

Aufgabe 1

Data Dictionary

```

Nummernliste = {Telefonnummer}
Telefonnummer = [(Provider) + (["*" + Landesvorwahl | "0" + Ortsvorwahl) + Hauptanschluss + (Durchwahl) |
(Provider) + Handylvorwahl + Handynummer |
Mehrwertdienst + Nummer]
Ziffer = [0|1|2|3|4|5|6|7|8|9] /* der Einfachheit halber */
Provider = "010" + 2{Ziffer}2
Landesvorwahl = 1{Ziffer}4
Ortsvorwahl = 2{Ziffer}6
Nummer = 4{Ziffer}9 /* generische Telefonnummer */
Hauptanschluss = 3{Ziffer}4
Durchwahl = "-" + 2{Ziffer}4
Handylvorwahl = 3{Ziffer}4
Handynummer = Nummer
Mehrwertdienst = [0700 | 0800 | 0180 | 0190]

```

Einige Entscheidungen und Vereinfachungen, die hier getroffen wurden: Der Handynummer wurde noch eine Vorwahl hinzugefügt, die Längen der jeweiligen Nummern basieren eher auf Erfahrung und Beobachtung als auf tatsächlichen Begebenheiten, können daher also abweichen. Theoretisch wären auch Handynummern mit der ersten Auswahl einer Telefonnummer abgedeckt, einfach aus dem Grunde, weil auch Handynummern eine Ländervorwahl haben können, etc.

Auf die explizite Spezifikation von Telefonnummern, die nur aus einer Durchwahl bestehen, wurde hier bewußt verzichtet; ebenso auf den Ausschluß von Telefonnummern, die eine 555 enthalten oder die explizite Angabe von Notrufnummern.

Ich denke, an sich ist das Ganze halbwegs selbsterklärend und wurde so zum Teil auch schon in der Übung gehandhabt.

Document Type Definition

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT Nummernliste (Telefonnummer*)>
<!ELEMENT Telefonnummer
((Provider?, Landesvorwahl?, Ortsvorwahl?, Hauptanschluss, Durchwahl?) |
(Provider?, Handylvorwahl, Handynummer) |
(Mehrwertdienst, Nummer))>
<!ELEMENT Provider (#PCDATA)>
<!ELEMENT Landesvorwahl (#PCDATA)>
<!ELEMENT Ortsvorwahl (#PCDATA)>
<!ELEMENT Hauptanschluss (#PCDATA)>
<!ELEMENT Durchwahl (#PCDATA)>
<!ELEMENT Handylvorwahl (#PCDATA)>
<!ELEMENT Handynummer (#PCDATA)>
<!ELEMENT Mehrwertdienst (#PCDATA)>
<!ELEMENT Nummer (#PCDATA)>

```

Hier wurden sämtliche Terminalsymbole notwendigerweise als #PCDATA gekennzeichnet, da eine DTD für Elemente keine feinere Beschränkung des Inhaltes erlaubt (ich wollte hier nun nicht einen Tag für eine einzelne Ziffer einführen ...). Eine DTD beherrscht auch einige andere Merkmale eines DD nicht, daher werden hier wohl einige Dinge etwas expliziter aussehen müssen als im DD. (Abgesehen davon, daß das XML-Format in dieser Form in meinen Augen eher Verschwendung von Markup darstellt, als irgendwas sinnvolles.)

Alternativen wären einerseits die Verwendung von Attributen (nicht in der Vorlesung behandelt), was beispielsweise für Elemente wie *Mehrwertdienst* die Möglichkeit bieten würde, den Wertebereich zu beschränken.

Oder aber man schreibt das Ganze nicht als DTD, sondern als XML Schema, was allerdings ebenfalls (noch) nicht behandelt wurde.

Weiterhin hätte man hier statt mit Sequenzen (,) auch mit Alternativen (|) arbeiten können, um die Reihenfolge der Unterelemente nicht fest vorzugeben.

Beispiel

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Nummernliste SYSTEM "nummernliste.dtd">
<Nummernliste>
  <Telefonnummer>
    <Ortsvorwahl>0381</Ortsvorwahl>
    <Hauptanschluss>12555678</Hauptanschluss>
  </Telefonnummer>
  <Telefonnummer>
    <Provider>01013</Provider>
    <Handyvorwahl>0171</Handyvorwahl>
    <Handynummer>349785554</Handynummer>
  </Telefonnummer>
  <Telefonnummer>
    <Mehrwertdienst>0800</Mehrwertdienst>
    <Nummer>235285553</Nummer>
  </Telefonnummer>
  <!-- Nun noch das obligatorische Beispiel,
        daß die DTD weder die komplette Syntax
        noch Semantik abdeckt -->
  <Telefonnummer>
    <Provider>Blubb</Provider>
    <Landesvorwahl>#?&</Landesvorwahl>
    <Hauptanschluss>Hier steht ein toller Text</Hauptanschluss>
  </Telefonnummer>
</Nummernliste>
```

