Übung 2

1. SAX-Parser für getypte Merkmalsstrukturen (10 Punkte)

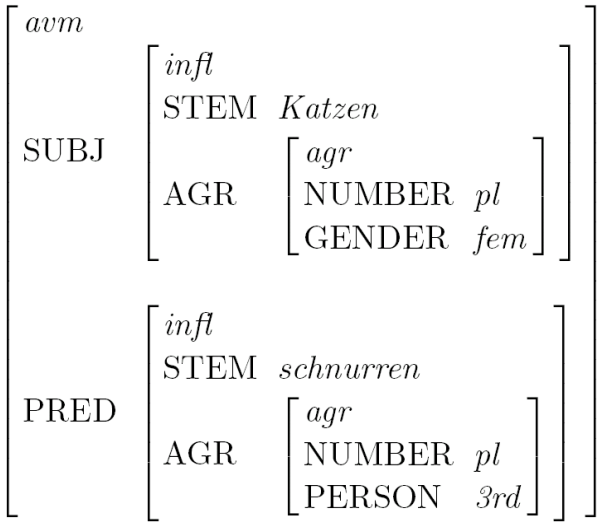
Eine getypte Merkmalsstruktur (eng. typed feature structures; TFS) ist eine rekursiv definierte Datenstruktur: Eine getypte Merkmalsstruktur besitzt genau einen Typ und eine (möglicherweise leere) Liste von Merkmalen (synonym: Attributen) mit jeweils genau einem Wert, der wieder eine getypte Merkmalsstruktur ist. Der Kasten links zeigt eine DTD für getypte Merkmalsstrukturen.

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

<!-- TFS DTD -->

<!ELEMENT FSLIST ( FS )* >
<!ELEMENT FS ( F )* >
<!ATTLIST FS type NMTOKEN #REQUIRED >
<!ELEMENT F  ( FS ) >
<!ATTLIST F  name NMTOKEN #REQUIRED >
```

TFS DTD (XML-Beispiel: siehe S. 2)



Matrix-Darstellung einer getypten Merkmalsstruktur

Gegeben ist in template_ue02.zip eine Objektrepräsentation (Java-Klassen) für getypte Merkmalsstrukturen, bestehend aus zwei Klassen: FeatValPair (Merkmals-Wert-Paare; Merkmal: String, Wert: Merkmalsstruktur) und TypedFeatStruct, welche einen Typ (String) und eine Liste von Merkmals-Wert-Paaren, implementiert als Generic, beinhaltet.

Implementiere einen SAX-Parser, der ein XML-Dokument aus einer Datei, welche eine Liste (FSLIST) von Merkmalsstrukturen (FS) entsprechend der oben angegebenen TFS-DTD enthält (Beispiel-XML s. S. 2; Testdatei: katzen.xml), in eine Liste aus den angegebenen TFS-Objektrepräsentationen überführt.
Hinweis: als Hilfsdatenstrukturen während des Parsens benötigst Du zwei Stacks (verwende die Klasse java.util.Stack) zur Verwaltung der aufgebauten Merkmalsstrukturen und der Merkmale.

Zum Testen können die toString()-Methoden verwendet werden.

Dateien in template_ue02.zip: FeatValPair.java, TypedFeatStruct.java, katzen.xml.

2. Überführung einer Datenstruktur in einen DOM-Baum (5 Punkte)

Implementiere Methoden, welche die Objektrepräsentation aus Aufgabe 1 in einen DOM-Baum entsprechend der TFS-DTD überführen (z.B. durch Erweitern der gegebenen Java-Klassen aus Aufgabe 1). Zum Ausdrucken des erzeugten DOM-Baums (Test, Kontrolle) kann die in der Klasse DOMPrinter bereit gestellte statische printXML-Methode verwendet werden.
Dateien in template_ue02.zip: DOMPrinter.java

```
<?xml version="1.0"?>
<!-- Beispiel zu Aufgabe 1; XML-Datei katzen.xml -->
<!DOCTYPE FSLIST [
  <!ELEMENT FSLIST ( FS )* >
  <!ELEMENT FS ( F )* >
  <!ATTLIST FS type  NMTOKEN #REQUIRED >
  <!ELEMENT F  ( FS ) >
  <!ATTLIST F  name  NMTOKEN #REQUIRED >
]>
<FSLIST>
  <FS type="avm">
     <F name="SUBJ">
       <FS type="infl">
          <F name="STEM">
             <FS type="Katzen"/>
          </F>
          <F name="AGR">
             <FS type="agr">
                <F name="NUMBER">
                   <FS type="pl"/>
                </F>
                <F name="GENDER">
                   <FS type="fem"/>
                </F>
             </FS>
          </F>
       </FS>
     </F>
     <F name="PRED">
       <FS type="infl">
          <F name="STEM">
             <FS type="schnurren"/>
          </F>
          <F name="AGR">
             <FS type="agr">
                <F name="NUMBER">
                   <FS type="pl"/>
                </F>
                <F name="PERSON">
                   <FS type="3rd"/>
                </F>
             </FS>
          </F>
       </FS>
     </F>
  </FS>
</FSLIST>
```