

XPATH и XQUERY

Цели на упражнението:

1. Използване на основни функции на XPATH в XSL
2. Използване на основни функции на XQUERY

Средства за обработка на XPATH и XQUERY изрази:

За изпълнението на това упражнения използвайте следните инструменти:

XPATH и XQUERY процесор

XQUERY процесор

Задачите свързани с XPATH могат да бъдат тествани с Internet Explorer и/или Mozilla Firefox, като XPATH изразите бъдат включени в XSL документ по начин подобен на дадения пример в задача 1.

Задача 1: Създайте файлове със следните XSL и XML документи и разгледайте XML документа с Internet Explorer или Mozilla Firefox. За атрибута select на елемента `xsl:value-of` заменете функцията `text()` с `'` и вижте новия резултат в Internet Explorer.

Може да включите XPATH изразите от задачи 2 и 3 в дадения по-долу XSL документ по подобен начин.

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="4:10">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>
    </tracklist>
  </cd>
</catalog>
```

```

    <track length="3:21">30 minut (Another Minute)</track>
    <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="3:20">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="3:31">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="4:03">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>

```

XSL документ - catalogDisplay.xsl:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <body>
        <h2>My CD Collection</h2>
        <table border="1">
          <tr bgcolor="#9acd32">
            <th align="left">Title</th>
            <th align="left">Artist</th>
            <th align="left">Year</th>
            <th align="left">Category</th>
            <th align="left">ID</th>

```

```

</tr>
<xsl:for-each select="catalog/cd">
<tr>
  <td><xsl:value-of select="title"/></td>
  <td><xsl:value-of select="artist"/></td>
  <td><xsl:value-of select="year"/></td>
  <td><xsl:value-of select="./@category"/></td>
  <td><xsl:value-of select="./@id"/></td>
</tr>
<tr>
  <td>
    <xsl:for-each select="tracklist/track">
      <li><xsl:value-of select="text()"/></li>
    </xsl:for-each>
  </td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Задача 2: За XML документа от задача 1 напишете XPATH изрази, които извеждат:

1. Стойностите на всички track елементи, чиито атрибут length е равен на '4:04' и са включени в елемент cd, имащ id равно на 8c0a600b
2. Всички track елементи на четни/нечетни позиции
3. Стойностите на всички track елементи, чиято дължина на текста е по-голяма от 35
4. Дължината на текста на всички track елементи, чиято дължина на текста е по-голяма от 15
5. Последния track елемент от всяко cd
6. Петия track елемент от всяко cd
7. Броя на track елементите за всяко cd
8. Всички track елементи, които съдържат 'Ya soshla s uma'
9. Всички track елементи, които започват с буквата 'D'
10. Всички track елементи, които завършват с изрази 'sta'
11. Стойностите на всички track елементи, разпечатани с главни букви

12. Стойността на елемента year, който е под-елемент на елемента cd, съдържащ под-под-елемент track с length = '3:55' и имащ стойност 'Robot (Robotronik)'

13. Среден брой track елементи от всички налични cd елементи

14. За всеки елемент cd изведете стойността на под-елементите му title и year, спазвайки следния модел:

Заглавие: title_value; Година на издаване: year_value

Упътване

Решение

1. `catalog/cd[@id='8coa600b']/tracklist/track[@length='4:04']`
2. `catalog/cd/tracklist/track[position() mod 2 = 0]`
3. `catalog/cd/tracklist/track[string-length(.) > 35]`
4. `catalog/cd/tracklist/track[string-length(.) > 15]/string-length(.)`
5. `catalog/cd/tracklist/track[last()]`
6. `catalog/cd/tracklist/track[5]`
7. `catalog/cd/tracklist/count(track)`
8. `catalog/cd/tracklist/track[contains(text(),'Ya soshla s uma')]`
9.
 - `catalog/cd/tracklist/track[starts-with(text(),'D')]`
 - `catalog/cd/tracklist/track[matches(text(),'^D.*')]`
10.
 - `catalog/cd/tracklist/track[ends-with(text(),'sta')]`
 - `catalog/cd/tracklist/track[matches(text(),'sta$')]`
11.
 - `catalog/cd/tracklist/track/upper-case(text())`
 - `catalog/cd/tracklist/track/translate(text(),'abcdefghijklmnopqrstuvwxyz','ABCDEFGHIJKLMNOPQRSTUVWXYZ')`
12.
 - `catalog/cd[tracklist/track[@length='3:55' and text()='Robot (Robotronik)']]/year`
 - `catalog/cd[tracklist/track[@length='3:55'][text()='Robot (Robotronik)']]/year`
13.
 - `sum(catalog/cd/tracklist/count(track)) div catalog/count(cd)`
 - `count(catalog/cd/tracklist/track) div catalog/count(cd)`

Задача 3: За XML документа от задача 1, като използвате оси (axes), съставете XPATH изрази, които :

1. Избират елемента:

1. tracklist
2. year

и извеждат негова стойност или стойност на негов атрибут

2. Избират атрибутите num и id съответно на елементите tracklist и cd

3. Избират всички елементи track, които се намират преди track елемента със стойност 'Doschitay do sta (Countdown)'

4. Избират всички елементи track, които се намират след track елемента със стойност 'Doschitay do sta (Countdown)'

5. Избират всички елементи track, които се намират след track елемента със стойност 'Doschitay do sta (Countdown)' и имат стойност на атрибута length '4:04'

6. Всички стойности на под-елементите на всички елементи cd в документа

7. Стойностите на всички елементи, които имат атрибут с име id

8. Стойностите на всички елементи, които имат какъвто и да било атрибут

9. Атрибута num с максимална стойност

Решение

1.
 - self::node()/descendant-or-self::node()/child::tracklist/attribute::num
 - catalog/cd/child::tracklist/attribute::num
 - catalog/cd/child::year
2.
 - catalog/cd/tracklist/attribute::num
 - catalog/cd/attribute::id
3. catalog/cd/tracklist/track[text()='Doschitay do sta (Countdown)']/preceding-sibling::*
4. catalog/cd/tracklist/track[text()='Doschitay do sta (Countdown)']/following-sibling::*
5. catalog/cd/tracklist/track[text()='Doschitay do sta (Countdown)']/following-sibling::track[@length='4:04']
6. //cd/descendant::*
7. //attribute::id
8. //attribute::*
9.
 - catalog/cd/tracklist[@num and count(self::*[xs:int(@num) < //@num]) = 0]/@num
 - catalog/cd/tracklist[@num=max(//*[@num != "]/number(@num))]/@num
 - catalog/cd/tracklist[@num=max(//*/@num)]/@num
 - max(//*/@num)

Задача 4: За XML документа от задача 1, съставете XQUERY израз:

1. Селектираш всички стойности на елемента track, който е под-елемент на tracklist, имащ атрибут num равен на 1.
2. Подредете резултата от 1. по азбучен ред
3. Подредете резултата от 1. в обратен азбучен ред

2. Създаващ следната структура:

