

INSY - Informationssysteme



htl krems
Bautechnik & IT

KW 20/2021 –
KW 20/2021

ЗАНИТ

Felix Schneider

19.05.2021

INSY_2

DI (FH) Alexander MESTL

—

XML

Inhaltsverzeichnis:

1	Aufgabenstellung.....	3
1.1	XML Schema	3
1.2	XPath	3
2	Abstract (English).....	3
2.1	XML Schema	3
2.2	XPath	4
3	Theoretische Grundlagen	5
3.1	XML Schema – kurz und knackig	5
3.2	XPath – kurz und knackig.....	5
4	Übungsdurchführung	6
4.1	einfache Datentypen	6
4.1.1	Übung 1 – Color	6
4.1.2	Übung 2 – Personenname	7
4.2	komplexe Datentypen	8
4.2.1	Übung 1 – Book	8
4.2.2	Übung 2 – Book2	9
4.2.3	Übung 4 – Person	10
4.3	XML (II) - XPath-Aufgaben	12
4.3.1	Knotenbaum malen	12
4.3.2	Lokalisierungspfade.....	13
4.3.2.1	1. Beispiel	13
4.3.2.2	2. Beispiel	17
4.3.2.3	5. Beispiel	20
5	Ergebnisse.....	26
6	Code.....	26
7	Kommentar.....	26

1 Aufgabenstellung

1.1 XML Schema

Mit Hilfe eines Schemas kann man die Ausgestaltung eines XML-Dokumentes definieren. Nach dem Durcharbeiten des Kapitels 26 sollte es euch möglich sein, die drei Aufgaben zu den "einfachen Datentypen" auf den Seiten 24 und 25 des Aufgabendokuments "insyExercise3.pdf" (neu in den Kursmaterialien) zu lösen.

Zum besseren Verständnis hab ich ein Beispiel gemacht und euch einen Screenshot davon angefügt. Ich verwende IntelliJ, aber ihr könnt die IDE / den Editor selber frei wählen. Wie man sich Einträge für neue XML- und XSD-Dateien im "New"-Menü anlegt, findet ihr unter dem JetBrains-Link!

Nach den einfachen Datentypen wenden wir uns den komplexen Datentypen zu. Das sind Datentypen, die aus mehreren Elementen einfacher Datentypen bestehen - das sollte nach dem Durchlesen des Kapitels 26 bereits bekannt sein.

Dazu machen wir die Beispiele Nr. 1 auf Seite 26, Nr. 2 auf Seite 27 und Nr. 4 auf Seite 28 des Übungsdokuments "insyExercise3.pdf".

Je nach IDE wird bereits während des Schreibens der XML-Datei bei gegebenem Schema die Korrektheit der Daten überprüft und/oder über einen eigenen Aufruf "Validierung" (oder so ähnlich). Wer nur mit einem Texteditor arbeitet, kann seine XML-Dokumente auch online validieren lassen, z. B. auf <https://www.xmlvalidation.com>.

1.2 XPath

Aus dem Aufgabendokument "insyExercise3.pdf" ist das zweite Beispiel zu den XPath-Grundlagen auf Seite 33 (der "Knotenbaum") zu lösen.

Danach sind die Aufgaben "3.2. Lokalisierungspfade - 1. Beispiel" auf Seite 34, "3.2. Lokalisierungspfade - 2. Beispiel" auf Seite 35 und "3.2. Lokalisierungspfade - 5. Beispiel" auf Seite 36 durchzuführen. Die xml-Dateien sind davor auf ihre Wohlgeformtheit zu überprüfen und gegebenenfalls zu korrigieren.

2 Abstract (English)

2.1 XML Schema

You can use a schema to define the layout of an XML document.

After working through chapter 26 you should be able to solve the three tasks on "simple data types" on pages 24 and 25 of the task document "insyExercise3.pdf" (new in the course materials).

For better understanding I made an example and attached a screenshot of it for you. I use IntelliJ, but you are free to choose the IDE / editor yourself. How to create entries for new XML and XSD files in the "New" menu you can find under the JetBrains link!

