1 jmenné prostory

```
<pricelist:offer
        xmlns:pricelist="http://www.eprice.cz/e-pricelist">
        <pricelist:item tax="22%">
                <pricelist:name>
                        <bib:book
                                xmlns:bib="http://www.my.org/bib">
                                <bib:author>Mark Logue</bib:author>
                                <bib:name>The King's Speech</bib:name>
                        </bib:book>
                </pricelist:name>
                <pricelist:price curr="CZK">259</pricelist:price>
        </pricelist:item>
</pricelist:offer>
```

2 Xml schéma a varianty DTD. U každého DTD říct, zda bere všechny varianty XML dokumentů, které projdou XML schématem.

XML: varianty

- XHTML
- MathML
- SVG - scalable vector Graphics
- X3D - extensible 3D
- XForms
- DocBook - Creation of Technical Documentation

3 Slovně popsaný xml dokument a k němu udělat XML schéma. (Chtěl, aby tam bylo libovolné pořadí 2 elementů)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" ...>
  <xs:element name="Inicialy" type="typ"/>
  <xs:complexType name="typ">
    <xs:choice>
      <xs:sequence>
        <xs:element name="jmeno" type="xs:string"/>
        <xs:element name="prijmeni" type="xs:string"/>
      </xs:sequence>
    <xs:sequence>
```

```
        <xs:element name="prijmeni" type="xs:string"/>
        <xs:element name="jmeno" type="xs:string"/>
      </xs:sequence>
    </xs:choice>
  </xs:complexType>
</xs:schema>
```

4

5 xs:assert

=Invariant
assert – error, when the expression does not return true
Using XPath

```
<xs:complexType name="Interval">
        <xs:attribute name="min" type="xs:integer"/>
        <xs:attribute name="max" type="xs:integer"/>
        <xs:assert test="@min␣<␣@max"/>
</xs:complexType>
```

6 příklad na SAX. Je to push nebo pull parser?

je to push (asi)

Streaming **pull** parsing refers to a programming model in which a client application calls methods on an XML parsing library when it needs to interact with an XML infoset; that is, the client only gets (pulls) XML data when it explicitly asks for it.

Streaming **push** parsing refers to a programming model in which an XML parser sends (pushes) XML data to the client as the parser encounters elements in an XML infoset; that is, the parser sends the data whether or not the client is ready to use it at that time.

7 příklady na xpath.

8 Vybrat z různých xpath správné.

9 Vypsat dopředné a rekurzivní osy v xpath. Jaký je mezi nimi rozdíl?

Axis:

- self
- parent
- ancestor

- ancestor-or-self
- child
- descendant
- descendant-or-self
- preceding-sibling
- preceding
- following-sibling
- following
- attribute
- namespace

dopředná osa (forward axis) – obsahuje pouze kontextový uzel a uzly, které následují za kontextovým uzlem v pořadí toku dokumentu

reverzní osa (reverse axis) – obsahuje pouze kontextový uzel a uzly, které předcházejí kontextovému uzlu v pořadí toku dokumentu
zdroj: https://is.muni.cz/th/mnnx5/thesis.pdf str.14

10 xslt. Vytvořit tabulku pro xml dokument.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1
    <xsl:output method="html" version="4.0" encoding="UTF-8" indent="yes" d
  <xsl:template match="/">
    <html>
      <body>
        <table>
          <xsl:apply-templates select="real-estate/properties/property"/>
        </table>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="properties/property">
        <tr>
            <td>
                <xsl:apply-templates select="@idProperty"/>
            </td>
        </tr>
  </xsl:template>
</xsl:stylesheet>
```

11 popsat algoritmus pro výběr šablony pro daný uzel v xslt. (Jak funguje apply templates?)

We can always apply only one template
We take the one with the highest priority

If it is not set, the priority is evaluated implicitly as follows:

- 0.5: path with more than one step
- 0: element/attribute name
- -0.25: *
- -0.5: node(), text(), ...

there is always a template to be applied (pre-defined, default)

12 to samé jako 10 jen pro xquery. Vysvětlit proč to nejde pokud to nejde.

```
<table>
  <tr>
    <th>Id</th>
    <th>Name</th>
    <th>Features</th>
  </tr>
  {
    for $p in fn:doc('data.xml')//property
    return

  <tr>
    <td>{ fn:data($p/@idProperty) }</td>
    <td>{ $p/name/text() }</td>
    <td>
          { $p/features/feature[1]/text() }
          {
            for $t in $p/features/feature[position() != 1]
            return fn:concat(", ", $t/text())
          }
        </td>
  </tr>
  }

</table>
```

13 Dietz numbering. Ukázat jak v něm poznám vztah předek-dědic.

Preorder traversal
Child nodes of a node follow their parent node

Postorder traversal
Parent node follows its child nodes

Nechť $L(v) = (x, y) a L(u) = (x', y')$, pak:
v je potomek $u <=> x' < x \& y' > y$

14 Co je xquery core? Jaký má vztah k xquery?

XQuery Core defines a syntactic subset of
XQuery with the same expressive power as
XQuery, but without duplicities

XQuery Core is useful mainly from the theoretical point of view.
The queries are long and complex

15 ?

16 atributové mapování

Generic-tree Mapping
Attribute = name of the edge

Edge *attribute* (sourceID, order, type, targetID)

17 ?

18 něco na generické mapování

- Edge mapping

  Edge (sourceID, order, label, type, targetID)

- Attribute mapping

  Edge *attribute* (sourceID, order, type, targetID)

- Universal mapping

  Uni (sourceID, ordera1, typea1, targetIDa1, ... orderak, typeak, targetIDak)

- Normalized universal mapping

  The universal table contains for each name just one record Others (i.e. multi-value attributes) are stored in *overflow tables*

19 SQL/XML dotaz - příklad

Extension of SQL which enables to work with XML data

SQL expressions -> XML values

- XMLELEMENT – creating XML elements
- XMLATTRIBUTES – creating XML attributes
- XMLFOREST – creating XML elements for particular tuples
- XMLCONCAT – from a list of expressions creates a single XML value
- XMLAGG – XML aggregation

Příklad:

```
SELECT E.id,
       XMLELEMENT (NAME "emp",
                   XMLATTRIBUTES (E.id AS "empid"),
                                  E.first || ' ' || E.surname) AS xvalue
FROM Employees E WHERE ...
```

Dlaší ukázka

```
SELECT COUNT (*) FROM Children D WHERE D.parent = E.id
```

20 ?