```
<records>
  <record cd_ID="CD_ID_VALUE" artist="ARTIST_VALUE">
    <info>Title: TITLE_VALUE, Year: YEAR_VALUE, Track numbers: COUNT_OF_TRACKS</info>
  </record>
</records>
```

в която стойностите на CD_ID_VALUE, ARTIST_VALUE, TITLE_VALUE, YEAR_VALUE, COUNT_OF_TRACKS отговарят съответно на стойностите на атрибута id на елемента cd, на елемента artist, на елемента title, на елемента year, на броя на елементите track за съответния елемент cd

3. Създаващ списък със стойността на всички track елементи от всички cd елементи, следващ модела:

```
<tracks>
  <track> TRACK_NAME_1</track>
  <track> TRACK_NAME_2</track>
  <track> .....</track>
  <track> TRACK_NAME_N</track>
</tracks>
```

Упътване

Решение

1.
 - Примерно решение 1 (<http://videlibri.sourceforge.net/cgi-bin/xidelcgi>):
for \$o in catalog/cd/tracklist
where \$o/@num = "1"
return \$o/track/text()
 - Примерно решение 2 (<http://videlibri.sourceforge.net/cgi-bin/xidelcgi>):
for \$o in catalog/cd
let \$l := \$o/tracklist
where \$l/@num = "1"
return \$l/track
 - Примерно решение 3 (<http://videlibri.sourceforge.net/cgi-bin/xidelcgi>):
for \$o in catalog/cd/tracklist[@num = "1"]/track
order by \$o ascending/descending
return \$o/text()
2.
 - Примерно решение 1 (<http://videlibri.sourceforge.net/cgi-bin/xidelcgi>):
element records
{
 for \$o in catalog/cd

```

return
element record {
  attribute cd_ID {$o/@id}, attribute artist {$o/artist},
  element info {
    text { "Title:"}, text {$o/title}, text { ", Year:"} ,text{$o/year}, text { ", Track numbers:"},
    text{count($o/tracklist/track)}
  }
}
}

```

- Примерно решение 2 (<http://try.zorba.io/>):

```

for $o in doc("mydoc.xml")/catalog
return
element records
{
  for $p in $o/cd
  return
  element record {
    attribute cd_ID {$o/@id}, attribute artist {$o/artist},
    element info {
      text { "Title:"}, text {$o/title}, text { ", Year:"} ,text{$o/year}, text { ", Track numbers:"},
      text{count($o/tracklist/track)}
    }
  }
}

```

3. for \$o in .

```

return
element tracks
{
  for $p in $o/catalog/cd/tracklist/track/text()
  return element track {$p}
}

```

Задача 5: Решете задача 4 като дефинирате една XQUERY функция и след това я използвате

Упътване

Решение

```

1. declare function local:trackElement($catalog as element()) as element()*
{
  for $o in $catalog/cd/tracklist[@num = "1"]/track
  order by $o descending
  return $o
};

```

```
{
  for $s in doc("mydoc.xml")/catalog return local:trackElement($s)
}
```

2. declare function local:trackElement(\$catalog as element()) as element()*

```
{
  for $o in $catalog/cd
  return
  element record {
    attribute c {$o/@id}, attribute artist {$o/artist},
    element info {
      text { "Title:"}, text {$o/title}, text { ", Year:"} ,text{$o/year}, text { ", Track numbers:"},
      text{count($o/tracklist/track)}
    }
  }
};
```

```
for $o in .
return
element records
{
  {for $s in doc("mydoc.xml")/catalog return local:trackElement($s) }
}
```

3. declare function local:trackElement(\$catalog as element()) as element()*

```
{
  for $p in $catalog/cd/tracklist/track/text()
  return element track {$p}
};
```

```
for $o in .
return
element tracks
{
  {for $s in doc("mydoc.xml")/catalog return local:trackElement($s) }
}
```