

Tehtävä 2: XPATH kyselykieli

Tämä tehtävä koostuu kahdesta osatehtävästä:

- Osassa 1 harjoitellaan XPath lausekkeiden kirjoittamista käyttäen online XPath tester sovellusta.
- Osassa 2 kirjoitetaan XPath lausekkeita annettuun XSLT-dokumenttiin ja tutustutaan alustavasti XSLT muunnosten suorittamiseen käyttäen online XSLT testeriä, Notepad++ editorin XML pluginia sekä Web-selainta. XSLT-kieleen tutustutaan tarkemmin vasta tehtävässä 3.

Osatehtävä 1: XPATH lausekkeiden harjoittelu online sovelluksella

Tutki tehtävän kysymyksiin liittyviä XML lähtötiedostoja (2), ja kirjoita oikeat xpath-lauseet, jotka palauttavat vastaukset esitettyihin kysymyksiin 1-20. Käytä online testeriä

<http://www.xpathtester.com/xpath> ratkaisujesi testaukseen ja talleta löytämäsi oikeat ratkaisut tekstitiedostoon *Tehtävä2_xpath_ratkaisut.txt* niiden myöhempää demoamista varten.

XPath-lausekkeiden haku kohdistetaan ko. kysymysjoukon lähtödokumenttiin. Kopioi sen sisältö XPath testerin XML-kenttään ja kirjoita XPath lauseke XPath-kenttää. Lauseke suoritetaan painamalla *Test*-painiketta.

A: lähtötiedosto *plant_catalog.xml* (.XML)

B: lähtötiedosto *ancient_wonders.xml* (.XML),

C: lähtötiedosto *ancient_wonders.xml* (.XML),

1.1 Kysymyksiä vastaavat XPath lausekkeet

A: Kysymykset 1-10

Kirjoita oikea XPath lauseke, joka poimii tiedostosta *plant_catalog.xml* oikeat solmut.

- Kysymys 1: Dokumentin juuren
- Kysymys 2: Juurielementin (CATALOG)
- Kysymys 3: Kaikki juurielementin lapsi-elementit (siis ei lapsenlapsia)
- Kysymys 4: Kaikki juurielementin lapsenlapsi-elementit
- Kysymys 5: Kolmannen Plant:n Common-elementin tekstisisältö
- Kysymys 6: Viimeisen kasvin kasvitieteellinen nimi (Botanical)
- Kysymys 7: Kaikkien niiden kasvien Zone-elementit, joiden hinta $Price \leq 3.20$
- Kysymys 8: Kasvit, joiden hinta yli 6.60 ja jotka viihtyvät vain auringossa (Sun)
- Kysymys 9: Yksivuotisten (Annual) kasvien Zone-elementtejä edeltävät sisarelementit
- Kysymys 10: Yksivuotisten (Annual) kasvien Zone-elementtien kaikki sisarelementit

B: Kysymykset 11-14

Lähtötiedosto: *ancient_wonders.xml*

Tämä tehtävä on suoraa jatkoa A kysymyksille. Tutki nyt toista tehtävätiedostoa *ancient_wonders.xml* (.XML), ja pyri sitten muodostamaan oikeat xpath-lauseet alla olevien kysymysten ratkaisemiseksi. Kirjoita XPath lauseke, joka poimii tiedostosta *ancient_wonders.xml* oikeat solmut:

- Kysymys 11: kaikkien kuvaelementtien (*main_image*) kaikkien attribuuttien valinta
- Kysymys 12: kaikkien ihmeiden (*wonder*) nimi-elementit, jotka ovat kreikaksi
- Kysymys 13: kaikkien niiden ihmeiden *wonder*-elementit, jotka on tuhottu (*year_destroyed*) ennen ajanlaskua (BC)
- Kysymys 14: kaikkien niiden ihmeiden nimielementit, jotka ovat tuhoutuneet vuosien 1300AD ja 1500AD välillä

C: Kysymykset 15-20 (samat tiedostot, mutta käytetään xpath-funktioita tai operaattoreita)

- Kysymys 15: Tietoja yhdistävä XPath lauseke, joka palauttaa ensimmäisen ihmeen nimen ja korkeustiedon lausemuodossa: "'name' is 'height' 'units' high"
- Kysymys 16: Laske kaikkien "source"-elementtien attribuuttien lukumäärä
- Kysymys 17: Laske kaikkien niiden ihmeiden yhteiskorkeus (summa) jaloissa (feet), joiden pituus on ilmaistu jalossa (feet)
- Kysymys 18: Sisältääkö neljännen ihmeen nimi sanan 'Lighthouse'?
- Kysymys 19: Laske viimeisen kuvan (*main_image*) pikselien määrä
- Kysymys 20: Missä kaupungissa (vain kaupunki, ei valtiota) sijaitsee 'Mausoleum at Halicarnassus' niminen maailman ihme?

