

TECHNOLOGIE SIECI WEB

Prowadzący: dr inż. **Jan Prokop**, e-mail: jprokop@prz.edu.pl,
Politechnika Rzeszowska, Wydział Elektrotechniki i Informatyki

LABORATORIUM

ĆWICZENIE nr 6

Temat: Podstawy języka XSLT

Specyfikacje języków:

XSLT: <http://www.w3.org/TR/xslt20/>

XPath: <http://www.w3.org/TR/xpath20/>

- Plik lista.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="lista.xsl"?>
<lista>
  <pracownik foto="foto1.jpg" www="http://www.onet.pl"> CV Pracownik1
    <nazwisko>Jan Prokop</nazwisko>
    <uczelnia>Politechnika Rzeszowska</uczelnia>
    <wydzial>Wydział Elektrotechniki i Informatyki</wydzial>
    <adres>ul. W. Pola 2, 35-959 Rzeszów</adres>
    <telefon>(0-prefix-17) 8651384</telefon>
    <mail>jprokop@prz.rzeszow.pl</mail>
    <wzrost>175</wzrost>
  </pracownik>
  <pracownik foto="foto2.jpg" www="http://www.wp.pl"> CV Pracownik2
    .....
  </pracownik>
  .....
</lista>
```

1. Język XSLT - element pętli

- Plik lista.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>Pracownicy</h2>
    <table border="1">
      <tr bgcolor="#00ff00">
        <th>Imię i Nazwisko</th>
        <th>Uczelnia</th>
      </tr>
      <xsl:for-each select="lista/pracownik">
        <tr>
          <td><xsl:value-of select="nazwisko"/></td>
          <td><xsl:value-of select="uczelnia"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

2. Język XSLT - Sortowanie

- Plik lista.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>Pracownicy - sortowanie po e-mail</h2>
    <table border="1">
      <tr bgcolor="#00ff00"><th>Imię i Nazwisko</th><th>E-mail</th></tr>
      <xsl:for-each select="lista/pracownik">
        <xsl:sort select="mail"/>
        <tr>
          <td><xsl:value-of select="nazwisko"/></td>
          <td><xsl:value-of select="mail"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body></html>
</xsl:template>
</xsl:stylesheet>
```

3. Język XSLT - Element wyboru

- Plik lista.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>Wzrost większy niż 180 cm</h2>
    <table border="1">
      <tr bgcolor="#00ff00">
        <th>Imię i Nazwisko</th>
        <th>Wzrost</th>
      </tr>
      <xsl:for-each select="lista/pracownik">
        <tr>
          <td><xsl:value-of select="nazwisko"/></td>
          <xsl:choose>
            <xsl:when test="wzrost>'180'">
              <td bgcolor="#ff00ff">
                <xsl:value-of select="wzrost"/></td>
            </xsl:when>
            <xsl:otherwise>
              <td><xsl:value-of select="wzrost"/></td>
            </xsl:otherwise>
          </xsl:choose>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

4. Język XSLT - Dostęp do wartości atrybutów i węzła tekstowego

- Plik lista.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">
  <html>
```

```

<body>
<hr />
  <xsl:for-each select="lista/pracownik">
    <font color="red" size="+3">
      <xsl:value-of select="nazwisko" /></font><p />
    <h3><xsl:value-of select="text()" /></h3><p />
    <img><xsl:attribute name="src">
      <xsl:value-of select="@foto" />
    </xsl:attribute></img><p />
    <b>Adres WWW: </b>
    <a target="_blank"><xsl:attribute name="href">
      <xsl:value-of select="@www" /></xsl:attribute>
      <xsl:value-of select="@www" /></a><p /><hr />
  </xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

5. XSLT - Dopasowanie na podstawie klucza

- Plik osoby.xml

```

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="osoby.xsl"?>
<osoby>
  <osoba nazwisko="Prokop" pesel="123456"/>
  <osoba nazwisko="Kowalski" pesel="234567"/>
  <osoba nazwisko="Nowak" pesel="345678"/>
</osoby>

```

- Plik osoby.xsl

```

<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:key name="klucz" match="osoba" use="@pesel"/>
<xsl:template match="/">
  <html>
  <body>
    <h3>Dopasowanie według klucza PESEL</h3>
    <xsl:for-each select="key('klucz','234567')"><p>
      <b>Nazwisko: </b> <xsl:value-of select="@nazwisko"/><br />
      <b>PESEL: </b> <xsl:value-of select="@pesel"/></p>
    </xsl:for-each>
  </body>
  </html>
</xsl:template>
</xsl:stylesheet>

