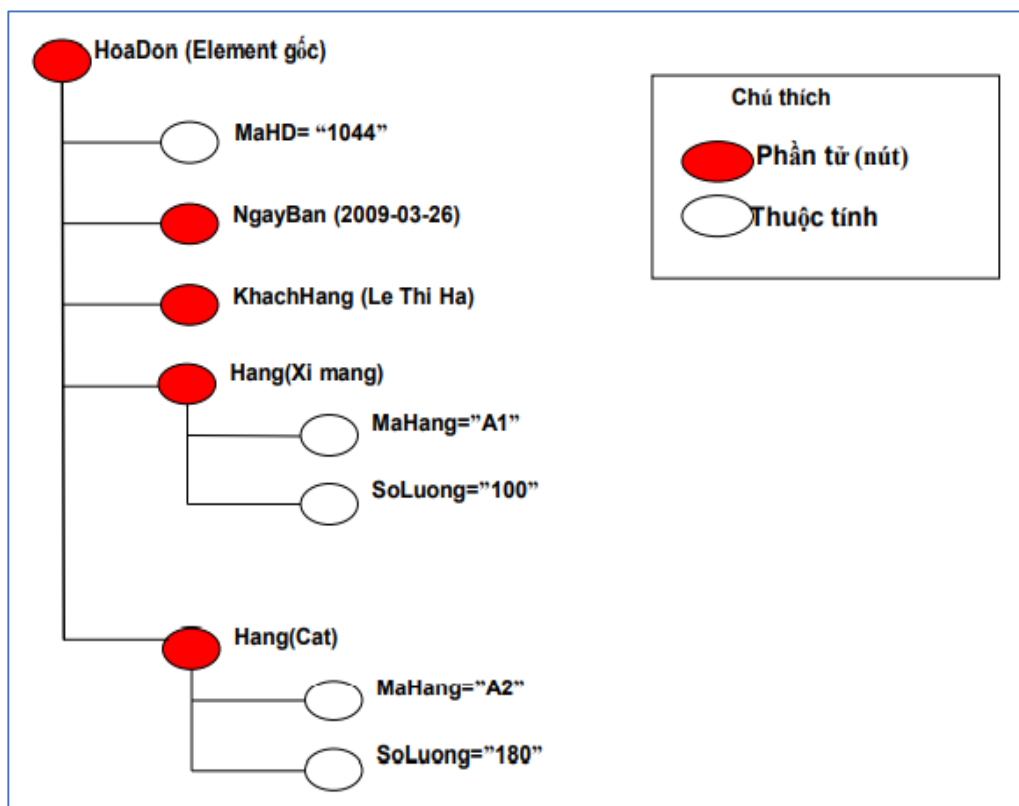
```
  <KhachHang>Le Thi Ha</KhachHang>
```

```
  <Hang MaHang="A1" SoLuong="100">Xi Mang</Hang>
```

```
  <Hang MaHang="A2" SoLuong="180">Cat</Hang>
```

```
</HoaDon>
```

- Tài liệu XML trên sẽ được biểu diễn dưới dạng cây như sau:



## 1.2. Cú pháp của Xpath

### 1.2.1. Đường dẫn tuyệt đối

+ Dùng dấu **/** để chỉ ra một đường dẫn tuyệt đối bắt đầu từ phần tử gốc.

**Ví dụ:** Trong hình trên, bây giờ chúng ta muốn chọn các nút **Hang** ta viết như sau:

***/HoaDon/Hang***

+ Trong trường hợp muốn đi đến thuộc tính của nút thì ta dùng **@** trước tên thuộc tính.

**Ví dụ:** Chọn các nút thuộc tính **MaHang**: ***/HoaDon/Hang/@MaHang***

### 1.2.2. Đường dẫn tương đối

Khi chúng ta muốn trích một phần tử nào đó mà chỉ biết tên của phần tử này chứ không biết là phần tử này nằm ở vị trí nào thì ta có thể dùng đường dẫn tương đối để làm điều này. Dùng dấu **//** để chỉ cho trình phân tích biết đây là đường dẫn tương đối.