1.2 Vinkit:

- Tässä tehtävässä kaikkien lausekkeiden tulee alkaa joko / tai // merkeillä tai suoraan jollain funktion nimellä, sillä lauseke kohdistuu koko XML dokumentin puuhun.
- Online testerin XPath- kentässä voidaan käyttää vertailuoperaattoreina "<" ja ">" merkkejä
- Jos Xpath lauseke on osana XML dokumentia (esim XSLT:ssä) tulee "<" ja ">" merkkien tilalla käyttää < ja > -entiteettejä, jotka ovat ns. entiteettimerkkejä, joita käytetään korvaamaan XML:ssä syntaksiin kuuluvia merkkejä
- Kun lainausmerkkien välissä olevassa XPath-lausekkeessa tarvitsee esittää jokin arvo lainausmerkeissä, tulee siinä käyttää yksinkertaisia lainausmerkkejä. Esimerkiksi seuraavassa predikaatin ehdossa: `<xsl:for-each select="/CATALOG/Plant[Zone='Annual']/Zone">`
- Funktio *position()* palauttaa nykyisen solmun järjestysnumeron nykyisessä solmujoukossa
- Outo virheilmoitus koskien ensimmäistä riviä. Syy: XML-deklaraation lainausmerkit kopioituvat virheellisesti online työkalun XML-kenttään. Ratkaisu: kirjoita lainausmerkit uudelleen tai poista koko rivi.

Osatehtävä 2: XPATH lausekkeet XSLT dokumentissa

Edellisissä osatehtävissä XPath polku aloitettiin aina XML-dokumentin puun juuresta käyttäen joko / tai // merkkejä. Tässä tehtävässä XPath-lausekkeita hyödynnetään XSL muunnoksissa valitsemaan lähtöjoukon puurakenteesta tietty joukko solmuja, johon kohdistetaan juuri tälle solmujoukolle määrättyjä muunnosoperaatioita (kopiointi, arvojen valinta, muunnokset arvoihin jne.). Kuhunkin solmujoukkoon kohdistettavat operaatiot tai muunnossäännöt on ryhmitelty omiin *template*:ihinsä, jotka niitä kutsuttaessa saavat valitun (*select – match*) solmujoukon käsiteltäväkseen. Template voi

edelleen valita saamastaan lähtöjoukosta jonkin osajoukon solmuja ja kutsua niiden käsittelyyn tarkoitettua toista (ali-) *template*:a. Koska eri *template*:iden lähtöjoukot ovat tämän vuoksi erilaisia, niissä käytettävät XPath polut alkavat (useimmiten) senhetkisestä lähtöjoukosta (*current nodes*) käyttäen `./` polkumerkintää.

Tässä tehtävässä on tarkoitus keksiä oikeamuotoinen XPath lauseke, joka poimii halutun joukon solmuja, jotka ovat vastauksena esitettyyn kysymykseen. Tekstimuotoiset kysymykset XSL-dokumentissa tulee siis korvata oikeilla XPath lausekkeilla, minkä jälkeen suoritetaan XSL-muunnos-operaatio.

Tämän osatehtävän tarkoitus on alustavasti tutustua XPath:n käyttöön XSL-muunnoksissa ja muunnoksen ajamiseen on-line työkalulla. XSLT-kieleen tutustutaan tarkemmin vasta tehtävässä 3.

On-line XSLT tester: <http://www.xpathtester.com/xslt>

Lähtötiedostot:

- Kohde XML: *Assembly_line_ext.xml* (= tehtävässä 1 kirjoitettu laajennettu malli)
- XSL: *AL2HTML_transform.xml* (tämän tehtävän liitetiedosto)

2.1 XSLT muunnos AL2HTML

Ajettavan muunnoksen tarkoituksena on generoida HTML-muotoinen yhteenvetoraportti (web sivu) annetusta KP-linjan XML mallista *assembly_line_ext.xml* (ALML), joka toteutettiin tehtävässä 1. XSLT tiedosto *AL2HTML_transform.xml* sisältää valmiit sääntöjoukot (template) tämän muunnoksen toteuttamiseksi. Tehtävänä on ainoastaan täydentää siitä puuttuvat muutamat XPath-lausekkeet, jotka vastaavat ko. tekstimuotoisiin kysymyksiin.

XPath lausekkeiden täydentäminen

Tutustu tiedoston *AL2HTML_transform.xml* sisältöön ja tutki minkälaisia merkintöjä siinä on käytetty. Kaikki *xsl:* prefixillä alkavat elementit kuuluvat XSLT-kielen nimiavaruuteen. Muut elementit ovat HTML-kielen merkintöjä ja ne kopioituvat sellaisenaan generoitavaan HTML dokumenttiin.

