

Zpracování XML dokumentů pomocí DOM a SAX

Podklady pro cvičení k předmětu Technologie XML

Irena Mlýnková
Martin Nečaský

SAX

- Událostmi řízený model
 - **Událost**
 - začátek/konec dokumentu
 - začátek/konec elementu
 - znaková data
 - ...
 - Parser vyvolává sekvenci událostí podle pořadí v dokumentu
 - Nelze přejít na libovolné místo ani zpět
-

SAX

☐ začátek/konec dokumentu

- `startDocument()`
- `endDocument()`

☐ začátek/konec elementu

- `startElement(String uri, String localName, String qName, Attributes atts)`
- `endElement(String uri, String localName, String qName)`

☐ znaková data

- `characters(char[] ch, int start, int length)`
-

Kompilace a spuštění

□ Kompilace(pro PříkladN):

javac -encoding utf-8 -d build/classes src/y36xml/PříkladN.java

□ Spuštění (pro PříkladN):

java -classpath build/classes y36xml/PříkladN

DOM

- ❑ Objektový model
 - ❑ Každá logická část XML dokumentu je reprezentována jako objekt
 - Element, atribut, text, ...
 - ❑ Objekty tvoří strom
 - ❑ Celý XML dokument typicky načten celý ve vnitřní paměti
-

DOM

□ Interface **Node**

- Reprezentuje obecný uzel v XML dokumentu
 - Specializováno do interfaců reprezentujících konkrétní typy uzlů jako např. dokument, element, atribut, ...
 - **getAttributes()**
 - **getChildNodes()**
 - **getFirstChild()**
 - **getNodeName()**
 - **appendChild()**
 - **removeChild(Node oldChild)**
-

DOM

□ Interface **Document**

- Reprezentuje celý dokumentu
 - **getDocumentElement()**
 - **getElementById(String elementId)**
 - **getElementsByTagName(String tagName)**
 - **createAttribute(String name)**
 - **createElement(String tagName)**
 - ...
-

DOM

- ❑ Další interfaces **Element, Attr, Text, ...**
 - ❑ Google: ***"java dom api"***
-

Kompilace a spuštění

☐ Kompilace(pro PříkladN):

`javac -encoding utf-8 -d build/classes src/y36xml/PříkladN.java`

☐ Spuštění (pro PříkladN):

`java -classpath build/classes y36xml/PříkladN`