After the simple data types we turn to the complex data types. These are data types that consist of multiple elements of simple data types - this should already be familiar after reading through Chapter 26.

For this we do the examples no. 1 on page 26, no. 2 on page 27 and no. 4 on page 28 of the exercise document "insyExercise3.pdf".

Depending on the IDE, the correctness of the data is already checked during the writing of the XML file for a given schema and/or via a separate call "validation" (or something like that). Those who only work with a text editor can also have their XML documents validated online, e.g. at <https://www.xmlvalidation.com>.

2.2 XPath

From the task document "insyExercise3.pdf", the second example on XPath basics on page 33 (the "node tree") is to be solved.

Afterwards the tasks "3.2. localization paths - 1st example" on page 34, "3.2. localization paths - 2nd example" on page 35 and "3.2. localization paths - 5th example" on page 36 have to be performed. The xml files must be checked beforehand for their well-formedness and corrected if necessary.

3 Theoretische Grundlagen

detailliert: INSY-Script

3.1 XML Schema – kurz und knackig

Es gibt zwei Datentypen: einfache, komplexe.

- Einfach Datentypen enthalten Standardtypen, wie zum Beispiel int, string, ...
- Komplexe Datentypen enthalten verschachtelt einfache Datentypen.

Mit restriction kann man simpleTypes einschränken (Länge, Enum, ...)

Mit selection kann man einstellen, welche Datentypen nacheinander im komplexen Datentyp vorkommen. Dabei nutzt man xs:element, um ein Element innerhalb des komplexen Typens zu beschreiben. Wenn ein Element dabei öfter vorkommen können soll, verwendet man das Attribut maxOccurs="unbounded", wie Sie beim letzten Beispiel sehen werden.

3.2 XPath – kurz und knackig

Mithilfe von XPath kann man Elemente selektieren (vergleichbar mit dem find-Befehl bei MongoDB).

Wikipedia:

Die XML Path Language (XPath) ist eine vom W3-Konsortium entwickelte Abfragesprache, um Teile eines XML-Dokumentes zu adressieren und auszuwerten. XPath dient als Grundlage einer Reihe weiterer Standards wie XSLT, XPointer und XQuery.

4 Übungsdurchführung

4.1 einfache Datentypen

4.1.1 Übung 1 – Color

► Aufgabenstellung: XML Schema schreiben ▼

- SCHREIBEN SIE EINEN EINFACHEN XML SCHEMA DATENTYPEN ZUR KLASSIFIZIERUNG VON FARBEN
- DER DATENTYP SOLL SO RESTRIKTIV WIE MÖGLICH SEIN.

► Lösung: XML Schema ▼

```

1  <!-- ----- -->
2  <!--           color.xsd           -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <xs:schema
6      xmlns:xs="http://www.w3.org/2001/XMLSchema">
7
8
9  </xs:schema>
10
11
12
13 <!-- Farben -->
14 Black, Lightslategray
15 Gray
16 Silver
17 White
18 Ivory
19 Linen
20 Beige
21 Khaki
22 Goldenrod
23 Dardred
24 Maroon

```

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="color">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Black" />
      <xs:enumeration value="Lightslategray" />
      <xs:enumeration value="Gray" />
      <xs:enumeration value="Silver" />
      <xs:enumeration value="White" />
      <xs:enumeration value="Ivory" />
      <xs:enumeration value="Linen" />
      <xs:enumeration value="Beige" />
      <xs:enumeration value="Khaki" />
      <xs:enumeration value="Goldenrod" />
      <xs:enumeration value="Dardred" />
      <xs:enumeration value="Maroon" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="element" type="color" />
  <xs:element name="root" />
</xs:schema>

```