Aufgabe 2

```
Student      = Name + 1{Vorname}3 + Geburtsdatum + Geburtsort + Wohnort
              + (Studienort) + 1{Studienfach} + {Nebenfach} + (Vordiplom)
Datum        = Tag + Monat + Jahr
Geburtsdatum = Datum
Vordiplom    = Datum + Note
Geburtsort   = Straße + Hausnummer + Ort + PLZ + 1{Telefonnummer}
Studienort  = Straße + Hausnummer + Ort + {Telefonnummer}
```

Ich habe hier mal davon abgesehen, die einzelnen Symbole weiter aufzuschlüsseln als nötig. Das meiste sind ohnehin Zeichenketten oder Zahlen also hielt ich das nicht für sonderlich notwendig. Studien- oder Nebenfächer als Auswahl zu definieren würde zwar gehen, ist aber meines Erachtens nach nicht der Sinn eines DD.

Aufgabe 3

```
<?xml version="1.0"?>
<!-- Einige allgemeine Entities, die wir wohl
      mehr als einmal brauchen werden -->
<!ENTITY % dateattr
"tag          CDATA #REQUIRED
monat         CDATA #REQUIRED
jahr          CDATA #REQUIRED"
>
<!ENTITY % locattr
"strasse      CDATA #REQUIRED
hausnummer   CDATA #ERQUIRED
plz           CDATA #REQUIRED
ort           CDATA #REQUIRED"
```

```

>
<!ELEMENT Studenten (Student*)>
<!ELEMENT Student (Name, (Vorname | Vorname, Vorname |
                        Vorname, Vorname, Vorname),
                        Geburtsdatum, Geburtsort, Wohnort, Studienort?,
                        Studienfach+, Nebenfach*, Vordiplom?)>
<!--ELEMENT Name (#PCDATA)-->
<!--ELEMENT Vorname (#PCDATA)-->
<!--ELEMENT Geburtsdatum EMPTY-->
<!--ATTLIST Geburtsdatum %dateattr;-->
<!--ELEMENT Wohnort (Telefonnummer+)-->
<!--ATTLIST Wohnort %locattr;-->
<!--ELEMENT Telefonnummer (#PCDATA)-->
<!--ELEMENT Studienort (Telefonnummer*)-->
<!--ATTLIST Studienort %locattr;-->
<!--ELEMENT Studienfach (#PCDATA)-->
<!--ELEMENT Nebenfach (#PCDATA)-->
<!--ELEMENT Vordiplom EMPTY-->
<!--ATTLIST Vordiplom %dateattr; note CDATA #REQUIRED-->

```

Ich habe hier mit Entities und Attributen gearbeitet. Der Stil orientiert sich ein wenig an den W3C-DTDs. Ich habe, wo es mir sinnvoll erschien, Attribute eingesetzt, da der Schreiboverhead für Elemente in meinen Augen recht hoch ist – und wenn ein Attribut den gleichen Zweck erfüllt ...

Aufgabe 4

Hier sei einmal angenommen, daß keine Nummernliste gemeint ist sondern eher eine Studentenliste ...

```

<?xml version="1.0"?>
<!DOCTYPE Studenten SYSTEM "studenten.dtd">
<Studenten>
  <Student>
    <Name>Mustermann</Name>
    <Vorname>Max</Vorname>
    <Vorname>Moritz</Vorname>
    <Geburtsdatum jahr="1980" monat="12" tag="12"/>
    <Geburtsort>Musterstadt</Geburtsort>
    <Wohnort strasse="Steinstraße" hausnummer="18" plz="18055" ort="Rostock">
      <Telefonnummer>0381 149855524</Telefonnummer>
    </Wohnort>
    <Studienfach>Informatik</Studienfach>
    <Nebenfach>Mathematik</Nebenfach>
    <Vordiplom jahr="2005" monat="5" tag="14" note="2.3"/>
  </Student>
  <Student>
    <Name>Schultze</Name>
    <Vorname>Erika</Vorname>
    <Geburtsdatum jahr="1981" monat="1" tag="25"/>
    <Geburtsort>Güstrow</Geburtsort>
    <Wohnort strasse="Tralalaweg" hausnummer="8" plz="12345" ort="Musterstadt">
      <Telefonnummer>0123 149855524</Telefonnummer>
      <Telefonnummer>0123 323455624</Telefonnummer>
    </Wohnort>
    <Studienort strasse="Kurt-Schumacher-Ring" hausnummer="207" plz="18109" ort="Rostock">
    </Studienort>
    <Studienfach>Wirtschaftsinformatik</Studienfach>
  </Student>
  <Student>
    <Name>Kunz</Name>
    <Vorname>Hans</Vorname>
    <Geburtsdatum jahr="1981" monat="7" tag="15"/>
    <Geburtsort>Schwerin</Geburtsort>
    <Wohnort strasse="Ulmenstr. 69" hausnummer="2" plz="18057" ort="Rostock">
      <Telefonnummer>0381 2001678</Telefonnummer>
    </Wohnort>
    <Studienfach>Wirtschaftsinformatik</Studienfach>
  </Student>
</Studenten>

```

Aufgabe 5

