Упражнение 4 – XPath и XQuery

Изготвено от: Кристияна Николова ФН: 62372

Задача 1:

Смяната на text() с '.' не променя резултата. Има два варианта за показване на получения резултат, защото резултатът не се променя при използването на двата начина.

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

catalog/cd[@id="8c0a600b"]/tracklist/track[@length="4:04"]

→Взимаме track елементите, които имат атрибут length, равен на „4:04“. Това става с track[@length="4:04"]. Задаваме, че атрибутът id на cd трябва да е равно на 8c0a600b. Това става с cd[@id="8c0a600b"].

2. Всички track елементи на четни/нечетни позиции

catalog/cd/tracklist/track[position() mod 2 = 0] –> четни

catalog/cd/tracklist/track[position() mod 2 != 0] –> нечетни

→Взимаме всички track елементи, които са на четни или нечетни позиции. Позицията на даден елемент получаваме така: track[position()]. С mod 2 разбираме дали позицията на елемента е четна или нечетна.

3. Стойностите на всички track елементи, чиято дължина на текста е по-голяма от 35:

catalog/cd/tracklist/track[string-length() > 35]

→ Функцията, която ни дава дължината на стринга е string-length(). Използваме я, за да намерим стойностите на всички track елементи с дължина на текста по-голяма от 35.

4. Дължината на текста на всички track елементи, чиято дължина на текста е по-голяма от 15:

catalog/cd/tracklist/track[string-length() > 15]/string-length()

→Подобно на задачата горе намираме първо всички track елементи, които имат дължина на текста по-голяма от 15. След това, отново с функцията string-length(), извеждаме дължината на текста на всички match-нати track елементи.

5. Последния track елемент от всяко cd:

catalog/cd/tracklist/track[last()]

→ Функцията, която ни връща последния елемент, е last(). Използваме я, за да намерим последния track елемент от всяко cd.

6. Петия track елемент от всяко cd:

catalog/cd/tracklist/track[position() = 5]

→ Функцията position() връща позицията на елемента. Търсим петия track елемент от всяко cd, следователно го постигаме с //track[position() = 5]

7. Броя на track елементите за всяко cd:

catalog/cd/tracklist/count(track)

→ Функцията, която връща броя на даден елемент е count(x), където x е търсеният елемент. Следователно намираме броя на track елементите за всяко cd чрез //count(track).

8. Всички track елементи, които съдържат 'Ya soshla s uma':

catalog/cd/tracklist/track[contains(text(), “Ya soshla s uma”)]

→ Функцията, която връща елемент по зададено търсене, е contains(място на търсене, текст за търсене). Следователно използваме //track[contains(text(), “Ya soshla s uma”)], за да намерим всички track елементи, които съдържат Ya soshla s uma.

9. Всички track елементи, които започват с буквата 'D':

catalog/cd/tracklist/track[starts-with(text(), “D”)]

→ Функцията starts-with(място, текст) връща всички елементи, които започват със зададения текст. Следователно използваме //track[starts-with(text(), “D”)], за да намерим всички track елементи, които започват с буквата “D”.

10. Всички track елементи, които завършват с израза 'sta':

catalog/cd/tracklist/track[ends-with(text(), “sta”)]

→ Подобно на задачата горе, функцията ends-with(място, текст) връща всички елементи, които завършват със зададения текст. Следователно използваме //track[ends-with(text(), “sta”)], за да намерим всички track елементи, които завършват с израза “sta”.

11. Стойностите на всички track елементи, разпечатани с главни букви:

catalog/cd/tracklist/track/upper-case(text())

→ С catalog/cd/tracklist/track намираме всички track елементи, а с upper-case(text()) разпечатваме текста, отговарящ на съответния track елемент, 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)':

catalog/cd/tracklist/track[@length=”3:55” and text()=”Robot (Robotronik)”]/year

→C //track[@length=”3:55” and text()=”Robot (Robotronik)”] намираме track елемента, който отговаря на условието, след това извеждаме стойността само на елемента year.

13. Среден брой track елементи от всички налични cd елементи:

count(catalog/cd/tracklist/track) div catalog/count(cd)

→ Функцията count(елемент) връща броя на срещания на търсения елемент. С div правим деление на броя на всички налични cd елементи. Така намираме средния брой на 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

catalog/cd/concat(“Заглавие: “, title/text(), “; Година на издаване: “, year/text())

→С функцията concat можем да добавяме текст.

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

1. Избират елемента:

1.1.tracklist

1.2.year

и извеждат негова стойност или стойност на негов атрибут:

self::node()/descendant-or-self::node()/child::tracklist/attribute::num

//tracklist/@num

→ Двете решения горе са еквивалентни. С //tracklist/@num извеждаме стойността.

catalog/cd/year

→Извеждаме стойността на елемента year.

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

//tracklist/@num

→С @num избираме атрибута num на елементите tracklist, които намираме с //tracklist.

//cd/@id

→С @id избираме атрибута id на елементите cd, които намираме с //cd.

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

catalog/cd/tracklist/track[text() = “Doschitay do sta (Countdown)”]/preceding-sibling::*

→ С track[text()=x], намираме всички track елементи, които имат стойност x. С preceding-sibling::* избираме всички елементи, които се намират преди съответния елемент.

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

catalog/cd/tracklist/track[text() = “Doschitay do sta (Countdown)”]/following-sibling::*

→Аналогично на задачата горе, чрез following-sibling::* получаваме всички елементи, които се намират след съответния елемент.