```

<?xml version="1.0" encoding="UTF-8" ?>
<?xml-model href="color.xsd" ?>
<root>
  <element>White</element>
  <element>Black</element>
  <element>Yellow</element>
  <element>Green</element>
</root>

```

4.1.2 Übung 2 – Personenname

► Aufgabenstellung: XML Schema schreiben ▼

- SCHREIBEN SIE EINEN EINFACHEN DATENTYPEN ZUR DARSTELLUNG VON PERSONENNAMEN
- ACHTEN SIE DARAUF DASS EIN PERSONENNAME MINIMAL 2 ZEICHEN UND MAXIMAL 50 ZEICHEN HAT.

► Lösung: XML Schema ▼

```

1  <!-- ----- -->
2  <!--          persondata.xsd          -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <xs:schema
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
6

```

```

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:simpleType name="personenname">
        <xs:restriction base="xs:string">
            <xs:minLength value="2" />
            <xs:maxLength value="50" />
        </xs:restriction>
    </xs:simpleType>

    <xs:element name="personenname" type="personenname" />
    <xs:element name="root" />
</xs:schema>

```

```

<?xml version="1.0" encoding="UTF-8" ?>
<?xml-model href="personenname.xsd" ?>

<root>
    <personenname>Felix Christian Schneider</personenname>
    <personenname>B</personenname>
    <personenname>Felix Christian Schneider Anton Xaver Edlinger Clemens Josef Schlipfinger</personenname>
</root>

```

4.2 komplexe Datentypen

4.2.1 Übung 1 – Book

1. Beispiel - Komplexe Datentypen



Beispielbeschreibung ▾

- **Schwerpunkt:** XML SCHEMA, EINFACHE DATENTYPEN, KOMPLEXE DATENTYPEN, SEQUENZ
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 248-253, 254

► Aufgabenstellung: XML Schema schreiben ▾

- SCHREIBEN SIE FÜR DAS FOLGENDE XML DOKUMENT EIN XML SCHEMA
- ACHTEN SIE AUF DIE VERKNÜPFUNG DES XML DOKUMENTS UND DES SCHEMAS.
- DIE ELEMENTE DES XML DOKUMENTS KOMMEN GENAU IN DER REIHENFOLGE VOR WIE SIE IM XML DOKUMENT AUFTRETEN.

► Codebeispiel: XML Dokument ▾

```

1  <!-- ----- -->
2  <!--          book.xml          -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <book xmlns:xs=
6      "http://www.w3.org/2001/XMLSchema-instance"
7      xs:noNamespaceSchemaLocation="book.xsd">
8      <title>Moby Dick</title>
9      <author>Herman Melville</author>
10     <print>5-5-1927</print>
11     <pages>814</pages>
12     <description>
13         Das erzählerische Rueckgrat des Romans ist
14         die schicksalhafte Fahrt des Walfangschiffes
15         Pequod, dessen Kapitaen Ahab mit blindem Has
16         den weissen Pottwal Moby Dick jagt, der ihm
17         ein Bein abgerissen hat.
18     </description>
19 </book>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="book">
    <xs:sequence>
      <xs:element name="title" type="xs:string" />
      <xs:element name="author" type="xs:string" />
      <xs:element name="print" type="xs:date" />
      <xs:element name="pages" type="xs:int" />
      <xs:element name="description" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="book" type="book" />
  <xs:element name="root" />
</xs:schema>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="book.xsd">
<root xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
      xs:noNamespaceSchemaLocation="book.xsd">
  <book>
    <title>Moby Dick</title>
    <author>Herman Melville</author>
    <print>1927-05-05</print>
    <pages>814</pages>
    <description>
      Das erzählerische Rueckgrat des Romans ist
      die schicksalhafte Fahrt des Walfangschiffes
      Pequod, dessen Kapitaen Ahab mit blindem Hass
      den weissen Pottwal Moby Dick jagt, der ihm
      ein Bein abgerissen hat.
    </description>
  </book>
  <book>
    <title>Herr der Ringe</title>
    <author>unbekannt</author>
    <print>5-5-1927</print>
    <pages>somestring is not possible</pages>
    <description>
      Lorem ipsum dolor sit amet consectetur adipiscing
      doloremque exercitationem quae pariatur repeller
      veritatis minus iste sint iure vitae voluptatibu
    </description>
  </book>
  <book>
    <author>Herman Melville</author>
    <title>Moby Dick</title>
    <pages>814</pages>
    <description>
      LOL
    </description>
    <print>nop</print>
  </book>
</root>

```


4.2.2 Übung 2 - Book2

2.Beispiel - Komplexe Datentypen



Beispielbeschreibung ▾

- **Schwerpunkt:** XML SCHEMA, EINFACHE DATENTYPEN, KOMPLEXE DATENTYPEN, SEQUENZ
- **Komplexität:** EINFACH
- **Skriptum:** SEITE 248-253, 254

► Aufgabenstellung: XML Schema schreiben ▾

- SCHREIBEN SIE FÜR DAS FOLGENDE XML DOKUMENT EIN XML SCHEMA
- ACHTEN SIE AUF DIE VERKNÜPFUNG DES XML DOKUMENTS UND DES SCHEMAS.
- DIE ELEMENTE DES XML DOKUMENTS KOMMEN GENAU IN DER REIHENFOLGE VOR WIE SIE IM XML DOKUMENT AUFTRETEN.

► Codebeispiel: XML Dokument ▾

