

Luento 7: XML-ohjelmointirajapinnat

AS-0.110 XML-kuvauskielten perusteet

Janne Kalliola

1

XML-ohjelmointirajapinnat

- Document Object Model (DOM)
 - käyttö
 - rakenne ja versiot
 - perusraajapinnat
 - rajapinnat tarkemmin
- Simple API for XML (SAX)
 - tapahtumapohjainen käsittely
 - tapahtumat
 - SAX:n toiminta
 - SAX:n rajapinnat ja luokat

2

Perusrajapinnat

- DOM ja SAX ovat perusrajapintoja
 - tarjoavat alhaisen tason käsittelymallit XML-dokumenteille
 - eivät ota kantaa XML-dokumenttien sisältöön millään tavalla
 - siis yleiskäyttöisiä
 - käytännössä kaikki muut XML-rajapinnat rakentuvat näiden päälle
- On olemassa suuri joukko erikoistuneita rajapintoja
 - osaa näistä käsitellään seuraavalla luennolla

DOM

W3C DOM-malli

- Document Object Model
- W3C:n suositus dokumenttien esittämiseksi ohjelmissa
 - alusta- ja kieliriippumaton
 - mahdollistaa dokumentin muokkauksen
 - sisältö
 - rakenne
 - tyyli

DOM:n käyttö

- DOM:sta löytyy rajapinnat
 - IDL:lle
 - Javalle
 - EcmaScriptille
- Rajapintojen käyttämiseksi täytyy ottaa käyttöön joku DOM-toteutus
 - useita toteutuksia saatavana ilmaiseksi eri kielille
- Lisätietoa aiheesta
 - <http://www.w3.org/DOM/>

DOM:n rakenne

- DOM määrittelee useita erilaisia rajapintoja
 - DOM Core – rakenteisten dokumenttien puumainen käsittely
 - DOM XML – XML-dokumenttien käsittely
 - DOM HTML – HTML-dokumenttien käsittely
 - DOM Events – XML-dokumenttien tapahtumapohjainen käsittely
 - DOM Validation – validointituki
 - DOM XPath – XPath-lausekkeiden käyttö DOM-puiden käsittelyssä

7

DOM:n versiot

- DOM:n ominaisuudet ovat kasvaneet uusien versioiden myötä
 - DOM 1 (10/1998) – Core + XML + HTML
 - DOM 2 (11/2000) – DOM 1 + CSS-tuki + tapahtumamalli + nimiavaruudet
 - DOM 3 (4/2004) – DOM 2 + XML Base + validointi + dokumenttien lataus ja tallennus
- Kurssin tarpeisiin riittää DOM 1
 - DOM 2:n tapahtumamalli myös näppärä

8

DOM muille kielille

- DOM:sta on olemassa erilliset sidonnat (bindings) myös seuraaville XML-pohjaisille kielille:
 - MathML 2.0
 - SMIL
 - SVG 1.0
- Sidonnassa on määritelty kielikohtaisia rajapintoja
 - rajapintojen avulla voidaan käsitellä suoraan XML-kielen tarjoamia rakenteita
 - ei tarvitse tehdä kaikkia operaatioita suoraan elementtitasolla

DOM 1:n perusrajapinnat (1/4)

- Document
 - kuvaa yhtä dokumenttia
 - dokumentti voi olla esimerkiksi XML:ää tai HTML:ää
 - sisältää tehdasmetodit dokumentin osien luomiseksi
 - dokumentin osat eivät voi esiintyä dokumentin ulkopuolella
- DocumentType
 - DTD
 - ei sisällä juurikaan mahdollisuuksia käsitellä DTD:n sisältöä

DOM 1:n perusrajapinnat (2/4)

- Node
 - dokumenttipuun solmu tai lehti
 - sisältää metodit lapsien lisäämiseksi
 - kaikki DOM:n dokumenttirajapinnat ovat myös Nodeja
- NodeList
 - järjestetty lista Nodeista

11

DOM 1:n perusrajapinnat (3/4)

- Element
 - vastaa XML:n elementtiä (tagi)
- Attr
 - elementin attribuutti
 - avain-arvopari
- Text
 - tekstisirpale
- CDATASection

12

DOM 1:n perusrajapinnat (4/4)

- Entity
 - korvausmerkintä jollekin merkille tai merkkijonolle
- Comment
 - kommentti dokumentissa

Rajapintojen toteutus

- Kuten todettua, DOM ei sisällä toteutusta rajapinnoille
- Toteutus hankittava muualta
 - useita toteutuksia tarjolla
 - Javalle esimerkiksi Apachen Xerces tai Sunin JAXP
- Toteutus erotettu rajapinnoista tehdasmetodeilla
 - kaikki puun solmut tuotetaan käyttäen rajapinnan Document metodeja
- Tämän lisäksi kielen kirjastoissa saattaa olla metodit DOM-jäsentimen luomiseksi
 - lopullinen DOM-toteutus ei välttämättä ole siis kehittäjän tiedossa tai se voi vaihtua ajoalustasta toiseen

