



Interneto technologijos

XSLT
Rekursija



Rekursija

- Instrukcija `xsl:apply-templates` suformuoja naują mazgų aibę, ir paleidžia antrą TM su nauja mazgų aibe, t.y.:
 - pirmoji TM pristabdoma,
 - įsimenama jos mazgų aibė ir einamasis mazgas
 - paleidžiama antra TM,
 - ji kažkada baigia savo darbą...
 - pirmos TM įsiminta mazgų aibė ir einamasis mazgas yra atstatomi, ir vėl atnaujinamas pirmos TM darbas
- Šablono vykdymo metu sutiktas statinis tekstas bei instrukcijų suformuotas tekstas yra kopijuojami į rezultato dokumentą



Instrukcija `xsl:apply-templates`

- Skirta rekursijos "užsukimui"

```
<xsl:apply-templates  
    select = "XPath išraiška, gražinanti  
            mazgų aibę"  
/>
```

- Jei `select` atributas praleistas, tai nauja mazgų aibė formuojama iš visų einamojo mazgo vaikų (`child::node()`)
- Jei `select` atributas yra nurodytas, tai jo XPath išraiškos gražinta mazgų aibė ir tampa nauja mazgų aibe



Rekursijos pavyzdys

- XML dokumentas:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<?xml-stylesheet href="transf.xslt"  
  type="text/xsl"?>
```

```
<knyga>
```

```
  <pavadinimas>XSLT</pavadinimas>
```

```
  <autorius>John Smith</autorius>
```

```
</knyga>
```

Pavyzdys su rekursija

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xsl:stylesheet version='1.0'
```

```
  xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
```

```
<xsl:template match="/">
```

```
  A <xsl:apply-templates/> B
```

```
</xsl:template>
```

Šioje vietoje pirma transformacijų mašina stabdoma, ir paleidžiama antra

```
<xsl:template match="knyga">
```

```
  C <xsl:apply-templates/> D
```

```
</xsl:template>
```

Šioje vietoje antra transformacijų mašina stabdoma, ir paleidžiama trečia

```
<xsl:template match="pavadinimas">
```

```
  <h1> <xsl:value-of select="."/> </h1>
```

```
</xsl:template>
```

```
<xsl:template match="autorius">
```

```
  <h2> <xsl:value-of select="."/> </h2>
```

```
</xsl:template>
```

Šiuos šablonus vykdo trečia mašina

```
</xsl:stylesheet>
```



Rezultatas

A

C

```
<h1>XSLT</h1>
```

```
<h2>John Smith</h2>
```

D

B

- Jei suprantate, kodėl rezultatas yra būtent toks, tai jūs suprantate, kaip veikia rekursija XSLT specifikacijoje



Detaliau

- A [antros mašinos rezultatas] B
- A [C [trečios mašinos rezultatas] D] B

A

C

```
<h1>XSLT</h1>
```

```
<h2>John Smith</h2>
```

D

B

Kitas pavyzdys

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version='1.0'
  xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
  <xsl:template match="/">
    A <xsl:apply-templates/> B
    AA <xsl:apply-templates select="//pavadinimas"/> BB
  </xsl:template>

  <xsl:template match="knyga">
    C <xsl:apply-templates/> D
  </xsl:template>

  <xsl:template match="pavadinimas">
    <h1> <xsl:value-of select="."/> </h1>
  </xsl:template>
  <xsl:template match="autorius">
    <h2> <xsl:value-of select="."/> </h2>
  </xsl:template>
</xsl:stylesheet>
```

Čia, grįžę iš pirmos
rekursijos, mes vėl
einam į rekursiją



Rezultatas

A

C

```
<h1>XSLT</h1>
```

```
<h2>John Smith</h2>
```

D

B

AA

```
<h1>XSLT</h1>
```

BB



Šablonai pagal nutylėjimą

- Yra rinkinys šablonų, kuriuos į kiekvieną transformaciją įtraukia pati transformacijų mašina
- *Rekursijos "užsukimas" pagal nutylėjimą:*

```
<xsl:template match="*|/">  
  <xsl:apply-templates/>  
</xsl:template>
```

- *Tekstinių mazgų ir atributų mazgų reikšmių kopijavimas pagal nutylėjimą:*

```
<xsl:template match="text()|@">  
  <xsl:value-of select="."/>  
</xsl:template>
```

Pavyzdys, kuris gali būti sutrumpintas

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version='1.0'
  xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
  <xsl:template match="/">
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="knyga">
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="pavadinimas">
    <h1> <xsl:value-of select="."/> </h1>
  </xsl:template>
  <xsl:template match="autorius">
    <h2> <xsl:value-of select="."/> </h2>
  </xsl:template>
</xsl:stylesheet>
```

Šiuos du šablonus galėtų vykdyti šablonas pagal nutylėjimą



Sutrupintas pavyzdys

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version='1.0'
  xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>

  <xsl:template match="pavadinimas">
    <h1> <xsl:value-of select="." /> </h1>
  </xsl:template>

  <xsl:template match="autorius">
    <h2> <xsl:value-of select="." /> </h2>
  </xsl:template>

</xsl:stylesheet>
```



Spąstai – kas bus šios transformacijos rezultatas?

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version='1.0'
  xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>

  <xsl:template match="/">
    A
  </xsl:template>

  <xsl:template match="pavadinimas">
    kuku
  </xsl:template>
</xsl:stylesheet>
```



Atsakymas

- Teisingai: A
- Neteisingai: A kuku
- Kodėl? Kadangi mūsų pirmo šablono testas gaudo "/", tai šablonas pagal nutylėjimą nesuveiks (jo mažesnis prioritetas). Mūsų šablonas rekursijos neužsuka – transformacijų mašina apdoroja vienintelį savo aibės elementą – mazgą "/" – ir baigia darbą. "knyga" niekas neiškviečia.

