



1. `?`, `<`, `>`, `"` und `'` haben in XML besondere Bedeutung. Wie werden diese im Text eines XML-Dokumentes abgebildet?

2. Was bedeutet die *DTD-Datendefinition* `#PCDATA`?



3. Ändern Sie die mitgelieferte DTD-Datei `party.dtd` so ab, dass eine Party *mindestens einen Gast* und *mindestens einen Wächter* enthalten muss. Wächter sollen die Eigenschaft `name` haben, die gesetzt werden muss. *Validieren* Sie die Datei mit Eclipse.

4. Tragen Sie dann in die Datei `partywithDTD.xml` den Wächter `Schorsch Schrecklich` ein. *Validieren* Sie die Datei mit Eclipse.

5. Ändern Sie die mitgelieferte Datei `party.xsd` so ab, dass auch sie dafür sorgt, dass *mindestens ein Partygast* und *ein Wächter mit Namen* eingetragen werden müssen.

6. Streichen Sie dann alle Gäste aus der Datei `partywithXSD.xml`, und *validieren* Sie diese. Tragen Sie dann Gäste und Wächter ein. Kontrollieren Sie, ob Sie die Inhalte korrekt eingegeben haben, indem Sie die Datei validieren.

7. Was bedeutet *RSS* und was ist ein *RSS-Feed*? Wie ist ein *RSS-Feed* aufgebaut, aus welchen *Teilen* besteht dieser (**HINWEIS:** <http://www.rssboard.org/>)?



8. *Abonnieren* Sie in Ihrem *Web-Browser* den *RSS-Feed* des *Nachrichtenmagazins Spiegels* (**HINWEIS:**  `RSS`).

9. Betrachten Sie dann folgende *RSS-Feeds*:

<http://www.provinz.bz.it/wetter/rss.asp>

<https://www.suedtirolnews.it/feed>

<http://rss.cnn.com/rss/edition.rss>



10. Sie sollen aus dem vorgestellten einfachen XML-Parser eine *RSS-Reader-Klasse* machen, welche einen *RSS-Feed* einliest und daraus Objekte der mitgelieferten Klassen `Channel` und `Item` aufbaut. Der Parser soll sowohl eine übergebene *XML-Datei* analysieren können als auch einen *RSS-Feed* aus dem *Internet* holen und parsen können. Überlegen Sie sich zuerst, welche *Methoden* der Parser zur Verfügung stellen soll (**HINWEIS:** `URL url = new URL(urlString);`
`InputStream in = url.openStream();`).