**Ví dụ:** để chọn ra các nút phần tử có tên là **Hang** chúng ta viết như sau: ***//Hang***

Khi đó trình phân tích sẽ truy tìm đến các phần tử có tên là **Hang**

### 1.2.3. Chọn các phần tử bằng ký tự đại diện

Để chọn tất cả các phần tử con của một phần tử nào đó ta dùng ký tự đại diện **\***.

**Ví dụ:** Chọn tất cả các phần tử con của phần tử **HoaDon** ta viết như sau: ***/HoaDon/\****

### 1.2.4. Chọn các phần tử theo điều kiện

Để lấy các phần tử theo một điều kiện nào đó chúng ta viết điều kiện trong dấu ngoặc vuông([ ]).

**Ví dụ:** để lấy mọi phần tử **Hang** có thuộc tính **SoLuong >120** ta viết như sau:

***/HoaDon/Hang[@SoLuong>120]***

### 1.2.5. Một số hàm Xpath thường dùng

Tên hàm	Ý nghĩa	Ví dụ
<b>count(*)</b>	Hàm lấy tổng số nút con của một phần tử nào đó	//Hang[count(*)=2] Chọn tất cả các phần tử Hang có số phần tử con là 2
<b>name()</b>	Lấy tên của phần tử	/HoaDon/*[name()='Hang'] Chọn tất cả các phần tử con của HoaDon có tên là Hang
<b>not()</b>	Hàm phủ định	//Hang/*[not(@*)] Chọn tất cả các phần tử con của Hang không chứa thuộc tính nào
<b>normalize-space(str)</b>	Hàm bỏ khoảng trắng	//Hang/*[normalize-space(@MaHang)='abc'] Chọn tất cả các phần tử con của Hang có thuộc tính MaHang=abc (không phân biệt khoảng trắng)

<b>starts-with(str,sub str)</b>	Hàm kiểm tra xem chuỗi str có chứa chuỗi substr (tính từ vị trí đầu tiên) hay không	//Hang/*[starts-with(name(),'P')] Chọn tất cả các phần tử con của Hang có tên bắt đầu bởi ký tự P
<b>contains(str,substr)</b>	Kiểm tra một chuỗi str có chứa chuỗi con substr hay không	//Hang/*[contains(name(),'u')] Chọn tất cả các phần tử con của phần tử Hang mà tên của các phần tử con này có chứa ký tự u
<b>string-length(str)</b>	Hàm lấy chiều dài của 1 chuỗi	//Hang/*[string-length(name())=5] Chọn tất cả các phần tử con của Hang mà độ dài tên của các phần tử con này là 5

<b>position()</b>	Cho biết vị trí hiện tại của phần tử	//Hang[position()=2] Chọn phần tử Hang có vị trí là 2
<b>floor()</b>	Lấy giá trị nhỏ nhất gần với giá trị chỉ định	
<b>ceiling()</b>	Lấy giá trị lớn nhất gần với giá trị chỉ định	
<b>last()</b>	Vị trí nút cuối cùng	//Hang[last()] Chọn phần tử Hang cuối cùng

## 2. XSLT

### 2.1. Giới thiệu

**XSLT** là ngôn ngữ chuyển đổi định dạng, được kết hợp với ngôn ngữ **XPath** cho phép trích lọc và biến đổi tài liệu XML thành các định dạng khác như **HTML** hoặc văn bản thuần túy.

