
3. JAXP wrap up

XML Namespaces

PB138

Homework

- Homework can be found here
</el/1433/jaro2013/PB138/ulohy/1/>
 - Deadline is 11.3.2013 23:59
 - You can be awarded -3 points (fake)
 - Late and corrected 0
 - You can be awarded +3 for correct solution
 - Use only JAXP, please study any method you are going to use in JavaDoc
-

Get Tiger Woods Tweets

- Get tweets-woods-big.xml
- Find the text of the 10 most retweeted tweets

Adding elements

- create method `addSalary(personId, int salary)` which adds `<salary>50000</salary>`
 - use `Document.createElement`
 - use `setTextContent`
 - use `Element.appendChild`
 - create another method `increaseSalary(personId, amount)` which will increase salary
 - use `Integer.parseInt`
-

Modify Tiger Woods tweets

- Create new xml document with elements

```
<tweets>
```

```
  <tweet retweetcount="XYZ">
```

```
    Text of the tweet.
```

```
  </tweet>
```

```
</tweets>
```

use saveToFile

- After saving you can see the DOCTYPE is lost
 - `transformer.setOutputProperty(OutputKeys.DOCTYPE_SYSTEM, document.getDoctype().getSystemId());`
-

Revising JAXP

- Create method in JAXP that will recursively list all the nodes of social_network.xml document
 - print getNodeName()
 - print human readable getNodeTypes()
 - use switch statement
-

XML Namespace

- element might be in namespace

```
<person xmlns="http://muj.namespace">....
```

- xmlns puts the element and all its descendant elements into the namespace
 - Create XML with 2 people. All the elements in the document will be in namespace "my.own.namespace".
-

XML Namespace

- We may use **prefix** to have fine control
- Prefix is usable in **the same element** or its descendants

```
<person xmlns="http://muj.namespace" xmlns:meno="
http://muj.jmenny.namespace">
  <meno:name>Filip</name>
</person>
```

- Add attribute **firstname** to elements and put them into namespace **"jmena"**
 - Add subelements **surname** into namespace **"prijmeni"**
-

XML Namespaces

- put <social_network> and all its subelements into namespace "social.seminar.fi.muni.cz"
 - put Steven Segal person and all his subelements into "people.type.hollywood" namespace
 - put all the <phones> and <phone> elements into "contact.phones" namespace
-

XML Namespaces

- Find out in which namespace is attribute `phone/@number` of Steven Segal.
 - Tip: `.getAttributes().item(x).getNamespaceURI()`
- Find out in which namespace is some `@xmlns` attribute

XPath

- <http://www.xpathtester.com/>
 - put our social_network.xml there without namespaces!
 - check output to new window
 - XPath version 1.0, 2.0
 - //person
 - descendant-or-self:person
 - /descendant-or-self:person
-

Basic XPath syntax

- Path to a set of elements: **step/step/step/...**
 - basic step is "**axisname::nodetest**"
 - Processing starts at invisible context node "Document" step by step
 - **child::social_network/child::people/child::person**
-

XPath evaluation sequence

- During evaluation, there is always one *context node*. At the very beginning, the context node is invisible "root of the document"
 - '/' at the beginning will ensure, evaluation starts with *context node* as a root of the document
 - child::social_network/child::people/child::person
 - 1. from child axis of *context node*, select all social_network elements
 - 2. for each result from 1.: set as context node, from child axis select elements called "people"
 - 3. for each result from 2.: set as context node, from child axis select elements called "person"
-

DOM4J

- <http://dom4j.sourceforge.net/dom4j-1.6.1/apidocs/index.html>
 - Use Dom4JParser and implement printDocumentElement()
 - getRootElement()
 - Use Dom4JParser and implement printPeopleNames()
 - getRootElement() to get the element
 - then use methods **element()** and **elements()**
-

Basic XPath usage

- The above says "from children, select social_network. Take result and from each child of each result element take person elements....."
 - Shortcut:**social_network/people/person**
 - Try getting all the phone elements using axis "child"
 - Try getting <phone> elements using **descendant** axis
 - Try getting people names using **attribute** axis
-

Homework

- Learn all **axis**
 - Learn all **nodetest**
 - Play with XPath
-

Next week

- XPath in depth
-