```

1 <!-- ----- -->
2 <!--          book.xml          -->
3 <!-- ----- -->
4 <?xml version="1.0" encoding="UTF-8"?>
5 <book xmlns:xs=
6   "http://www.w3.org/2001/XMLSchema-instance"
7   xs:noNamespaceSchemaLocation="book.xsd">
8   <title>Moby Dick</title>
9   <author>
10    <given-name>Herman</given-name>
11    <middle-name>J. L.</middle-name>
12    <last-name>Melville</last-name>
13  </author>
14  <print>5-5-1927</print>
15  <pages>814</pages>
16  <description>
17    Das erzählerische Rueckgrat des Romans ist
18    die schicksalhafte Fahrt des Walfangschiffes
19    Pequod, dessen Kapitaen Ahab mit blindem Hass
20    den weissen Pottwal Moby Dick jagt, der ihm
21    ein Bein abgerissen hat.
22  </description>
23  <literature>
24    <symbol>
25      Ahabs Jagd auf den Weissen Wal steht im
26      Widerspruch zu den materiellen Interessen
27      von Mannschaft und Eignern. Dem Ersten
28      Maat Starbuck erscheint Ahabs Rachsucht
29      gegen das unvernueftige Tier Moby Dick
30      gotteslaesterlich.
31    </symbol>
32  </literature>
33 </book>

```

```

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:complexType name="book">
    <xs:sequence>
      <xs:element name="title" type="xs:string" />
      <xs:element name="author" type="author" />
      <xs:element name="print" type="xs:date" />
      <xs:element name="pages" type="xs:int" />
      <xs:element name="description" type="xs:string" />
      <xs:element name="literature" type="literature" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="author">
    <xs:sequence>
      <xs:element name="given-name" type="xs:string" />
      <xs:element name="middle-name" type="xs:string" />
      <xs:element name="last-name" type="xs:string" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="literature">
    <xs:sequence>
      <xs:element name="symbol" type="xs:string" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="book" type="book" />
  <xs:element name="root" />
</xs:schema>

```

