Solution: Exercise-6

Q1:

count(//file)	3
count(//directory)	3
count(/directory)	0
count(/root)	1
count(node()/directory)	2
count(//@*)	14
count(//*)	10
count(//linkS)	2
count(//linkC)	1
count(//@name)	7
string(/)	„“
//@name	{„root“, „verzeichnis1“, „file1“, „dir1“, „verzeichnis2“, „file1“, „file2“}
boolean(//directory/@name)	true
boolean(//linkS/@name)	false
node-name(//file/parent::node())	<dieser Xpath gibt einen Fehler!>
node-name(//linkC/parent::node())	„root“

```
<?xml version="1.1"?>
<root partitionID="1" name="root">
<directory id="1" name="verzeichnis1">
<file name="file1"></file>
</directory>
<directory id="2" name="dir1">
<directory id="3" name="verzeichnis2">
<file name="file1"></file>
</directory>
<file name="file2"></file>
</directory>
<linkC id="1"></linkC>
<linkS id="2"></linkS>
<linkS id="3"></linkS>
</root>
```

Q2: Develop an XML application for a flight company. The data must be expressed in XML, so as to ensure compatibility with other flight companies.

An international flight company operates flights between several airports. A passenger has a name, an address and a passport number. When a passenger books a ticket, the flight company registers a credit card number. A flight has an id, number of seats, and a date. Each airport is characterised by its name, code, and tax value in dollars. Create an application which is able to retrieve all reservations on a given date, the passengers flying through a certain airport on a given date

2.1

```
<?xml version="1.0" encoding="UTF-8"?>
<doc>
   <Airport airId="LHR">
     <name>London Heathrow</name>
     <tax>100</tax>
   </Airport>

   <Airport airId="ZRH">
     <name>Zurich</name>
     <tax>150</tax>
   </Airport>

   <Flight flightId="LX123">
    <seats>100</seats>
    <date>2005-08-09</date>
    <source>LHR</source>
    <destination>ZRH</destination>
   </Flight>

   <Passenger>
    <name>Student</name>
    <passportnumber>123456</passportnumber>
    <address>Univstr. 6</address>
   </Passenger>

   <Reservation>
     <flightRef>LX123</flightRef>
     <passRef>123456</passRef>
     <creditCard>1234 5678</creditCard>
     </Reservation>
</doc>
```

2.2 Using a DTD or Schema.

2.3

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<!-- Passenger Element -->
<xs:element name="Passenger">
  <xs:complexType>
     <xs:sequence>
       <xs:element name="name" type="xs:string"/>
       <xs:element name="passportnumber" type="xs:string"/>
       <xs:element name="address" type="xs:string"/>
    </xs:sequence>
   </xs:complexType>
</xs:element>

<!-- Flight Element -->
<xs:element name="Flight">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="seats">
            <xs:simpleType>
              <xs:restriction base="xs:int">
                 <xs:minExclusive value="30"/>
                 <xs:maxInclusive value="1000"/>
              </xs:restriction>
             </xs:simpleType>
          </xs:element>

          <xs:element name="date" type="xs:date"/>
          <xs:element name="source" type="xs:string"/>
          <xs:element name="destination" type="xs:string"/>
     </xs:sequence>
     <xs:attribute name="flightId">
       <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:minLength value="0"/>
            <xs:maxLength value="5"/>
         </xs:restriction>
       </xs:simpleType>
     </xs:attribute>
  </xs:complexType>
```

```
</xs:element>


<!-- Airport Element -->
<xs:element name="Airport">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="tax" type="xs:float"/>
    </xs:sequence>
    <xs:attribute name="airId">
      <xs:simpleType>
        <xs:restriction base="xs:string">
           <xs:minLength value="0"/>
           <xs:maxLength value="3"/>
       </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>

<!-- Root element of the document.-->
<xs:element name="doc">
 <xs:complexType>
   <xs:sequence>
     <xs:element ref="Airport" minOccurs="2" maxOccurs="unbounded" />
     <xs:element ref="Flight" minOccurs="1" maxOccurs="unbounded" />
     <xs:element ref="Passenger" minOccurs="1" maxOccurs="unbounded" />
   </xs:sequence>
 </xs:complexType>
</xs:element>
<!-- root -->
</xs:schema>
```