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':

catalog/cd/tracklist/track[text() = “Doschitay do sta (Countdown)”]/following-sibling::track[@length=”4:04”]

→Аналогично на задачата горе, намираме всички елементи, които се намират след track елемента със стойност 'Doschitay do sta (Countdown)' и имат стойност length “4:04”. Това постигаме чрез following-sibling::track[@length=”4:04”].

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

catalog/cd/descendant::*

→ Descendant връща стойността на наследника на съответния елемент, а чрез descendant::* получаваме всички стойности на под-елементите на всички елементи cd в документа.

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

//attribute::id

//@id

→ Двете решения са еквивалентни. Чрез @id извеждаме стойността на всички елементи, които имат атрибут с име id.

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

//attribute::*

//@*

→ Двете решения са еквивалентни. Аналогично на горната задача, намираме стойността на всички елементи, които имат какъвто и да било атрибут. Това го задаваме чрез @*.

9. Атрибута num с максимална стойност:

catalog/cd/tracklist[@num=max(//*/@num)]/@num

→tracklist[@num=max(//*/@num)] връща елемента с максимална стойност на атрибута num.

→ tracklist[@num=max(//*/@num)]/@num връща стойността на елемента с максимална стойност на атрибута num.

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

1. Селектиращ всички стойности на елемента track, който е под-елемент на tracklist, имащ атрибут num равен на 1:

doc("catalog.xml")//tracklist[@num="1"]/track

→ catalog.xml ни е документът, който съдържа xml-а от условието на задачата.

Търсим елемента track, който е под-елемент на елемента tracklist с атрибут num = 1 (tracklist[@num=”1”]) в документа catalog.xml .

Подредете резултата от 1. по азбучен ред:

for $x in doc(“catalog.xml”)/catalog/cd/tracklist[@num=”1”]/track

order by $x ascending

return $x/text()

→ За всяко x в намерения елемент от горното условие, задаваме подреждане по азбучен ред, което правим чрез order by $x ascending. След това връщаме стойността на намерения елемент чрез $x/text().

1.1. Подредете резултата от 1. в обратен азбучен ред:

for $x in doc(“catalog.xml”)/catalog/cd/tracklist[@num=”1”]/track

order by $x descending

return $x/text()

→ За всяко x в намерения елемент от горното условие, задаваме подреждане по обратен азбучен ред, което правим чрез order by $x descending. След това връщаме стойността на намерения елемент чрез $x/text().

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:

```
<records>
    {
      for $x in doc("catalog.xml")//cd
      return
      <record cd_ID="{$x/@id}" artist="{$x/artist/text()}">
        <info>Title: {$x/title/text()}, Year: {$x/year/text()}, Track numbers:
       {count($x//track)}</info>
      </record>
    }
</records>
```

→ Създаваме елемент records. В него за всеки елемент x в намерения елемент cd от документа catalog.xml създаваме елемент record, който има атрибут cd_ID, равен на стойността на атрибута id на елемента cd, и атрибут artist, равен на стойността на елемента artist в cd. След това създаваме под-елемент info на елемента record, който съдържа в себе си заглавие Title, което отговаря на стойността на заглавието title в cd елемента, година Year, която отговаря на стойността на годината year в cd елемента, и Track numbers, които имат за стойност броя на всички track елементи в съответния cd елемент.

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>
```

```
<tracks>
        {
                for $x in doc(“catalog.xml”)/catalog/cd
```

```
            return
                <track>{$x//track/text()}</track>
        }
</tracks>
```

→ Създаваме елемент tracks. В него за всеки елемент x в намерения елемент cd от документа catalog.xml създаваме елемент track. Елементът track има стойност, отговаряща на стойността на съответния елемент track от намерения елемент cd. Така създаваме списък със стойността на всички track елементи от всички cd елементи, следващ модела от условието на задачата по-горе.

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

1.

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

→ Използваме създадената функция:

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

→ Създаваме функция, която за всяко i в намерения елемен, задаваме подреждане по азбучен ред (или обратен азбучен ред), което правим чрез order by $i ascending/descending. След това връщаме стойността на намерения елемент чрез $i/text().

2.

```
declare function local:trackElement($catalog as element()) as element()* {
    for $i in $catalog/cd
    return
    element record {
      attribute cd_ID {$i/@id}, attribute artist {$i/artist},
      element info {
```

```
            text { "Title:"}, text {$i/title}, text { ", Year:"} ,text{$i/year}, text { ", Track numbers:"},
text{count($i/tracklist/track)}
        }
    }
};
for $k in .
   return
   element records {
      {
         for $s in doc("catalog.xml")/catalog
         return local:trackElement($s)
      }
}
```

→ Създаваме функция която за всяко $i от $catalog/cd създава елемент record, който има атрибут cd_ID, равен на стойността на атрибута id на елемента cd, и атрибут artist, равен на стойността на елемента artist в cd. След това създаваме под-елемент info на елемента record, който съдържа в себе си заглавие Title, което отговаря на стойността на заглавието title в cd елемента, година Year, която отговаря на стойността на годината year в cd елемента, и Track numbers, които имат за стойност броя на всички track елементи в съответния cd елемент.

→ При използването на функцията създаваме елемент records, където за всяко $s от doc(“catalog.xml”)/catalog използваме декларираната вече функция.

3.

```
declare function local:trackElement($catalog as element()) as element()* {
      for $s in $catalog/cd/tracklist/track/text()
      return element track {$s}
   };


for $i in .
   return
      element tracks
      {
```

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

→ Създаваме функция, която за всяко $s от $catalog/cd/tracklist/track/text() създава елемент track със стойност $s. При използване на функцията за всяко $k от doc(“catalog.xml”)/catalog връща резултата от използваната функция.