

A7B36XML, AD7B36XML: **XML Technologies**

Practical Classes 9 and 10:

XQuery: Exercises

12. and 19. 5. 2017

Martin Svoboda

svoboda@ksi.mff.cuni.cz

<http://www.ksi.mff.cuni.cz/~svoboda/courses/2016-2-A7B36XML/>

Exercise 1

- Express the following XQuery query
 - Select owners from country *Česká republika*
 - I.e. find a sequence of all corresponding *owner* elements
 - Order the result according to owner names

Exercise 2



- Express the following XQuery query
 - Create a sequence of all properties transformed with respect to the following pattern

```
<property idProperty="PROPERTY-ID">  
  <owner>OWNER-NAME</owner>  
  <address>STRUCTURED-PROPERTY-ADDRESS</address>  
</property>
```

...

- Use direct constructors and nested queries

Exercise 3



- Express the following XQuery query
 - Create a sequence of all agency employees

```
<employee>
  <name>CONCATENATED-EMPLOYEE-NAME</name>
  works at
  <agency id="AGENCY-ID">AGENCY-NAME</agency>
</employee>
...
```

- Use computed constructors

Exercise 4



- Express the following XQuery query
 - Create one flat sequence with property identifiers, flat names and comfort levels of all available flats

```
<property>PROPERTY-ID</property>  
<name>FLAT-NAME</name>  
<comfort>COMFORT-LEVEL</comfort>
```

...

- Order this sequence using
 - property names in a descending order, and
 - then using flat names in an ascending order

Exercise 5

- Express the following XQuery query
 - Select identifiers of all flats with comfort levels *B* or *C* such that their *rate* value is below the overall average

Exercise 6



- Express the following XQuery query
 - Create an XHTML fragment with a table of properties

```
<table>
  <tr>
    <th>Id</th><th>Name</th><th>Features</th>
  </tr>
  ...
  <tr>
    <td>PROPERTY-ID</td>
    <td>PROPERTY-NAME</td>
    <td>COMMA-SEPARATED-LIST-OF-FEATURES</td>
  </tr>
  ...
</table>
```

Exercise 7



- Express the following XQuery query
 - Create a sequence of luxury and common properties according to the following pattern
 - Property is considered to be luxury if ratings of all its top-level flats are greater than *10000*

```
<luxuryProperty name="PROPERTY-NAME" />
```
 - Otherwise it is just a common property

```
<commonProperty name="PROPERTY-NAME" />
```
 - Preserve the original order of properties
 - Use quantifiers
 - Once finished, also try to find a solution without quantifiers

Exercise 8



- Express the following XQuery query
 - Aggregate flats according to their comfort levels

```
<group comfort="COMFORT-LEVEL">
  <flat id="FLAT-ID" />
  ...
</group>
...
```

- Order the created comfort groups alphabetically
- Include only comfort groups with 2 or more flats
- Order flats inside these groups using their ids

Exercise 9



- Express the following XQuery query
 - Select properties having the maximal number of advertised flats

```
<property id="PROPERTY-ID" flats="FLATS-COUNT" />
```

...

- Order the result using property names

Exercise 10

- Express the following XQuery query
 - Join data about real estate agencies
 - Use *data.xml* and *data-agencies.xml*

```
<agency id="AGENCY-ID">
  <contact type="email">EMAIL-ADDRESS</contact>
  <contact type="web">WEB-PAGE</contact>
  <contact type="phone">PHONE-NUMBER</contact>
</agency>
...
```

Exercise 11



- Express the following XQuery query
 - Integrate data about flats from two differently structured sources
 - Use *data.xml* and *data-flats.xml*

```
<flats>
  <flat
    id="FLAT-ID"
    name="FLAT-NAME"
    property="PROPERTY-ID"
    comfort="COMFORT-LEVEL" />
  ...
</flats>
```

- Order individual flats using their identifiers