

Dotazování nad XML dokumenty pomocí XQuery

Řešení příkladů

Irena Mlýnková, Martin Nečaský

Příklad 1:

```
<predmety>
{
  for      $predmet in fn:doc("predmety.xml")//predmet
  where    $predmet/katedra = "13136"
  order by $predmet/nazev
  return
    <nazev-predmetu>{fn:data($predmet/nazev)}</nazev-predmetu>
}
</predmety>
```

Příklad 2:

```
<katedry>
{
  for      $katedra in fn:doc("katedry.xml")//katedra
  where    $katedra/externista
  return
    <nazev-katedry>{fn:data($katedra/nazev)}</nazev-katedry>
}
</katedry>
```

Příklad 3.

```
<predmety>
{
  for      $predmet in fn:doc("predmety.xml")//predmet
  let      $pocet-vyucujicich :=
    fn:count(fn:distinct-values($predmet/(garant|autor)))
  where    $predmet/katedra = "13136"
  return
    <predmet>
      <nazev-predmetu>{fn:data($predmet/nazev)}</nazev-predmetu>
      <pocet-vyucujicich>
        {$pocet-vyucujicich}
      </pocet-vyucujicich>
    </predmet>
}
</predmety>
```

Příklad 4:

```
let $kody-kateder := fn:distinct-values(
    fn:data(fn:doc("predmety.xml"))//katedra
),
$max-pocet := fn:max(
    for $kod in $kody-kateder
    return fn:count(
        fn:doc("predmety.xml")//
        predmet[katedra = $kod]
    )
)
return
    for $kod in $kody-kateder
    where $max-pocet = fn:count(
        fn:doc("predmety.xml")//predmet[katedra = $kod]
    )
    return $kod
```

Příklad 5:

```
<clenove-bez-vyuky>
{
  for $clen in fn:doc("katedry.xml")//(clen|externista)
  where not(fn:doc("predmety.xml")//(autor|garant)=$clen)
  return $clen
}
</clenove-bez-vyuky>
```

Příklad 6:

```
<katedry>
{
  for $kod-katedry in fn:distinct-
values(fn:doc("predmety.xml")//katedra)
  let $katedra := fn:doc("katedry.xml")//katedra[@kod = $kod-katedry]
  return
    <katedra kod="{ $katedra/@kod }">
      { $katedra/nazev }
      {
        for $predmet in fn:doc("predmety.xml")//predmet
        where $predmet/katedra = $kod-katedry
        return
          <predmet>
            { $predmet/nazev }
            { $predmet/(garant|autor) }
          </predmet>
      }
    </katedra>
}
</katedry>
```

Příklad 7:

```
<katedry>
{
  for    $katedra in fn:doc("katedry.xml")//katedra
  let    $predmety-katedry :=
          fn:doc("predmety.xml")//predmet[katedra = $katedra/@kod]
  return
    <katedra kod="{ $katedra/@kod}">
      { $katedra/nazev }
      {
        for    $predmet in $predmety-katedry
        return
          <predmet>
            { $predmet/nazev, $predmet/(garant|autor) }
          </predmet>
      }
    </katedra>
}
</katedry>
```

Příklad 8:

```
<table>
<tr><th>Pořiznak</th><th>Kód</th><th>Název</th></tr>
{
  for    $predmet in fn:doc("predmety.xml")//predmet
  return
    <tr>
      <td>
        {
          if (fn:exists(
            fn:doc("katedry.xml")//katedra[@kod = $predmet/katedra]/
              externista[. = $predmet/(garant|autor)]
          ))
          then text {"E"}
          else text {"I"}
        }
      </td>
      <td>{fn:data($predmet/@kod)}</td>
      <td>{ $predmet/nazev}</td>
    </tr>
}
</table>
```

Příklad 9:

Knihovna s rekurzivní funkcí:

```
module namespace mylib="http://mylib";

declare function mylib:pocet-prerekvizit($predmet as element()) as
xs:integer
{
  let $pocet-prerekvizit := fn:count($predmet/prerekvizita),
      $pocet-pre-prerekvizit :=
        fn:sum(
          for $kod-prerekvizity in $predmet/prerekvizita
          let $prerekvizita :=
            fn:doc("predmety.xml")//predmet[@kod = $kod-
prerekvizity]
          return mylib:pocet-prerekvizit($prerekvizita)
        )
  return
    fn:sum(($pocet-prerekvizit,$pocet-pre-prerekvizit))
};
```

Dotaz importující knihovnu a využívající rekurzivní funkci:

```
import module namespace mylib = "http://mylib" at "query09_1.xq";

<predmety>
{
  for $predmet in fn:doc("predmety.xml")//predmet
  return
    <predmet kod="{ $predmet/@kod }">
      {mylib:pocet-prerekvizit($predmet)}
    </predmet>
}
</predmety>
```

Příklad10:

```
let $maloobchodni-objednavky :=
  for $obj in fn:doc("maloobchod.xml")//objednavka[@datum = "19.9.2007"]
  return
    element objednavka {
      attribute kod {fn:concat("M", $obj/@kod)},
      $obj/zakaznik,
      element cena {fn:sum(
        for $polozka in $obj/polozka
        return $polozka/cena * $polozka/(mnozstvi|kusu)
      )}
    }
let $velkoobchodni-objednavky :=
  for $obj in fn:doc("velkoobchod.xml")//objednavka[datum = "19.9.2007"]
  let $zak := fn:doc("velkoobchod.xml")//zakaznik[@kod =
$obj/zakaznik/@kod]
  return
    element objednavka {
      attribute kod {fn:concat("V", $obj/@kod)},
      element zakaznik {
        $zak/jmeno,
        element adresa {
          fn:concat(
            $zak/adresa/ulice, " ", " ",
            $zak/adresa/mesto, " ", " ",
            $zak/adresa/psc
          )
        }
      },
      element cena {
        fn:sum(
          for $polozka in $obj/polozka
          return $polozka/cena * $polozka/(mnozstvi|kusu)
        )}
      }
  return
    <objednavky>
      {$maloobchodni-objednavky union $velkoobchodni-objednavky}
    </objednavky>
```

Příklad 11:

```
let $spatne-predmety :=
  for $predmet in doc("predmety.xml")//predmet
  where every $vyucujici in $predmet/(garant|autor)
    satisfies doc("katedry.xml")//externista[. = $vyucujici]
  return $predmet
return
  if (fn:exists($spatne-predmety))
  then
    <error>
      {for $predmet in $spatne-predmety
      return
        <nazev-predmetu>{fn:data($predmet/nazev)}</nazev-predmetu>}
    </error>
  else
    <ok/>
```