Node (1/2)

- Kaikkien DOM-rajapintojen kantarajapinta
- Metodit puussa liikkumiseen
 - parentNode()
 - firstChild()
 - lastChild()
 - nextSibling()
 - previousSibling()
- Metodit puun muokkaamiseen
 - appendChild(Node)
 - removeChild(Node)
 - insertBefore(Node, Node)
 - replaceChild(Node, Node)

15

Node (2/2)

- Metodit solmun tyypin ja nimen käsittelyyn
 - nodeName()
 - nodeType()
 - nodeValue()
 - attribuutille palauttaa attribuutin arvon
 - tekstile, CDATA-alueelle, kommentille ja käsittelyohjeelle palauttaa niiden sisällön
 - muille palautetaan null
 - setNodeValue(String)

16

Document

- Document-puun juurielementti
 - lapsina käsittelyohjeita, DTD ja itse dokumentin juurielementti
 - sisältää metodit muiden solmujen luomiseen
 - createAttribute(String)
 - createElement(String)
 - createTextNode(String)
 - metodit dokumentin tulkintaan
 - getDocType()
 - getDocumentElement()

17

DocumentType

- Kuvaa DTD:ta
- DOM 1 ei salli DTD:n muokkaamista
- Sisältää metodit entiteettien ja notaatioiden lukemiseen
 - getEntities()
 - getNotations()

18

Element

- Yksittäinen elementti
- Metodit attribuuttien lisäämiseen, poistamiseen ja niiden arvojen käsittelyyn
 - `setAttribute(String, String)`
 - `getAttribute(String)`
- Attribuutit talletetaan omiksi solmuiksi
 - Elementin kautta käsittely onnistuu kuitenkin merkkijonopohjaisesti
 - yleensä helpompaa

19

Attr

- Attribuutti
- Liittyy aina yhteen elementtiin
- Attribuutit eivät ole DOM:ssa Elementin lapsia
 - pidetään tallessa erillisessä tietorakenteessa
 - Attr:n puutoiminnallisuudet eivät toimi
- Metodit arvon kysymiseen ja asettamiseen
 - `getValue()`
 - `setValue(String)`

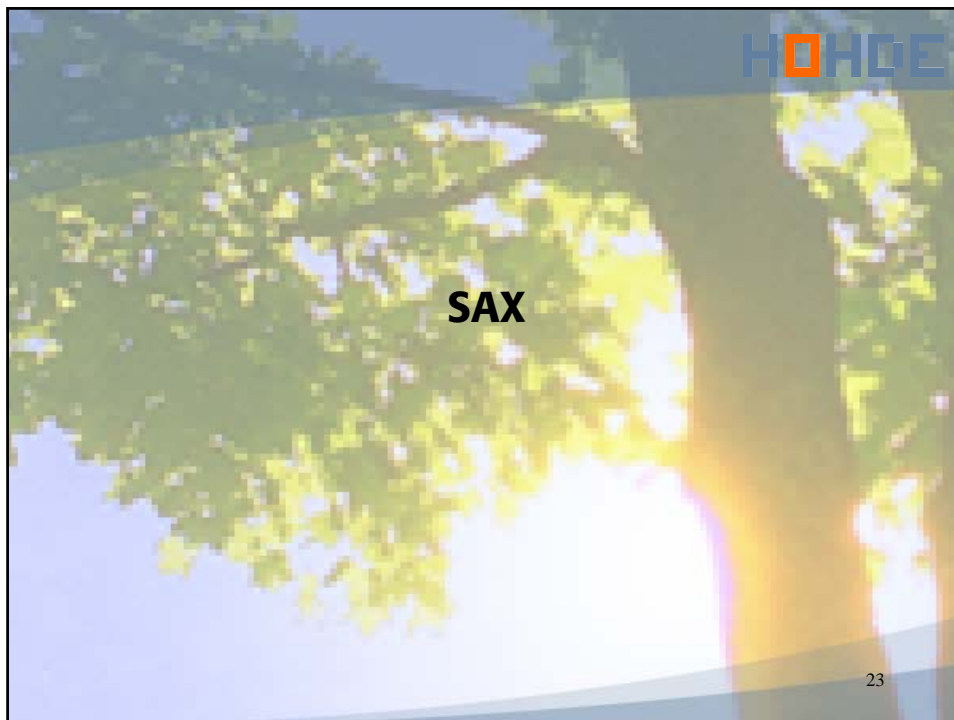
20

Tekstirajapinnat

- Kaksi tekstirajapintaa
 - CDATASection
 - Text
- Molemmat perivät rajapinnan CharacterData
 - sisältää metodit sisällön käsittelyyn
 - setData(String)
 - getData()
 - replaceData(int, int, String)
 - deleteData(int, String)
 - appendData(String)

