

NPRG036, A7B36XML: XML Technologies

Practical Class 4:

# XML Schema: Exercises

14. 4. 2016

**Jiří Helmich**

**Martin Svoboda**

<http://www.ksi.mff.cuni.cz/~svoboda/courses/2015-2-NPRG036/>

<http://www.ksi.mff.cuni.cz/~svoboda/courses/2015-2-A7B36XML/>

# Exercise 1

- Create an XSD for a real estate XML document
  - Its root element *real-estate* contains a sequence of sub-elements *agencies*, *owners*, *properties* and *flats*, all with an empty content
  - Do not define any global complex type
  - Do not declare a target namespace
- Modify the XML document such that its validity can be checked with respect to this XSD

# Exercise 2



- Extend the previous XSD schema for 02.xml
  - Elements *agencies*, *owners*, *properties* and *flats* can now contain particular *agency*, *owner*, *property* and *flat* elements respectively
    - Agency = name, e-mail, phone
    - Owner = name
    - Property = name, number of flats
    - Flat = name, description, rate
  - Define global complex types for all the 4 main entities
    - Do not define global complex types for anything else
  - Use ordinary *xs:string* simple type for all text values

# Exercise 3

- Extend the previous XSD schema for 03.xml
  - Add the following compulsory attributes
    - Identifiers of owners, properties and flats
    - References from flats to properties they belong to
    - Comfort levels of flats
  - Add the following optional attributes
    - References from properties to owners they belong to
    - Dates when flats were posted
  - Use *xs:string* simple type once again for all values

# Exercise 4



- Extend the previous XSD schema for 04.xml
  - Choose appropriate predefined simple types for all text and attribute values and/or introduce your own derived global/anonymous simple types
    - Use DTD-based identifiers for owners, properties and flats as well as their mutual relationships
    - Allow only *A, B, C, D, E, F* values for comfort levels of flats
    - Rates of flats are positive integers or *0*
    - Flat identifiers must match regular expression `[fF][0-9]{1,5}`
    - Numbers of property flats should be equal to at most *1500*
    - Flat names must have at least *5* and at most *100* characters

# Exercise 5



- Extend the previous XSD schema for 05.xml
  - Add *features* to properties and flats
  - Add an optional attribute to each *feature* describing a list of categories to which it is associated
    - Only a fixed list of predefined categories is permitted (*location, equipment, security, comfort*)
    - Do not allow its further derivations using restriction
  - *Features* element might be missing, otherwise 2 to 5 particular *feature* elements are expected

# Exercise 6



- Extend the previous XSD schema for 06.xml
  - Define global *address* element
    - Address = street, postal code, city, and country
    - Country is optional and when missing, postal code must be placed at the end (i.e. after city and not before)
  - Add these addresses to agencies, owners, and properties (one address to each entity)
    - Use references for this purpose
  - All agency data (name, e-mail, phone and address) can now occur in arbitrary order

# Exercise 7

- Extend the previous XSD schema for 07.xml
  - Modify flat *description*
    - It should now contain a mixed content with references to owners and properties using their IDs
      - E.g. ... <ownerRef ownerId="o1">...</ownerRef> ...
    - There can be an arbitrary number of these references, but owner references must occur before property references
  - Add optional property *description*
    - It should contain any well-formed and valid fragment of an XHTML web page
      - <http://www.w3.org/1999/xhtml>



# Exercise 8



- Extend the previous XSD schema for 08.xml
  - Add the following keys
    - Identify properties using their *idProperty* attributes
    - Identify flats using their *idFlat* attributes
    - Identify flats also using pairs of their names and properties they belong to, i.e. using *name* and *propertyRef*
  - Add the following foreign key
    - References from flats to properties they belong to

# Exercise 9

- Extend the previous XSD schema for 09.xml
  - Use assertions to verify the following integrity constraints
    - Each owner has a *name* and this name is immediately followed by *address* of residency

# Exercise 10



- Extend the previous XSD schema
  - Introduce two new specific types of properties
    - Skyscraper
      - Extends *property* by adding a new attribute *numberOfFloors*
    - Cottage
      - Restricts *property* by removing *features* and *ownerRef*

# Exercise 11

- Modify XML document 11.xml such that it becomes valid with respect to the previous XSD
  - i.e. such that elements *property* in *properties* are allowed to be instances of the newly introduced specific types for skyscrapers or cottages
  - Do not change the XSD

# Exercise 12



- Modify the previous XSD schema for 12.xml
  - Use explicit substitutability instead of implicit for specific types of properties
    - I.e. use substitution groups