

XSLT

Цели на упражнението:

- Използване на основни функции на XSLT

Средства за обработка на XSLT:

За реализация на това упражнения използвайте браузер с вграден XSLT процесор като например Microsoft Internet Explorer, Microsoft Edge или Mozilla Firefox.

Специално за Mozilla Firefox, извършете следната промяна в настройките:

- отворете локалната страница "about:config"
- натиснете върху бутона за приемане на риска
- потърсете свойството с име `privacy.file_unique_origin`
- ако `privacy.file_unique_origin` е със стойност `true`, променете я на `false`.

Задача 1: За дадения по-долу XML документ създайте XSL документ, който чрез `<xsl:if>`, извежда в HTML формат:

1. Стойностите на всички track елементи, които имат стойност на атрибута `length` '4:04'
2. Стойностите на всички track елементи, чиято дължина е по-малка от 15, както и техните дължини
3. Всички track елементи на четни/нечетни позиции

XML документ:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="catalogDisplay.xsl"?>
<catalog>
  <cd category="Techno" id="8coa600b">
    <artist>TATU</artist>
    <title>200 po vstrechnoy</title>
    <year>2002</year>
    <tracklist num="1">
      <track length="4:04">Zachem ya? (Tell Me Why)</track>
      <track length="3:34">Ya soshla s uma (Mad as Hell)</track>
      <track length="4:40">Nas ne dogonyat (On The Run)</track>
      <track length="4:39">Doschitay do sta (Countdown)</track>
      <track length="3:21">30 minut (Another Minute)</track>
    </tracklist>
  </cd>
</catalog>
```

```

    <track length="4:19">Ya tvoy vrag (Wake Up, Time To Die)</track>
    <track length="4:20">Ya tvoya ne pervaya (First In First Out)</track>
    <track length="3:55">Robot (Robotronik)</track>
    <track length="4:04">Malchik-gay (Zoophilic Lolita)</track>
    <track length="3:53">Nas ne dogonyat (HarDrum rmx)</track>
    <track length="4:04">30 minut (HarDrum rmx)</track>
  </tracklist>
</cd>
<cd category="Techno" id="cd100010">
  <artist>Tatu</artist>
  <title>Zvezdnaya seriya 2001</title>
  <year>2001</year>
  <tracklist num="2">
    <track length="4:04">Ya soshla s uma</track>
    <track length="3:53">Robot</track>
    <track length="4:08">Zachem ya?</track>
    <track length="4:36">Doschitaj do sta</track>
    <track length="4:17">Ya tvoj vrag</track>
    <track length="4:38">Nas ne dogonyat</track>
    <track length="3:18">Mal'chik-gej</track>
    <track length="3:18">30 minut</track>
    <track length="4:18">Ya ne tvoya pervaya</track>
    <track length="3:50">Nas ne dogonyat (hardrum remix)</track>
    <track length="4:04">30 minut (hardrum remix)</track>
    <track length="4:03">Ya soshla s uma (dj ram remix)</track>
    <track length="4:10">Ya soshla s uma (hardrum remix)</track>
    <track length="3:37">Ya soshla s uma (dj ram breakbeat remix)</track>
    <track length="7:00">Ya soshla s uma (S.Galoyan remix)</track>
    <track length="5:36">Ya soshla s uma (Video)</track>
  </tracklist>
</cd>
</catalog>

```

Решение

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:template match="/">
    <html>
      <body>
        <table>
          <tr>
            <td>1.
              <xsl:for-each select="//track">
                <xsl:if test="@length='4:04'">
                  <li><xsl:value-of select="text()"/> </li>
                </xsl:if>
              </xsl:for-each>
            </td>
          </tr>
        </table>
      </body>
    </html>
  </template>
</xsl:stylesheet>