Apibendrinimas:

Transformacijos vykdymas

- Transformacijų mašinai įėjime paduodama mazgų aibė A
 - I-ai TM: $A = \{ / \}$
- Ciklas: kiekvienam aibės A nariui m :
 - TM ieško šablono, kurio testas (match) atitiktų mazgą m
 - Jei šablonas randamas, tai vykdomos jo instrukcijos
 - Mazgo m apdorojimas baigiamas
- Šablonai pagal nutylėjimą:

```
<xsl:template match="*|/">  
  <xsl:apply-templates/>  
</xsl:template>
```

```
<xsl:template match="text()|@">  
  <xsl:value-of select="."/>  
</xsl:template>
```



Pavyzdys – XML dokumentas

- XML dokumentas:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<?xml-stylesheet href="transf.xslt" type="text/xsl"?>
```

```
<knyga>
```

```
  <pavadinimas>XSLT</pavadinimas>
```

```
  <autorius>John Smith</autorius>
```

```
</knyga>
```




Transformacija, pvz. 1

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version='1.0'
  xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>

  <xsl:template match="pavadinimas">
    Kuku
  </xsl:template>

</xsl:stylesheet>
```

■ Rezultatas:

Kuku
John Smith

Čia rezultatas "Kuku" sutampa su tuo, ką tikimės gauti, bet "John Smith" principu "ant pirštų" neįmanoma paaiškinti



Transformacija, pvz. 2

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version='1.0'
  xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
```

```
<xsl:template match="/">
```

```
  Tra lia lia
```

```
</xsl:template>
```

```
<xsl:template match="pavadinimas">
```

```
  Kuku
```

```
</xsl:template>
```

```
</xsl:stylesheet>
```

■ Rezultatas:

```
Tra lia lia
```

"Kuku" nėra spausdinamas! Principas "ant pirštų" vėl netinka, nes pirmame šablone buvo nutraukta rekursija (t.y., neužsukta rekursija vaikiniam elementams)

Transformacija, pvz. 3

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version='1.0'
  xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>

  <xsl:template match="/">
    Tra lia lia
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="pavadinimas">
    Kuku
  </xsl:template>
</xsl:stylesheet>
```

Pratęsiame "visų mazgų prabėgimą", todėl bus atspausdintas ir "Kuku" su "John Smith"

■ Rezultatas:

Tra lia lia
Kuku
John Smith



Atributų mazgų apdorojimas

- `apply-templates` be `select` XML dokumento atributų nepaims!
 - Ims einamojo mazgo vaikus (`child::*`), o atributai pasiekiami ašimi `attribute` (`attribute::*` arba `@*`)
- Todėl, norėdami su `apply-templates` paimti, pavyzdžiui, visus einamosios žymės atributus, turime rašyti:
 - `<apply-templates select="@*">`



Pavyzdys

```
<a att1="ing">
  te
  <b att2="ai">
    is
  </b>
</a>
```

```
<xsl:stylesheet ...>
  <xsl:template match="/">
    <xsl:apply-templates
      select="* | //@*" />
  </xsl:template>
</xsl:stylesheet>
```

```
<xsl:template match="* | /">
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="text() | @*">
  <xsl:value-of select="." />
</xsl:template>
```

Šablonas pagal
nutylėjimą P (Pirmas)

Šablonas pagal
nutylėjimą A (Antras)



Atsakymas

- Aibė {/}, startuoja I-a TM, ima "/", randa vartotojo šabloną, vykdo `apply-templates` instrukcijos `select` attribute esantį XPath, gauna aibę {a, att1, att2}, paleidžia II-ą TM
- II-a TM ima pirmą aibės elementą - žymę a, vartotojo šablono neranda, randa šabloną P, kuris vykdo instrukciją `apply-templates`, suformuoja vaikų aibę {"te", b}, paleidžia III-ią TM
- III-ia TM ima "te", randa šabloną pagal nutylėjimą A, vykdo jo instrukciją `value-of`, išveda tekstą "te"
- III-ia TM ima paskutinį aibės elementą - žymę b, vykdo šablono P instrukciją `apply-templates`, suformuoja vaikų aibę {"is"}, paleidžia IV-ą TM
- IV-a TM ima "is", vykdo šabloną A, spausdina tekstą "is", baigia darbą
- III-ia baigia darbą
- II-a TM ima antrą aibės narį - atributą att1, vartotojo šablono neranda, randa šabloną A, vykdo jo instrukciją `value-of`, spausdina tekstą "ing"
- II-a TM ima paskutinį aibės narį att2, --/--, spausdina "ai", baigia darbą
- I-a baigia darbą
- **Atsakymas:** "teisingai"