### 2.2. Cấu trúc file XSLT

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <!-- HTML, CSS và các lệnh (phần tử) xử lý cho phép trích rút
      thông tin từ tệp tin XML nguồn và kết xuất vào tệp tin kết
      quả. -->
  </xsl:template>
</xsl:stylesheet>
```

- Phần tử gốc trong tài liệu XSLT là phần tử **xsl:stylesheet**, nó chứa một hay nhiều phần tử **xsl:template** (khuôn mẫu cho tài liệu XML)
- Để biến đổi XML sang định dạng khác, ta cần phải có 2 tài liệu: tài liệu XML cần biến đổi và file định kiểu XSLT.
- Thuộc tính **match** trong phần tử **template** để chỉ ra **node** xuất phát.
- Để tham chiếu một file tài liệu xslt vào trong tài liệu XML bằng cách thêm vào đầu tài liệu XML dòng:

**<?xml-stylesheet type="text/xsl" href="URL"?>**

Trong đó **URL** là địa chỉ của tài liệu xslt mà chúng ta muốn tham chiếu

**Ví dụ:** Ở đầu tài liệu XML nào đó ta thêm dòng sau:

**<?xml-stylesheet type="text/xsl" href="test.xslt"?>**

Khi đó tài liệu XML sẽ được hiển thị trên trình duyệt theo file định kiểu **test.xslt**

### 2.3. Một số phần tử(element) thường dùng của XSLT

#### 2.3.1. Phần tử value-of

Phần tử **value-of** cho phép trích rút thông tin từ tệp tin XML hay từ giá trị của một biến và sau đó đưa vào tệp tin kết quả.

**Cú pháp:**

+ Nếu trích rút thông tin từ tập Xml nguồn:

**<xsl:value-of select="Biểu thức Xpath" />**

+ Nếu trích rút thông tin từ biến:

**<xsl:value-of select="\$Tên\_biến" />**

**Lưu ý:** Biểu thức Xpath bên trong thuộc tính **Select** có thể là:

- + Một biểu thức Xpath duy nhất.
- + Một biến duy nhất.
- + Một nút (phần tử) duy nhất.
- + Một biểu thức số học với thành phần là biểu thức Xpath hay biến.

### 2.3.2. Phần tử variable

Phần tử **variable** cho phép trích rút thông tin từ tập tin XML và đưa vào một biến (đúng ra là một hằng vì nội dung biến này không thể thay đổi được).

**Cú pháp:**

**<xsl:variable name="Tên\_biến" select="Biểu thức Xpath" />**

**Ví dụ:** Cho tập tin **test.xml** lưu trữ 2 số nguyên

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="test.xslt" ?>
<GOC>
<SO>6</SO>
<SO>8</SO>
</GOC>

Tạo file **test.xslt** biến đổi tập xml trên để có nội dung sau hiển thị trên trình duyệt:

**Tổng 2 số là:6+8=14**

**Cách 1: Trích rút thông tin trực tiếp**

<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:output method="html"/>

<xsl:template match="/">

<html>

```

<body>
  <b>Tổng 2 số là:</b>
  <xsl:value-of select="/GOC/SO[1]"/> +
  <xsl:value-of select="/GOC/SO[2]"/> =
  <xsl:value-of select="/GOC/SO[1] + /GOC/SO[2]" />
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

## Cách 2 : Thông qua các Biến

```

<xsl:template match="/">
  <html>
    <body>
      <b>Tổng 2 số là:</b>
      <xsl:variable name="a" select="/GOC/SO[1]" />
      <xsl:variable name="b" select="/GOC/SO[2]"/>
      <xsl:value-of select="$a" /> + <xsl:value-of select="$b" /> =
      <xsl:value-of select="$a + $b"/>
    </body>
  </html>
</xsl:template>

```

### 2.3.3. Phần tử if

Phần tử **if** cho phép thực hiện một số thẻ xử lý khi điều kiện thoả mãn (Biểu thức logic=True).

#### Cú pháp:

```

<xsl:if test="Biểu thức logic">
  Các thẻ xử lý
</xsl:if>

```

**Lưu ý: “Biểu thức logic”** bao gồm các biểu thức tính toán (trên chuỗi XPath hay giá trị biến) cùng với các phép toán quan hệ: >, >=, <, <=, =, != và các phép toán logic: and, or, not.