```

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

<root xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">
  <book>
    <title>Moby Dick</title>
    <author>
      <given-name>Herman</given-name>
      <middle-name>J. L.</middle-name>
      <last-name>Melville</last-name>
    </author>
    <print>1927-05-05</print>
    <pages>814</pages>
    <description>
      Das erzählerische Rueckgrat des Romans ist
      die schicksalhafte Fahrt des Walfangschiffes
      Pequod, dessen Kapitaen Ahab mit blindem Hass
      den weissen Pottwal Moby Dick jagt, der ihm
      ein Bein abgerissen hat.
    </description>
    <literature>
      <symbol>
        Ahabs Jagd auf den Weissen Wal steht im
        Widerspruch zu den materiellen Interessen
        von Mannschaft und Eignern. Dem Ersten
        Maat Starbuck erscheint Ahabs Rachsucht
        gegen das unvernueftige Tier Moby Dick
        gotteslaesterlich.
      </symbol>
    </literature>
  </book>
</root>

```

4.2.3 Übung 4 – Person

4.Beispiel - Komplexe Datentypen



Beispielbeschreibung ▾

- **Schwerpunkt:** XML SCHEMA, EINFACHE DATENTYPEN, KOMPLEXE DATENTYPEN
- **Komplexität:** MITTEL
- **Skriptum:** SEITE 248-253, 254

▸ Aufgabenstellung: XML Schema schreiben ▾

- SCHREIBEN SIE FÜR DAS FOLGENDE XML DOKUMENT EIN XML SCHEMA
- ACHTEN SIE AUF DIE VERKNÜPFUNG DES XML DOKUMENTS UND DES SCHEMAS.

▸ Codebeispiel: XML Dokument ▾

```

1  <!-- ----- -->
2  <!--          euro.xml          -->
3  <!-- ----- -->
4  <?xml version="1.0" encoding="UTF-8"?>
5  <euro xmlns:xs=
6      "http://www.w3.org/2001/XMLSchema-instance"
7      xs:noNamespaceSchemaLocation="euro.xsd">
8      <teams>
9          <team>Schwitzer Team</team>
10         <team>France</team>
11     </teams>
12     <player>Gareth Bale</player>
13     <player>Jmai Vardi</player>
14     <qualifying>
15         <group>
16             <team-play>FRA</team-play>
17             <team-play>SUI</team-play>
18             <news>
19                 The last match in Group A ended
20             </news>
21         </group>
22         <group>
23             <team-play>RUS</team-play>
24             <team-play>GER</team-play>
25             <team-play>JAP</team-play>
26         </group>
27     </qualifying>
28 </euro>

```

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="euro">
    <xs:sequence>
      <xs:element name="teams" type="teams" />
      <xs:element name="player" type="xs:string" maxOccurs="unbounded" />
      <xs:element name="qualifying" type="qualifying" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="teams">
    <xs:sequence>
      <xs:element name="team" type="xs:string" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="qualifying">
    <xs:sequence>
      <xs:element name="group" type="group" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="group">
    <xs:sequence>
      <xs:element name="team-play" type="xs:string" maxOccurs="unbounded" />
      <xs:element name="news" type="xs:string" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="euro" type="euro" />
  <xs:element name="root" />
</xs:schema>
```

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

<root xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:noNamespaceSchemaLocation="euro.xsd">
  <euro>
    <teams>
      <team>Schwitzer Team</team>
      <team>France</team>
    </teams>
    <player>Gareth Bale</player>
    <player>Jmai Vardi</player>
    <qualifying>
      <group>
        <team-play>FRA</team-play>
        <team-play>SUI</team-play>
        <news>
          The last match in Group A ended
        </news>
      </group>
      <group>
        <team-play>RUS</team-play>
        <team-play>GER</team-play>
        <team-play>JAP</team-play>
      </group>
    </qualifying>
  </euro>
</root>
```

4.3 XML (II) - XPath-Aufgaben

4.3.1 Knotenbaum malen

Dieser Baum wurde mithilfe von <https://www.xmlviewer.org/> erstellt:



Dieser Baum wird von oben nach unten durchnummeriert (die Reihenfolge ist linear).

4.3.2 Lokalisierungspfade

4.3.2.1 1. Beispiel

Aufgabenstellung: Lokalisierungspfade

- DEFINIEREN SIE DIE FOLGENDEN **Lokalisierungspfade**.

1. Finden Sie alle <subproject> Elemente

Kontext: Wurzelknoten

XPATH

XPath Expression:

`//subproject`

Context Node:

`/project`

Reset

Check