```

```

        </td>
    </tr>
    <tr>
        <td>2.
        <xsl:for-each select="//track">
            <xsl:if test="string-length(.) < 15">
                <li><xsl:value-of select="text()"/> (<xsl:value-of select="string-length(.)"/>)</li>
            </xsl:if>
        </xsl:for-each>
        </td>
    </tr>
    <tr>
        <td>3.
        <xsl:for-each select="//track">
            <xsl:if test="position() mod 2 = 0">
                <li><xsl:value-of select="text()"/> </li>
            </xsl:if>
        </xsl:for-each>
        </td>
    </tr>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Задача 2: За дадения в задача 1 XML документ, създайте XSL документ, който чрез `<xsl:choose>` за всеки елемент `track`, извежда в HTML формат:

1. "A big string" ако дължината на стойността на `track` елемента е по-голяма от 15
2. "A small string" ако дължината на стойността на `track` елемента е по-малка от 15
3. "A medium string" ако дължината на стойността на `track` не отговаря на нито едно от горните условия

Решение

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <xsl:template match="/">
        <html>
            <body>
                <table>
                    <tr>
                        <td>
                            <xsl:for-each select="//track">
                                <xsl:choose>

```

```

        <xsl:when test="string-length(.) > 15">
            <li>A big string:<xsl:value-of select="text()"/> </li>
        </xsl:when>
        <xsl:when test="string-length(.) <= 15">
            <li>A small string:<xsl:value-of select="text()"/></li>
        </xsl:when>
        <xsl:otherwise>
            <li>A medium string:<xsl:value-of select="text()"/></li>
        </xsl:otherwise>
    </xsl:choose>
</xsl:for-each>
</td>
</tr>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Задача 3: За дадения в задача 1 XML документ, създайте XSL документ, който чрез `<xsl:sort>` сортира стойностите на track елементите:

1. В нарастващ ред

2. В намаляващ ред

3. В нарастващ или намаляващ ред в следния формат: track_1, track_2,...,track_n

Решение

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <xsl:template match="/">
        <html>
            <body>
                <table>
                    <tr>
                        <td> 3.1.
                        <ol>
                            <xsl:for-each select="catalog/cd/tracklist/track">
                                <xsl:sort select=""/>
                                <li><xsl:value-of select="text()"/></li>
                            </xsl:for-each>
                        </ol>
                    </td>
                </tr>
            </body>
        </html>
    </xsl:template>

```

```

<td> 3.2.
<ol>
  <xsl:for-each select="catalog/cd/tracklist/track">
    <xsl:sort select="." order="descending"/>
    <li><xsl:value-of select="text()"/></li>
  </xsl:for-each>
</ol>
</td>
</tr>
<tr>
<td> 3.3.
  <xsl:for-each select="catalog/cd/tracklist/track">
    <xsl:sort select="." order="descending"/>
    <xsl:value-of select="text()"/>
    <xsl:text>_</xsl:text>
    <xsl:number value="position()" />
    <xsl:if test="position() != last()">
      <xsl:text>, </xsl:text>
    </xsl:if>
  </xsl:for-each>
</td>
</tr>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Задача 4: За XSL документа, от задача 3, дефинирайте константа като използвате `<xsl:variable>` и изведете стойността ѝ

Решение

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:template match="/">
    <html>
      <body>
        <table>
          <tr>
            <td>
              <table border="2" bordercolor="blue" cellpadding="2">
                <xsl:variable name="header" >
                  Description
                </xsl:variable>
                <xsl:variable name="header2" select="'Alternative Description'"/>

```

```

        <xsl:for-each select="catalog/cd/tracklist/track">
            <xsl:sort select="text()" order="descending"/>
            <tr>
                <th>
                    <xsl:value-of select="$header2"/>
                </th>
            </tr>
            <tr>
                <td>
                    <xsl:value-of select="."/>
                </td>
            </tr>
        </xsl:for-each>
    </table>
</td>
</tr>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Задача 5: За XSL документа, от задача 3, дефинирайте параметър като използвате `<xsl:param>` и изведете стойността му

Решение

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <xsl:template match="/">
        <xsl:param name="param1" select="Track title"/>
        <html>
            <body>
                <table>
                    <tr>
                        <td>
                            <table border="2" bordercolor="blue" cellpadding="2">
                                <xsl:for-each select="catalog/cd/tracklist/track">
                                    <xsl:sort select="text()" order="descending"/>
                                    <tr>
                                        <th>
                                            <xsl:value-of select="$param1"/>
                                        </th>
                                    </tr>
                                    <tr>
                                        <td>