```

6. Transformacja za pomocą metody transformNode() języka Java Script

```

<html>
<body>
<script type="text/javascript">
var xmldoc;
var xsldoc;
  xmldoc = new ActiveXObject('Microsoft.XMLDOM');
  xmldoc.async = 'false';
  xmldoc.load("Nazwa_Pliku_XML.xml");
  xsldoc = new ActiveXObject('Microsoft.XMLDOM');
  xsldoc.async = 'false';
  xsldoc.load("Nazwa_Pliku_XSL.xsl");
  document.write(xmldoc.transformNode(xsldoc));
</script>
</body>
</html>

```

7. XSLT - wybrane wzorce dopasowania

7.1. Dopasowanie węzłów

/	Dopasuje element główny dokumentu
/name	Dopasowuje element główny o nazwie <name>
name	Dopasowuje dowolny element o nazwie <name>
//name	Dopasowuje wszystkich potomków elementu <name>
.	Dopasowuje bieżący węzeł
./name	Dopasowuje wszystkich potomków o nazwie name bieżącego węzła
..	Dopasowuje rodzica bieżącego węzła
namespace:name	Dopasowuje węzeł elementu <name> w przestrzeni namespace
namespace:*	Dopasowuje dowolny węzeł elementu w określonej przestrzeni
processing-instruction()	Dopasowanie instrukcji przetwarzania
processing-instruction('target')	Dopasowanie instrukcji przetwarzania o nazwie target (<?target ...?>)

7.2. Dopasowanie gałęzi

name[3]	Dopasowuje trzeci element będący dzieckiem bieżącego węzła
name[last()]	Dopasuje ostatnie dziecko węzła kontekstowego (Uwaga ! Nie ma funkcji o nazwie first())
/book/chapter[9]/paragraf[3]	Dopasuje trzeci akapit <paragraf> dziewiątego rozdziału <chapter> książki <book>
name[firstname="XXXX"]	Dopasowuje element z podelementem o wartości XXXX
name[position() < 6]	Dopasowuje 5 pierwszych dzieci bieżącego węzła
name[position() mod 2 = 0]	Dopasowuje węzły parzyste

7.3. Dopasowanie atrybutów

@attr	Odwołanie do wartości atrybutu attr dla bieżącego węzła
@*	Dopasowuje wszystkie atrybuty bieżącego węzła
name[@attr]	Dopasowuje element <name> z atrybutem attr
name[@*]	Dopasowuje element <name> z dowolnym atrybutem
*[@attr]	Dopasowuje wszystkie elementy z atrybutem attr
name[@attr="XXXX"]	Dopasowuje element <name> z atrybutem attr o wartości XXXX
name[not(@attr)]	Dopasuje elementy name bez atrybutu attr
name[@attr="XXXX"] [3]	Dopasowuje 3 dziecko elementu z atrybutem attr o wartości XXXX
name[3][@attr="XXXX"]	Dopasowuje 3 element <name> z atrybutem attr o wartości XXXX
name[@attr1 and @attr2]	Dopasowuje element który ma zarówno atrybut attr1 jak i atrybut attr2
../@attr	Dopasowuje atrybut attr rodzica bieżącego węzła

7.4. Dopasowanie kilku elementów

name1 name2	Dopasuje name1 lub name2
//name1 //name2	Dopasowuje wszystkie elementy o nazwach name1 i name2 w dokumencie
/name1 //name2	Dopasowuje element o nazwie name1 i wszystkie elementy o nazwie name2 w dokumencie

7.5. Dopasowanie węzła tekstowego i komentarza

text()	Dopasowuje wszystkie węzły tekstowe, które są „dziećmi” węzła kontekstowego
comment()	Dopasowuje węzły komentarzy

7.6. Dopasowanie nieznanych elementów

*	Dopasowuje nieznane elementy
/	Dopasowuje wszystkie elementy będące „wnukami” bieżącego elementu
/*/*/name	Dopasowuje wszystkie elementy mające 2 przodków
//*	Dopasowuje wszystkie elementy w dokumencie