```
</partners>
<subprojects>
  <subproject id="342566" code="simulation">
    <name>Computersimulation</name>
    <started-at>21.12.2014</started-at>
    <institute>
      Angewandte Mathematik
    </institute>
    <stuff>
      <project-manager>
        <first-name>
          Christian Karl
        </first-name>
        <last-name>
          Lauer
        </last-name>
      </project-manager>
      <member>
        <first-name>
          Mika
        </first-name>
        <last-name>
          Harkonnen
        </last-name>
      </member>
      <member>
        <first-name>
          Stefan
        </first-name>
        <last-name>
          Jell
        </last-name>
      </member>
    </stuff>
  </subproject>
</subprojects>
```

2. Finden Sie alle `<subproject>` Elemente
Kontext: subprojects

XPATH

XPath Expression:

`/subproject`

Context Node:

`/project/subprojects`

Reset

Check

```

</partners>
<subprojects>
  <subproject id="342566" code="simulation">
    <name>Computersimulation</name>
    <started-at>21.12.2014</started-at>
    <institute>
      Angewandte Mathematik
    </institute>
    <stuff>
      <project-manager>
        <first-name>
          Christian Karl
        </first-name>
        <last-name>
          Lauer
        </last-name>
      </project-manager>
      <member>
        <first-name>
          Mika
        </first-name>
        <last-name>
          Harkonnen
        </last-name>
      </member>
      <member>
        <first-name>
          Stefan
        </first-name>
        <last-name>
          Jell
        </last-name>
      </member>
    </stuff>
  </subproject>
</subprojects>

```

3. Finden Sie alle id Attribute der `<subproject>` Elemente.
Kontext: Wurzelknoten

XPATH

XPath Expression:

`//subproject/@id`

Context Node:

`/project`

Reset

Check

```

<subprojects>
  <subproject id="342566" code="simulation">
    <name>Computersimulation</name>

```

4. Finden Sie alle Mitarbeiter der Subprojekte.

Kontext: Wurzelknoten

XPath Expression:

```
//subprojects//member
```

Context Node:

```
/project
```

Reset

Check

```
<member>
  <first-name>
    Mika
  </first-name>
  <last-name>
    Harkonnen
  </last-name>
</member>
<member>
  <first-name>
    Stefan
  </first-name>
  <last-name>
    Jell
  </last-name>
</member>
stuff>
```

5. Finden Sie die Namen aller Mitarbeiter der Subprojekte.

Kontext: project

XPath Expression:

```
//subprojects//stuff//first-name
```

Context Node:

```
/project
```

Reset

Check

```
<stuff>
  <project-manager>
    <first-name>
      Christian Kar.
    </first-name>
    <last-name>
      Lauer
    </last-name>
  </project-manager>
  <member>
    <first-name>
      Mika
    </first-name>
    <last-name>
      Harkonnen
    </last-name>
  </member>
  <member>
    <first-name>
      Stefan
    </first-name>
    <last-name>
      Jell
    </last-name>
  </member>
</stuff>
subproject>
```

6. Finden Sie die Namen aller Mitarbeiter der Subprojekte.

Kontext: partners

XPath Expression:

Context Node:

Reset

Check

Found no results for expression '//subprojects//stuff//first-name' within context '//partners'

7. Finden Sie alle Attribute der Subprojekte

Kontext: Wurzelknoten

XPath Expression:

Context Node:

Reset

Check

```
</partners>  
<subprojects>  
  <subproject id="342566" code="simulation">  
    <name>Computersimulation</name>  
    <started-at>21.12.2014</started-at>
```


4.3.2.2 2. Beispiel

DEFINIEREN SIE DIE FOLGENDEN **Lokalisierungspfade**.

DIE EINGABEDATEI IST DIE **EURO.XML** DATEI.

1. Alle <nation> Elementknoten Kontext: Wurzelknoten

XPath Expression:

 Context Node:


```
<euro year="2016">
  <teams>
    <team>
      <nation>
        <name>France</name>
        <short-name>FRA</short-name>
      </nation>
      <position type="defense">
        <player>
```

2. Alle <nation> Elementknoten Kontext: <winner> Element

XPath Expression:

 Context Node:

Found no results for expression '//nation' within context '//winner'

3. Alle <last-name> Elementknoten der <player> Knoten Kontext: Wurzelknoten

XPath Expression:

//player//last-name

Context Node:

/euro

Reset

Check

4. Alle <last-name> Elementknoten der <player> Knoten
Kontext: <winner> Element

XPath Expression:

//player//last-name

Context Node:

//winner

Reset

Check

Found no results

5. Alle <short-name> Elementknoten der <team> Knoten
Kontext: Wurzelknoten

