

---

# XML-Technologien, Übungsblatt 3: XSLT - Mancala-Spielbrett

## Table of Contents

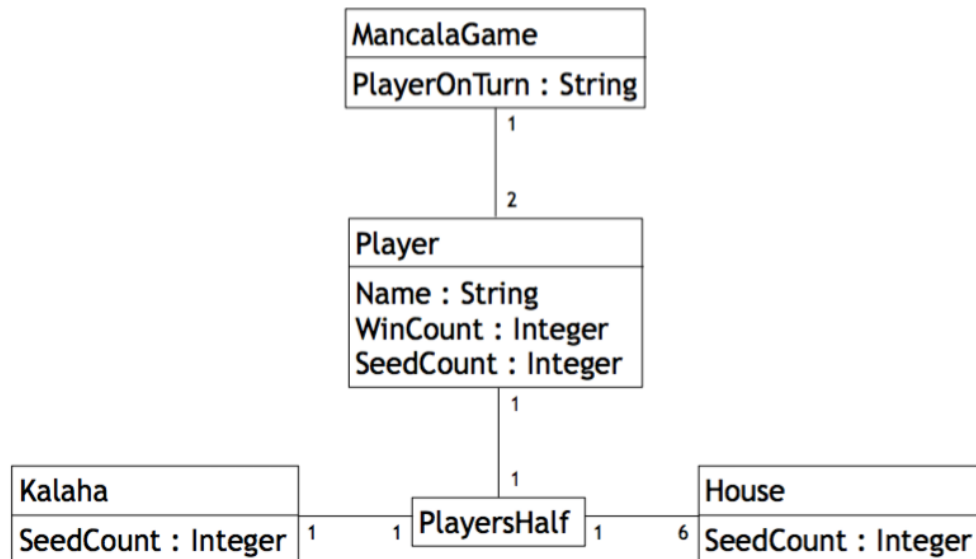
1. Zustandsdaten .....	1
2. Übersetzung nach SVG .....	2
2.1. XSLT-Template .....	2
2.2. SVG parametrisieren .....	2
3. Resultat .....	2

## 1. Zustandsdaten

Um den aktuellen Spielzustand eines Mancala-Spiels zu beschreiben werden diese Daten benötigt:

- Verteilung der Steine: Wie viele Steine pro Mulde und Kalaha
- Spieler am Zug
- Informationen zum Spieler: Name, Anzahl gewonnener Spiele, Anzahl der Steine, die im aktuellen Spiel bereits gewonnen wurden

Als UML-Diagramm dargestellt sehen die Daten zu einem Zustand folgendermaßen aus:



Als DTD lesen sich die Zustandsdaten wie folgt:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT MancalaGame (Player, Player, PlayerOnTurn) >

<!ELEMENT Player (Name, WinCount, PlayersHalf, SeedCount)>
<!ELEMENT Name (#PCDATA) >
<!ELEMENT WinCount (#PCDATA) >
<!ELEMENT SeedCount (#PCDATA) >
```

```
<!ELEMENT PlayersHalf (Kalah, House, House, House, House, House, House) >
<!ELEMENT Kalah (SeedCount) >
<!ELEMENT House (SeedCount)>

<!ELEMENT PlayerOnTurn (#PCDATA) >
```

## 2. Übersetzung nach SVG

### 2.1. XSLT-Template

Als Grundlage haben wir das vom Oxygen XML Editor generierte XSLT-Template verwendet:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs"
  version="2.0">

</xsl:stylesheet>
```

### 2.2. SVG parametrisieren

Anschließend haben wir den Code des SVGs aus Übungsblatt 2 übernommen, eingefügt und anschließend parametrisiert.

#### 2.2.1. Verteilung der Spielsteine

Mithilfe der XSLT-Direktiven `xsl:for-each` und `xsl:value-of` wurden die relevanten Daten aus der auf der DTD für Zustandsdaten basierenden XML-Datei ausgelesen. Zusätzlich wurde `xsl:variable name="..." select="position()"` verwendet um eine Zählvariable innerhalb der For-Each-Loop zu haben und auf dieser basierend die Position von Mulden, Kalahas oder Spielern zu erkennen.

Zunächst haben wir pro Mulde und Kalaha die Anzahl der Spielsteine als Text ausgegeben. Zusätzlich, als Schmankerl, haben wir die Anzahl der Spielsteine grafisch visualisiert und dazu in XSLT die Direktiven `xsl:template` und `xsl:call-template` verwendet um ein sich rekursiv selbst aufrufendes Template zu erstellen, das pro Aufruf einen Spielstein in die SVG-Datei einträgt.

#### 2.2.3. Spieler-Daten

Als wir die Spieler-Daten in das SVG einbetten wollten, ist uns aufgefallen, dass wir dieses zunächst noch erweitern mussten. So fügten wir diverse Text-Felder ein und befüllten sie in gewohnter Vorgehensweise mit den relevanten Daten aus der XML-Datei, die den Spielzustand beschreibt.

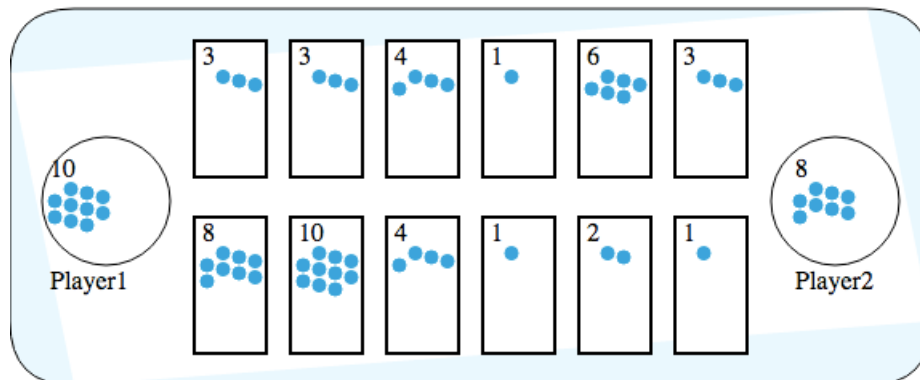
## 3. Resultat

Die schlussendliche Ausgabe unserer Transformation sieht folgendermaßen aus:

Kalaha

Start new Game

Undo Move



Anzahl Gewinne: 5

Anzahl Samen: 6

Anzahl Gewinne: 13

Anzahl Samen: 2

Spieler am Zug: Player1