#### 2.3.4. Phần tử choose

Phần tử **choose** cho phép sử dụng nhiều điều kiện khác nhau. (giống như câu lệnh switch trong C)

**Cú pháp:**

```
<xsl:choose>
  <xsl:when test="Biểu thức logic 1">
    Các thẻ xử lý khi biểu thức logic 1=true
  </xsl:when>
  <xsl:when test="Biểu thức logic 2">
    Các thẻ xử lý khi biểu thức logic 2=true
  </xsl:when>
  ...
  <xsl:otherwise>
    Các thẻ xử lý khi tất cả các biểu thức logic trên đều sai
  </xsl:otherwise>
</xsl:choose>
```

**Ví dụ:** Tạo file **test.xslt** biến đổi tệp **test.xml** trên để có nội dung hiển thị trên trình duyệt là: **Số lớn nhất giữa 2 số a và b là: ?**

**Cách 1: sử dụng if**

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html"/>
  <xsl:template match="/" >
    <html>
    <body>
```

<xsl:variable name="So_1" select="/GOC/SO[1]"/>
<xsl:variable name="So_2" select="/GOC/SO[2]"/>
Số lớn nhất giữa 2 số
<xsl:value-of select="\$So_1"/> và <xsl:value-of select="\$So_2"/> là:
<xsl:if test="\$So_1 > \$So_2" >
<xsl:value-of select="\$So_1"/>
</xsl:if>
<xsl:if test="\$So_1 <= \$So_2" >
<xsl:value-of select="\$So_2"/>
</xsl:if>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

**Cách 2: dùng choose (tự làm)**

## BÀI TẬP

**Bài 1:** Cho file **Gptb1.xml** lưu trữ 2 giá trị hệ số a và b. Xây dựng file **ptb1.xslt** để giải PTB1  $AX+B=0$  và hiển thị kết quả lên trình duyệt.

<?xml version="1.0" encoding="utf-8"?>

<?xml-stylesheet type="text/xsl" href="ptb1.xslt" ?>

<goc>

<hsa>6</hsa>

<hsb>12</hsb>

</goc>

**Bài 2:** Cho file **TienDien.xml** lưu trữ chỉ số điện đầu tháng và cuối tháng. Xây dựng file **TienDien.xslt** để tính tiền điện phải trả và hiển thị kết quả lên trình duyệt.

Biết rằng: 100kw đầu giá 3000đ/kw

50kw tiếp theo giá 4000đ/kw

50kw tiếp theo giá 4500đ/kw

>200kw giá 5000đ/kw



### 2.3.5. Phần tử for-each

Phần tử **for-each** cho phép lặp lại việc thực hiện các thẻ xử lý trên tập hợp các nút là kết quả của một chuỗi truy vấn XPath.

**Cú pháp:**

**<xsl:for-each select="Biểu thức XPath">**  
**Các thẻ xử lý**  
**</xsl:for-each>**

**Ví dụ:** Tập CongTy.xml lưu trữ tên các đơn vị trong một công ty như sau:

```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet type="text/xsl" href="CongTy.xslt" ?>
<CONG_TY Ten="Công ty X">
  <DON_VI Ten="Đơn vị A" />
  <DON_VI Ten="Đơn vị B" />
  <DON_VI Ten="Đơn vị C" />
  <DON_VI Ten="Đơn vị D" />
</CONG_TY>
```

Tạo tập **CongTy.xslt** biến đổi tập **CongTy.xml** trên để có nội dung sau hiển thị trên trình duyệt:

**Công ty X**  
**DANH SÁCH ĐƠN VỊ**

STT	TÊN ĐƠN VỊ
1	Đơn vị A
2	Đơn vị B
3	Đơn vị C
4	Đơn vị D

## Code CongTy.xslt:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-prefixes="msxsl">
  <xsl:output method="html" />
  <xsl:template match="/" >
    <html>
      <body>
        <h1> <xsl:value-of select="/CONG_TY/@Ten"/> </h1>
        <h1> DANH SÁCH ĐƠN VỊ </h1>
        <table border="2" width="40%" cellpadding="0">
          <tr>
            <th>SỐ TT</th>
            <th>TÊN ĐƠN VỊ</th>
          </tr>
          <xsl:for-each select="CONG_TY/DON_VI" >
            <tr>
              <td> <xsl:value-of select="position()"/> </td>
              <td> <xsl:value-of select="@Ten"/> </td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

### 2.3.6. Phần tử apply-templates

Khi một tài liệu XSLT chứa nhiều phần tử **template**, chúng ta có thể áp dụng chúng vào một khung trình bày nào đó bằng cách dùng phần tử **apply-templates**

Cú pháp:

**<xsl:template match="Biểu thức XPath">**  
**Các thẻ xử lý**  
**</xsl:template>**

Gọi thực hiện: **<xsl:apply-templates select="Biểu thức XPath"/>**

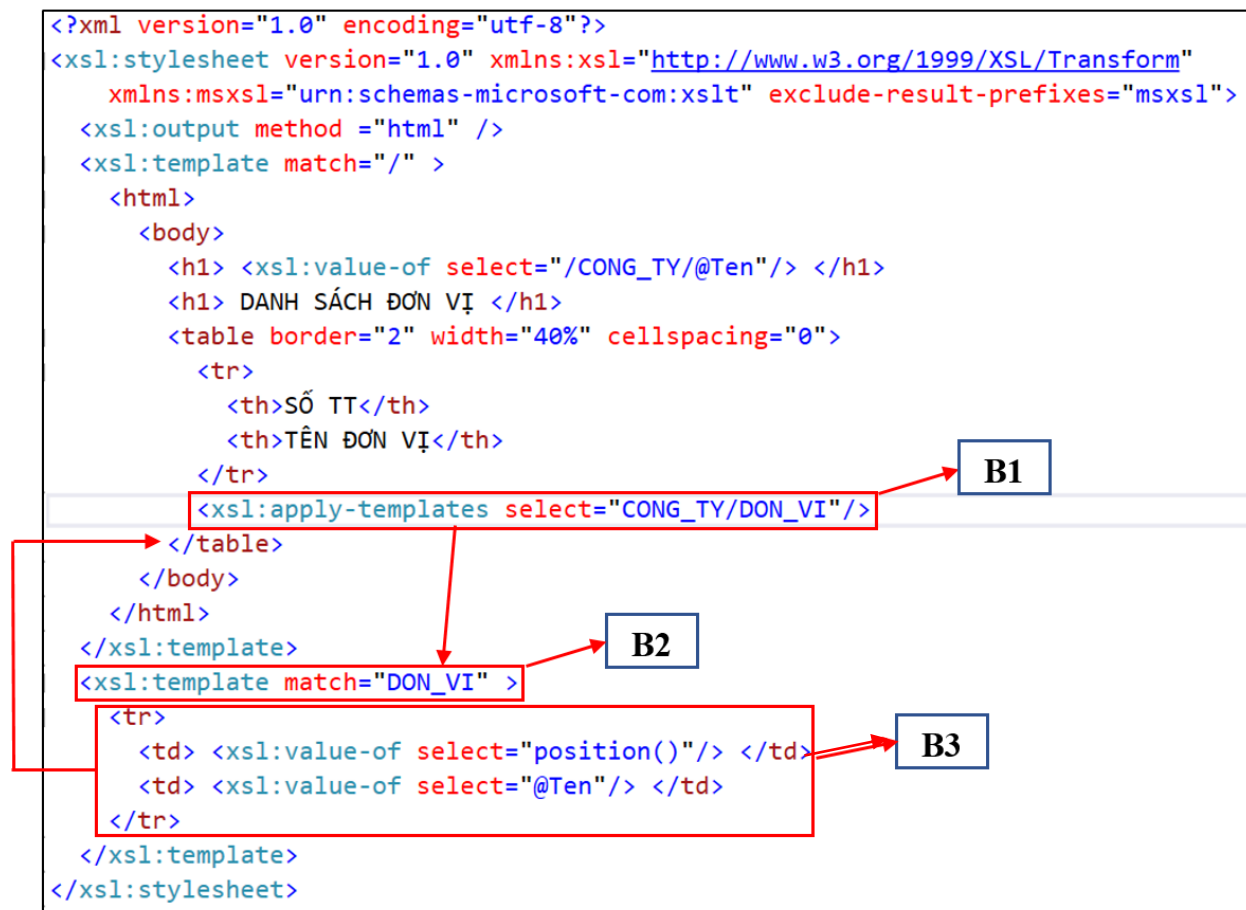
Quá trình gọi thực hiện của phần tử **apply-templates** như sau:

**Bước 1:** Xác định “Biểu thức XPath” của phần tử **apply-templates**

**Bước 2:** Tìm thuộc tính **match="Biểu thức XPath"** so khớp đúng với “Biểu thức XPath” ở bước 1.

**Bước 3:** Gọi thực hiện nhiều lần các thẻ xử lý trong **xsl:template**, mỗi lần ứng với một nút hiện thời thuộc danh sách được xác định ở bước 1.

**Ví dụ 1:** Tập **CongTy.xslt** ta có thể viết lại như sau:(dùng **Apply-templates**)



**Ví dụ 2:** Tạo tệp **CongTy.xslt** đưa ra danh sách chọn (ComboBox) các đơn vị từ tệp **CongTy.xml** trên trình duyệt:

DANH SÁCH ĐƠN VỊ:

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-prefixes="msxsl">
  <xsl:output method="html" />
  <xsl:template match="/">
    <html>
      <body>
        DANH SÁCH ĐƠN VỊ:
        <select name="dv">
          <xsl:apply-templates select="CONG_TY/DON_VI" />
        </select>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="DON_VI">
    <option>
      <xsl:value-of select="@Ten"/>
    </option>
  </xsl:template>
</xsl:stylesheet>