Täydennä puuttuvat XPath lausekkeet tiedostossa *AL2HTML_transform.xml* korvaamalla kaikki kysymystekstit, jotka alkavat sanalla *QUESTION*, oikealla XPath lausekkeella

Esimerkkirivi:

```
Number of Modules #<em><xsl:value-of select="QUESTION: the number of all module children"/></em>
```

XPath lausekkeita kannattaa kuitenkin ensin testata online XPath testerillä käyttäen kohteena tiedostoa *assembly_line_ext.xml*. Kun oikeat XPath lausekkeet on löydetty online testerillä ne täytyy kirjoittaa *xsl*-koodiin hieman toisin, koska tällöin polun lähtökohta 'current node' ei olekaan dokumentin juuri-solmu. XSL template:n sisällä polun lähtökohtana olevana solmujoukkona on templatien *match*-argumentin arvon nimeämät solmut. Esimerkiksi `<xsl:template match="station">` template sisältäisi säännöt 'station' elementtien käsittelyyn, jolloin 'current node' olisi 'station' elementtien joukko.

Kirjoita `./` merkinnällä alkavat muokatut XPath polut *.xml* tiedostoon ja lopuksi muuta sen nimi muotoon *AL2HTML_transform.xml*

Assembly Line Report

Station ID:ST001

NAME:distribution_station

DESCRIPTION:Main task of a distribution station is to supply other stations with material

Total Station Capacity #12

Number of Modules #3

Module ID:M001

NAME:stack_magazine_module

Module Capacity #10

Module Speed (pcs/min):20

Number of Actuators #1

Kuva 1. AL2HTML muunnoksella *assembly_line_ext.xml* dokumentista generoidun html-sivun selainnäkymä.

2.2 AL2HTML muunnoksen ajaminen kolmella eri tavalla

HTML:n generointi online XSLT ohjelmalla <http://www.xpathtester.com/xslt>

Paina XSL kentän perässä olevaa 'set'-painiketta ja kopioi xsl tiedoston sisältö 'Content' kenttään ja paina 'Save'. Tämän jälkeen XML lähtödokumentti kopioidaan XML kenttään, jonka jälkeen voidaan ajaa XSLT muunnos painamalla 'Transform' painiketta. Muunnoksen tuloksena generoidun HTML/XML kooditekstin tulisi aueta uuteen selaimen välilehteen. Kopio teksti editorille ja talleta se tehtäväkansioon html-muotoiseksi tiedostoksi 'ALReport_html_online.html'. Avaa tiedosto web-selaimella tarkasteltavaksi. Aukeavan web-sivun tulisi näyttää lähes samalta kuin kuvassa 1.

HTML:n generointi suoraan web-selaimella

XML-dokumenttiin *assembly_line_ext.xml* tulee lisätä seuraava prosessointiohje toiselle riville heti xml-deklaraation jälkeen: `<?xml-stylesheet type="text/xsl" href="AL2HTML_transform.xsl"?>`

Tämä prosessointiohje mahdollistaa muunnoksen toteuttamisen suoraan selaimella avaamalla kohdetiedosto *assembly_line_ext.xml* suoraan web-selaimella (open with ...) Selain osaa suorittaa prosessointiohjeen osoittaman XSLT muunnoksen ja näyttää sen tuloksena generoidun html-tiedoston web-sivuna. Aukeavan web-sivun tulisi näyttää lähes samalta kuin kuvassa 1.

HTML:n generointi Notepad++ ohjelmalla

Avaa *assembly_line_ext.xml* tiedosto Notepad++ editorilla (Jos se on asennettuna). Valitse menusta *Plugins/XML tools/xsl transform*, jolloin 'XML Transformation Settings' dialogi aukeaa. Navigoi (...) ja avaa tiedosto *AL2HTML_transform.xsl* ja paina 'Transform' painiketta. Muunnoksen tuloksena generoidun HTML/XML kooditekstin tulisi aueta uuteen Notepad++:n välilehteen.

Tallenna se tehtäväkansioon html-muotoiseksi tiedostoksi 'ALReport_html_notepad.html' ja avaa tiedosto web-selaimella tarkasteltavaksi. Aukeavan web-sivun tulisi näyttää lähes samalta kuin kuvassa 1.

3. Reflektiokysymys

Tutustu *AL2HTML_transform.xsl* tyylisivuun hiukan tarkemmin ja pohdi, missä sen *template*:issä tulosedokumentin eri tekstikentät on generoitu ja numeroarvot laskettu. Entä miltä riveiltä löytyvät näiden *template*:iden kutsut `<xsl:apply-templates>`?