DOM-puun luonti ja käyttö

- Yleensä DOM-puu luodaan pyytämällä Document-rajapinnan toteuttava olio
 - loput puusta luodaan Document-rajapinnan tehdasmetodeilla
- DOM-puu saattaa tulla myös valmiiden metodien paluuarvona
 - esimerkiksi XML-dokumentin lukeminen tiedostosta
- DOM-puuta voidaan muokata ja siinä olevaa tietoa voidaan poimia rajapinnan metodeilla
 - DOM sisältää myös metodeja puun läpikäyntiin



• Tapahtumapohjainen käsittely

- Dokumenttipuu on helppo, mutta välillä tehoton tapa tulkita dokumentti
 - aina puuta ei ole tarkoituksenmukaista rakentaa
 - ohjelma voi rakentaa oman puunsa tai muun tietorakenteen, jolloin muunnos rakenteesta toiseen on turhan raskas

25

• Tapahtumat

- XML-dokumentti luetaan jostakin lähteestä
- Jokainen XML:n elementti aiheuttaa tapahtuman
 - XML tulkitseva ohjelma saa dokumentin rakenteen tapahtumien kautta
 - erillistä puuta ei rakenneta

26

Tapahtumaesimerkki

- Dokumentti

```
<document>
  <hello>Hello, World!</hello>
</document>
```

- Vastaavat tapahtumat

```
start document
start element: document
start element: hello
characters: Hello, World!
end element: hello
end element: document
end document
```

27

SAX:n rakenne

- Jäsenin (XMLReader)

- käsittelee XML-dokumentin
- luo tarvittavat tapahtumat

- Sovellus

- toteuttaa tapahtumien käsittelijät
- vastaanottaa ja käsittelee tapahtumat

28

Toiminta

1. Ohjelma haluaa lukea XML-dokumentin
2. Ohjelma antaa XML-dokumentin virtana parserille
3. Parseri lukee dokumentin
 - jokaisesta elementistä lähetetään tieto ohjelmalle
 - ohjelma käyttää tietoa hyväkseen tai hylkää sen
4. Parseri lopettaa suorituksensa
5. Ohjelmalla on tarvitsemansa tiedot dokumentista

SAX-rajapinnat

- SAX pohjautuu melkein kokonaan rajapintoihin
 - mukana myös muutama apuluokka
- Rajapinnat ja tärkeimmät luokat pakkauksessa `org.xml.sax`
- Apuluokat pakkauksessa `org.xml.sax.helpers`

SAX-jäsentimen toteuttamat rajapinnat

- Ohjelma luo SAX-jäsentimen ja ohjaa/kyselee sen toimintaa näiden rajapintojen kautta
 - XMLReader
 - jäsentimen toiminnallisuuden määrittely
 - dokumentin sisäänluku
 - Attributes
 - elementtien attribuutit listassa
 - Locator
 - tapahtuman paikka dokumentissa

31

Ohjelman rajapinnat

- Ohjelma toteuttaa osan seuraavista rajapinnoista saadakseen tapahtumia
 - ContentHandler
 - ottaa vastaan dokumenttitapahtumat
 - dokumentin alkaminen ja loppuminen
 - elementtien alkaminen ja loppuminen
 - merkkijonot elementtien välissä
 - ErrorHandler
 - virheellisten dokumenttien käsittely
 - EntityResolver
 - Entiteettien käsittely
 - DTDHandler
 - DTD:n käsittely

32

SAX:n luokat

- InputSource
 - dokumentin lukuun liittyviä rutiineja
- SAXException
 - poikkeusten käsittelyyn
- SAXParseException
 - dokumenttien käsittelyyn liittyvät poikkeukset
- HandlerBase
 - perustoiminnallisuudet parserin rajapinnoille
 - voidaan käyttää toteuttamattomien osien tilalla

33

SAX-toteutus

- SAX-rajapinnat on toteutettu yleensä DOM-jäsentimiin
 - DOM-jäsentimin käyttää sisäisesti SAX:ia dokumentin lukemiseen merkkijonosta puumuotoon
- SAX-jäsentimen saa vastaavalla tavalla kuin DOM-jäsentimenkin
 - periaatteessa sama luokka voi toteuttaa molemmat rajapinnat
 - yleensä toteutus on kuitenkin erillinen

34

Kysymyksiä? Kommentteja?