```

## BÀI TẬP

**Bài 1:** Cho file **nhanvien.xml** lưu trữ thông tin về các nhân viên trong từng đơn vị:

```
Tài liệu nhanvien.xml
<congtv>
  <donvi madv="a01">
    <tendv>Phòng tổ chức</tendv>
    <dienthoai>0437856864</dienthoai>
    <nhanvien>
      <manv>tc01</manv>
      <hoten>Lê Thu Hà</hoten>
      <ngaysinh>1988-12-23</ngaysinh>
      <gioitinh>nữ</gioitinh>
      <hsluong>2.34</hsluong>
    </nhanvien>
    ...
  </donvi>
  ...
</congtv>
```

**1.1** Hãy sử dụng XSLT để hiển thị thông tin về nhân viên theo mẫu sau: (**Sử dụng 2 cách: *for-each* và *apply-templates***)

BẢNG LƯƠNG THÁNG 11-2020					
Mã đơn vị: ?					
Tên đơn vị: ?					
Điện thoại: ?					
DANH SÁCH NHÂN VIÊN					
SỐ TT	MÃ NV	HỌ TÊN	NGÀY SINH	HS LƯƠNG	LƯƠNG
THỦ TRƯỞNG ĐƠN VỊ					

**1.2** Hiển thị các nhân viên có  $HSLuong \geq 3$ .

**1.3** Tạo CSS để định dạng:

- + Dòng tiêu đề của bảng có chữ đậm màu xanh lá cây, cỡ chữ 25px, nền màu đỏ.
- + Dữ liệu số trong bảng được căn phải; dữ liệu chuỗi trong bảng được căn trái.

