

Praktikum 3: Rekursion

Aufgabe: Türme von Hanoi

Schreiben Sie einen `THanoiService` für unseren Experimentierkasten, der als Input die Anzahl Scheiben (=Anzahl Plätze pro Turm) annimmt und als Output die notwendigen Züge darstellt um alle Scheiben vom ersten Turm (`turm[0]`) zum zweiten Turm (`turm[1]`) zu verschieben.

Das Programm soll bei jedem Schritt prüfen, ob der betreffende Schritt korrekt ist, und wenn nein eine `THanoiException` werfen.

Hinweis:

- In der Vorlage `THanoi.java` sind die meisten der notwendigen Funktionen schon da; ergänzen müssen Sie nur wenige Zeilen für die eigentliche Rekursion.

Aufgabe 2: Test ob XML Dokument "well-formed" ist

XML Dokumente werden als "well-formed" bezeichnet, wenn Sie u.A. die Eigenschaft erfüllen, dass zu jedem öffnenden Tag `<a>` ein schliessendes Tag `` korrespondiert, mit der Ausnahme der einfachen Tags `<a/>`, die sich selber schliessen. Schreiben Sie eine Erweiterung `CheckWellFormed`, die die korrekte Tag-Schachtelung von XML-Dokumenten überprüft.

```
<dining-room>
  <table>
    <manufacturer>The Wood shop</manufacturer>
    <quality>fair</quality>
  </table>
  <chair>
    <quantity>2</quantity>
    <quality>excellent</quality>
    <cushion>
      <color>blue</color>
    </cushion>
  </chair>
  <chair>
    <quantity>3</quantity>
    <quality>average</quality>
  </chair>
</dining-room>
```

Sie können von den folgenden vereinfachenden Annahmen ausgehen:

- Elemente enthalten keine Attribute (wie obiges Beispiel)
- Tags enthalten keinen Whitespace (u.a. Leerzeichen)
- Die Abkürzung für leere Elemente (`<a/>`) wird nicht gebraucht.

Ein wohlgeformtes XML-Dokument *Faecherliste.xml* finden Sie in den Vorlagen.

Hinweise:

- Achten Sie auf ein gut strukturiertes Programm, indem Sie z.B. eine Klasse *Tag* und einen Stack von Tags definieren.
- Eine Fehlermeldung wie:
Tag "Kapitel" on row 20, column 18 doesn't match Tag "kapitel"
ist sicher besser als
Error at position 352