```

```

        <xsl:value-of select="."/>
      </td>
    </tr>
  </xsl:for-each>
</table>
</td>
</tr>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Задача 6: За XSL документа, от задача 3, като използвате <xsl:call-template>, дефинирайте и извикайте следните шаблони:

1. Шаблон с име year, който извежда стойността на year елемента
2. Шаблон с име title, който извежда стойността на title елемента
3. Шаблон с име artist, който извежда стойността на artist елемента

Решение

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:template match="/">
    <html>
      <body>
        <table border="2" cellspacing="0px" cellpadding="5px">
          <tr>
            <th>Year</th>
            <th>Title</th>
            <th>Artist</th>
          </tr>
          <xsl:for-each select="catalog/cd">
            <tr>
              <td>
                <xsl:call-template name="year"/>
              </td>
              <td>
                <xsl:call-template name="title"/>
              </td>
              <td>
                <xsl:call-template name="artist"/>
              </td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>

```

```

        </xsl:for-each>
    </table>
</body>
</html>
</xsl:template>
<xsl:template name="year">
    <b><xsl:value-of select="year"/></b><br/>
</xsl:template>
<xsl:template name="title">
    <b><xsl:value-of select="title"/></b><br/>
</xsl:template>
<xsl:template name="artist">
    <b><xsl:value-of select="artist"/></b><br/>
</xsl:template>
</xsl:stylesheet>

```

Задача 7: За XSL документа, от задача 3, като използвате `<xsl:call-template>` и `<xsl:with-param>`, дефинирайте и извикайте следните шаблони с параметри:

1. Шаблон с име `year`, който има параметър с име `yearName` и извежда стойността на `year` елемента
2. Шаблон с име `title`, който има параметър с име `titleName` и извежда стойността на `title` елемента
3. Шаблон с име `artist`, който има параметър с име `artistName` и извежда стойността на `artist` елемента

Решение

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <xsl:template match="/">
    <html>
        <body>
            <table border="2" cellspacing="0px" cellpadding="5px">
                <tr>
                    <th>Year</th>
                    <th>Title</th>
                    <th>Artist</th>
                </tr>
                <xsl:for-each select="catalog/cd">
                    <tr>
                        <td>
                            <xsl:call-template name="yearWithParam">
                                <xsl:with-param name="yearName" select="year"/>
                            </xsl:call-template>
                        </td>
                        <td>
                            <xsl:call-template name="titleWithParam">

```



```

        <xsl:with-param name="titleName" select="title"/>
    </xsl:call-template>
</td>
<td>
    <xsl:call-template name="artistWithParam">
        <xsl:with-param name="artistName" select="artist"/>
    </xsl:call-template>
</td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
<xsl:template name="yearWithParam">
    <xsl:param name="yearName"/>
    <p>7.1: <xsl:value-of select="$yearName"/></p><br/>
</xsl:template>
<xsl:template name="titleWithParam">
    <xsl:param name="titleName"/>
    <p>7.2: <xsl:value-of select="$titleName"/></p><br/>
</xsl:template>
<xsl:template name="artistWithParam">
    <xsl:param name="artistName"/>
    <p>7.3: <xsl:value-of select="$artistName"/></p><br/>
</xsl:template>
</xsl:stylesheet>

```

Задача 8: За XML документа от задача 1, като използвате <xsl:copy-of>, съставете XSL документ, който връща в XML формат следните 2 под-елемента на елемента cd:

1. Стойността на под-елемента title
2. Под-елемента tracklist заедно с цялата негова структура

Решение

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
    <xsl:template match="catalog/cd">
        <cd>
            <title>
                <xsl:value-of select="title"/>
            </title>
            <xsl:copy-of select="tracklist" />
        </cd>
    </xsl:template>
</xsl:stylesheet>

```

```
</xsl:template>
</xsl:stylesheet>
```

Задача 9: Решете задача 8 като използвате <xsl:element> и <xsl:attribute> вместо <xsl:copy-of>

Решение

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
  <xsl:template match="catalog">
    <xsl:element name="result">
      <xsl:for-each select="cd">
        <xsl:element name="cd">
          <xsl:element name="title">
            <xsl:value-of select="title"/>
          </xsl:element>
          <xsl:element name="tracklist">
            <xsl:attribute name="id">
              <xsl:value-of select="tracklist/@num"/>
            </xsl:attribute>
            <xsl:for-each select="tracklist/track">
              <xsl:element name="track">
                <xsl:attribute name="length">
                  <xsl:value-of select="@length"/>
                </xsl:attribute>
                <xsl:value-of select="text()"/>
              </xsl:element>
            </xsl:for-each>
          </xsl:element>
        </xsl:for-each>
      </xsl:element>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