## \* Hướng dẫn

### 1.1 và 1.2: Code nhanvien.xslt (Cách 1 sử dụng phần tử for-each)

```
<xsl:template match="congtv">
  <html>
    <body>
      <h1 align="center">BẢNG LƯƠNG THÁNG 11-2009</h1>
      <xsl:for-each select="donvi">
        <h2>Mã đơn vị: <xsl:value-of select="@madv"/></h2>
        <h2>Tên đơn vị: <xsl:value-of select="tendv"/></h2>
        <h2>Điện thoại: <xsl:value-of select="dienthoai"/></h2>
        <h2 align="center"> DANH SÁCH NHÂN VIÊN </h2>
        <table border="2" width="100%" cellpadding="0">
          <tr>
            <th>Số TT</th>
            <th>Mã NV</th>
            <th>Họ tên</th>
            <th>Ngày sinh</th>
            <th>Hệ số lương</th>
            <th>Lương</th>
          </tr>
          <xsl:for-each select="nhanvien">
            <!-- <xsl:for-each select="nhanvien[hsluong>=3]"> -->
            <!-- Hoặc <xsl:if test="hsluong>=3"> -->
            <tr>
              <td> <xsl:value-of select="position()"/></td>
              <td> <xsl:value-of select="manv"/> </td>
              <td> <xsl:value-of select="hoten"/> </td>
              <td> <xsl:value-of select="ngaysinh"/> </td>
              <td> <xsl:value-of select="hsluong"/> </td>
              <td> <xsl:value-of select="hsluong*730000"/> </td>
            </tr>
          </xsl:for-each>
        </table>

        <h2 ALIGN="Right">THỦ TRƯỞNG ĐƠN VỊ</h2>
        <br/>
        <center>-----</center>
      </xsl:for-each>
    </body>
  </html>
</xsl:template>
```

### 1.3. CSS

```
<xsl:template match="congy">
<html>
<head>
<title>Ví dụ</title>
<style>
.TieuDe{font-weight:bold;color:Blue;font-size: 25px;background-color:Red;}
.so{text-align:right;}
.chuoi{ text-align: left}
</style>
</head>
<body>
<h1 align="center">BẢNG LƯƠNG THÁNG 11-2009</h1>
<xsl:for-each select="donvi">
<h2>Mã đơn vị: <xsl:value-of select="@madv"/></h2>
<h2>Tên đơn vị: <xsl:value-of select="tendv"/></h2>
<h2>Điện thoại: <xsl:value-of select="dienthoai"/></h2>
<h2 align="center"> DANH SÁCH NHÂN VIÊN </h2>
<table border="2" width="100%" cellpadding="0">
<tr class="TieuDe">
<th>Số TT</th>
<th>Mã NV</th>
<th>Họ tên</th>
<th>Ngày sinh</th>
<th>Hệ số lương</th>
<th>Lương</th>
</tr>
<xsl:for-each select="nhanvien">
<!--<xsl:for-each select="nhanvien[hsluong=3]">-->
<!--<xsl:if test="hsluong=3">-->
<tr>
<td class="so"> <xsl:value-of select="position()"/> </td>
<td class="chuoi"><xsl:value-of select="manv"/></td>
<td class="chuoi"><xsl:value-of select="hoten"/></td>
...

```

**Bài 2:** Cho file **Danhsach.xml** lưu trữ thông tin nhân sự của các công ty:

```
<DS>
<congy TenCT="Công ty dược Nam Hà">
<donvi>
<madv>A01</madv>
<tendv> phòng tổ chức </tendv>
<dienthoai>0437856868</dienthoai>
<nhanvien>
<manv>tc01</manv>
<hoten>Lê Thu Lan</hoten>
<ngaysinh>1988-12-21</ngaysinh>
<gioitinh>nữ</gioitinh>
<ngaycong>23</ngaycong>
</nhanvien>
...
</donvi>
...
</congy>
...
</DS>
```

Sử dụng **XSLT** để hiển thị thông tin theo mẫu sau:

**\*Trong đó: Lương** được tính theo công thức sau: 20 ngày công đầu lương là 150000đ/1ngày; 5 ngày tiếp theo lương là 200000đ/1 ngày; còn lại tính 250000đ/1 ngày.

### **BẢNG LƯƠNG THÁNG**

<b>Tên công ty:</b> (ví dụ CÔNG TY DƯỢC NAM HÀ) <b>Tên phòng:</b> ? (hiển thị tên đơn vị)				
STT	Họ tên	Ngày sinh	Ngày công	Lương

Tạo CSS để định dạng:

- + Nền bảng màu vàng
- + Dữ liệu số trong bảng màu đỏ, cỡ chữ 24px