```
<position type="defense">
  <player>
    <pl>
      <first-name>Laurent</first-name>
      <last-name>Koscielny</last-name>
    </pl>
  </player>
  <club>Arsenal London</club>
</position>
<position type="playground" />
<position type="attack">
  <player>
    <name>
      <first-name>Antoine</first-name>
      <last-name>Griezmann</last-name>
    </name>
    <club>Atletico Madrid</club>
  </player>
  <player>
    <name>
```

XPath Expression:

//team//short-name

Context Node:

/euro

Reset

Check

```
<team>
  <nation>
    <name>France</name>
    <short-name>FRA</short-name>
  </nation>
```

6. Alle <short-name> Elementknoten der <team>
Knoten
Kontext: <winner> Element

XPath Expression:

//team//short-name

Context Node:

//winner

Reset

Check

Found no results

7. Alle id Attribute der Gruppen
Kontext: Wurzelknoten

XPath Expression:

Context Node:

Reset

Check

```

</teams>
<groups>
  <group id="1">
    <team>FRA</team>
    <team>SPA</team>
    <team>GER</team>
    <team>GBS</team>
    <team>POR</team>
    <team>BEL</team>
  </group>
</groups>

```

4.3.2.3 5. Beispiel

- GEBEN SIE FÜR JEDEN DER FOLGENDEN XPATH DEN TYP DES LÖSUNGSOBJEKTS AN.

<!-- ----- -->

1. Finden Sie die Nachnamen aller französischen Abwehrspieler.

Kontext: Wurzelknoten

XPath Expression:

Context Node:

Reset

Check

```

<position type="defense">
  <player>
    <pl>
      <first-name>Laurent</first-name>
      <last-name>Koscielny</last-name>
    </pl>
    <club>Arsenal London</club>
  </player>
  <player>
    <name>
      <first-name>Djibril</first-name>
      <last-name>Sidibe</last-name>
    </name>
    <club>AS Monaco</club>
  </player>
</position>

```

2. Finden Sie die Namen der Klubs in denen die spanischen Stürmer spielen.

Kontext: Wurzelknoten

XPath Expression:

```
//team[nation/short-name="SPA"]/position[@type="attack"]//club
```

Context Node:

```
/euro
```

Reset

Check

```
</position>
<position type="attack">
  <player>
    <name>
      <first-name>David</first-name>
      <last-name>Villa</last-name>
    </name>
    <club>FC Barcelona</club>
  </player>
  <player>
    <name>
      <first-name>Diego</first-name>
      <last-name>Costa</last-name>
    </name>
    <club>Atletico Madrid</club>
  </player>
  <player>
    <name>
      <first-name>Lucas</first-name>
      <last-name>Vazquez</last-name>
    </name>
    <club>Real Madrid</club>
  </player>
  <player>
    <name>
      <first-name>Aritz</first-name>
      <last-name>Aduriz</last-name>
    </name>
    <club>Athletic Bilbao</club>
  </player>
</position>
</team>
```

3. Finden Sie die Namen aller Spieler die im Sturm spielen

Kontext: Wurzelknoten

XPath Expression:

```
//team/position[@type="attack"]//name
```

Context Node:

```
/euro
```

Reset

Check

```
</position>
<position type="playground" />
<position type="attack">
  <player>
    <name>
      <first-name>Antoine</first-name>
      <last-name>Griezmann</last-name>
    </name>
    <club>Atletico Madrid</club>
  </player>
  <player>
    <name>
      <first-name>Olivier</first-name>
      <last-name>Giroud</last-name>
    </name>
    <club>Arsenal</club>
  </player>
  <player>
    <name>
      <first-name>Nabil</first-name>
      <last-name>Fekir</last-name>
    </name>
    <club>Lyon</club>
  </player>
  <player>
    <name>
      <first-name>Florian</first-name>
      <last-name>Thauvin</last-name>
    </name>
    <club>Marseille</club>
  </player>
  <player>
    <name>
      <first-name>Alexandre</first-name>
      <last-name>Lacazette</last-name>
    </name>
    <club>Lyon</club>
  </player>
</position>
```

4. Finden Sie die `<name>` Elemente aller `<team>` Teams die unter der ersten 8 sind.
Kontext: Wurzelknoten

XPATH

XPath Expression:

```
/teams/team/nation[short-name/text() = //last-eight/match/team/text()]/name
```

Context Node:

```
/euro
```

Reset

Check

```
teams>
  <team>
    <nation>
      <name>France</name>
      <short-name>FRA</short-name>
    </nation>
    <position type="defense">
      <player>
        <pl>
          <first-name>Laurent</first-name>
          <last-name>Koscielny</last-name>
        </pl>
        <club>Arsenal London</club>
      </player>
      <player>
        <name>
          <first-name>Djibril</first-name>
          <last-name>Sidibe</last-name>
        </name>
        <club>AS Monaco</club>
      </player>
    </position>
    <position type="playground" />
    <position type="attack">
      <player>
        <name>
          <first-name>Antoine</first-name>
          <last-name>Griezmann</last-name>
        </name>
        <club>Atletico Madrid</club>
      </player>
    </position>
  </team>
</teams>
```

5. Geben Sie die Anzahl aller Spieler an, die an der Euro teilnehmen.

XPath Expression:
`count(//player)`

Context Node:
`/euro`

Reset

Check

The expression 'count(//player)' within context '/euro' resulted in: 170

6. Geben Sie die Anzahl aller französischen Abwehrspieler an.

XPATH

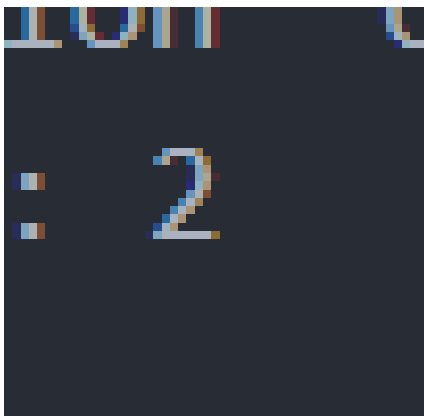
XPath Expression:
`count(//teams/team[nation/name = "France"]/position[@type = "defense"]/player)`

Context Node:
`/euro`

Reset

Check

The expression 'count(//teams/team[nation/name = "France"]/position[@type = "defense"]/player)' within context '/euro' resulted in: 2



7. Geben Sie für jede Position der spanischen Mannschaft den ersten Spieler aus.

XPATH

XPath Expression:

`//team[nation/name = "Spain"]/position/player[1]`

Context Node:

`/euro`

Reset

Check

Found 3 results for expression '`//team[nation/name = "Spain"]/p`
Found node of type Element with name 'player' at line 70
Found node of type Element with name 'player' at line 107
Found node of type Element with name 'player' at line 144
The expression '`//team[nation/name = "Spain"]/position/player[1`

```
</team>
<team>
  <nation>
    <name>Spain</name>
    <short-name>SPA</short-name>
  </nation>
  <position type="defense">
    <player>
      <name>
        <first-name>Jordi</first-name>
        <last-name>Alba</last-name>
      </name>
      <club>FC Barcelona</club>
    </player>
    <player>
      <name>
        <first-name>Dani</first-name>
        <last-name>Carvajal</last-name>
      </name>
      <club>Real Madrid</club>
    </player>
    <player>
      <name>
        <first-name>Sergio</first-name>
        <last-name>Ramos</last-name>
      </name>
      <club>Real Madrid</club>
    </player>
  </position>
</team>
```

5 Ergebnisse

6 Code

7 Kommentar