Задача 10: За XML документа от задача 1, като използвате <xsl:apply-templates>, създайте XSL документ, който съдържа неименовани шаблони селектиращи стойностите на следните елементи и техните атрибути (ако са дефинирани):

1. title

2. year

3. track

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
  <xsl:template match="/">
    <html>
      <head>
        <title>Tatu Catalog</title>
      </head>
      <body>
        <h2>Tatu Catalog</h2>
        <table border="2" cellspacing="0px" cellpadding="5px">
          <thead>
            <tr>
              <th>CD Title</th>
              <th>Year</th>
              <th>Tracks</th>
            </tr>
          </thead>
          <tbody valign="top">
            <xsl:apply-templates select="catalog"/>
          </tbody>
        </table>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="cd">
    <tr>
      <td>
        <xsl:apply-templates select="title"/>
        <xsl:for-each select="title/@*">
          <br/>Attribute name:<xsl:value-of select="name()"/> - attribute value: <xsl:value-of select="."/>
        </xsl:for-each>
      </td>
      <td>
        <xsl:apply-templates select="year"/>
        <xsl:for-each select="year/@*">
          <br/>Attribute name:<xsl:value-of select="name()"/> - attribute value: <xsl:value-of select="."/>
        </xsl:for-each>
      </td>
      <td>
        <ol>
          <xsl:apply-templates select="tracklist/track"/>
        </ol>
      </td>
    </tr>
  </xsl:template>
  <xsl:template match="catalog/cd/tracklist/track">

```

```

</li>
  <xsl:apply-templates select="text()"/>
  <xsl:for-each select="@*">
    <br/><i>Attribute name:</i><xsl:value-of select="name()"/> - <i>atribute value:</i> <xsl:value-of
select="."/>
  </xsl:for-each>
</li>
</xsl:template>
</xsl:stylesheet>

```

Задача 11: Решете задача 10, като използвате <xsl:apply-templates> заедно с атрибута mode и по този начин резултатният XSL документ съдържа **неименовани шаблони** селектиращи следните стойности:

1. Стойността на елемента title
2. Стойността на елемента year
3. Стойностите на елемента track без неговите атрибути (при mode="withoutAttributes")
4. Стойностите на елемента track заедно с неговите атрибути (при mode="withAttributes")

Решение

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
  <xsl:template match="/">
    <html>
      <head>
        <title>Tatu Catalog</title>
      </head>
      <body>
        <h2>Tatu Catalog</h2>
        <table border="2" cellpadding="5px" cellspacing="0px">
          <thead>
            <tr>
              <th>CD Title</th>
              <th>Year</th>
              <th>Tracks without atributes</th>
              <th>Tracks with atributes</th>
            </tr>
          </thead>
          <tbody valign="top">
            <xsl:apply-templates select="catalog"/>
          </tbody>
        </table>
      </body>
    </html>

```

```

</xsl:template>
<xsl:template match="cd">
  <tr>
    <td>
      <xsl:apply-templates select="title"/>
    </td>
    <td>
      <xsl:apply-templates select="year"/>
    </td>
    <td>
      <ol>
        <xsl:apply-templates select="tracklist/track" mode="withoutAttributes"/>
      </ol>
    </td>
    <td>
      <ol>
        <xsl:apply-templates select="tracklist/track" mode="withAttributes"/>
      </ol>
    </td>
  </tr>
</xsl:template>
<xsl:template match="catalog/cd/tracklist/track" mode="withAttributes">
  <li>
    <xsl:apply-templates select="text()"/>
    <xsl:for-each select="@*">
      <br/><i>Attribute name:</i><xsl:value-of select="name()"/> - <i>atribute value:</i> <xsl:value-of
select="."/>
    </xsl:for-each>
  </li>
</xsl:template>
<xsl:template match="catalog/cd/tracklist/track" mode="withoutAttributes">
  <li>
    <xsl:apply-templates select="text()"/>
  </li>
</xsl:template>
</xsl:stylesheet>

```