Information Systems 2

Prof. Dr. Dr. L. Schmidt-Thieme
MSc. André Busche

Übung 9

0. Allerlei

1. Übung

2. Hands on some things

2.1 Saxon

2.2 Corba

0. Allerlei

1. Übung

2. Hands on some things

2.1 Saxon

2.2 Corba

Software (XML Editoren)

Altova XML Spy
http://www.altova.com/

Oxygen XML Editor
http://www.oxygenxml.com

(WTP Eclipse-Plugins)

… und sehr, sehr viele andere!

Ein guter Editor …

- validiert nach DTD/XML Schema

- Erlaubt „point-and-click“ XSLT Transformationen/XQuery anfragen

- Unterstützt verschiedene XML Parser/Prozessoren

0. Allerlei

1. Übung

2. Hands on sme things

2.1 Saxon

2.2 Corba

Aufgabe 1a)

Zeigen Sie 2 selbstkonstruierte Beispiele für mögliche praktische Einsätze von XQuery! Geben Sie an, was für Informationen Sie in den XML-Dokumenten speichern, und welche Informationen Sie anfragen!

Wahlfreie Antwort ...

XML: Speicherung von Flügen
XQuery: Anfrage von/nach/wann/etc.

XML: Speicherung von Kursen und deren Beziehungen (Vorbedingungen, etc.)
XQuery: Gegeben Vorkenntnisse X1, X2, X3, List verfügbare Kurse in Format Y auf..

Aufgabe 1b)

```
<?xml version="1.0" encoding="UTF-8"?>
<project>
   <task> <description>Build up the tent</description>
      <responsible>Peter</responsible>
      <startsAt>10:00</startsAt>
      <endsAt>12:00</endsAt> </task>
   <task> <description>Transport the food for lunch</description>
      <responsible>Anna</responsible>
      <startsAt>09:30</startsAt>
      <endsAt>10:30</endsAt> </task>
   <task> <description>Prepair the lunch</description>
      <responsible>Tomas</responsible>
      <startsAt>11:00</startsAt>
      <endsAt>12:00</endsAt> </task>
   <task> <description>Distribute lunch</description>
      <responsible>Tomas, Anna</responsible>
      <startsAt>12:00</startsAt>
      <endsAt>13:00</endsAt> </task>
   <task> <description>Make a concert</description>
      <responsible>Peter</responsible>
      <startsAt>13:30</startsAt>
      <endsAt>15:00</endsAt> </task>
   <task> <description>Collect rubish</description>
      <responsible>UNASSIGNED</responsible>
      <startsAt>15:00</startsAt>
      <endsAt>17:00</endsAt> </task>
</project>
```

Aufgabe 1b -1)

```
for $x in /project/task
where ($x/startAt="13:30")
return $x/responsible
```

Aufgabe 1b -1)

```
<?xml version="1.0" encoding="UTF-8"?>
<project>
  <task> <description>Build up the tent</description>
    <responsible>Peter</responsible>
    <startsAt>10:00</startsAt>
    <endsAt>12:00</endsAt> </task>
  <task> <description>Transport the food for lunch</description>
    <responsible>Anna</responsible>
    <startsAt>09:30</startsAt>
    <endsAt>10:30</endsAt> </task>
  <task> <description>Prepair the lunch</description>
    <responsible>Tomas</responsible>
    <startsAt>11:00</startsAt>
    <endsAt>12:00</endsAt> </task>
  <task> <description>Distribute lunch</description>
    <responsible>Tomas, Anna</responsible>
    <startsAt>12:00</startsAt>
    <endsAt>13:00</endsAt> </task>
  <task> <description>Make a concert</description>
    <responsible>Peter</responsible>
    <startsAt>13:30</startsAt>
    <endsAt>15:00</endsAt> </task>
  <task> <description>Collect rubish</description>
    <responsible>UNASSIGNED</responsible>
    <startsAt>15:00</startsAt>
    <endsAt>17:00</endsAt> </task>
</project>
```

```
for $x in /project/task
where
($x/startAt="13:30")
```

Welche(r) Knoten ist/sind an $x gebunden?

Welche(r) Knoten ist/sind an $x gebunden?

KEINER!

```
<task> <description>Make a concert</description>
       <responsible>Peter</responsible>
       <startsAt>13:30</startsAt>
       <endsAt>15:00</endsAt> </task>
```

<startsAt>

```
for $x in /project/task
where ($x/startAt="13:30")
```

Aufgabe 1b -2)

```
<html><body><table>
{ for $x in /project/task
order by $x/description
return <tr><td>{data($x/description)}</td>
    <td>{data($x/startsAt)}</td>
    <td>{data($x/endsAt)}</td>
}
</table></body></html>
```

Ist die Query richtig?

Aufgabe 1b -2)

```
<html><body><table>
{ for $x in /project/task
order by $x/description
return <tr><td>{data($x/description)}</td>
    <td>{data($x/startsAt)}</td>
    <td>{data($x/endsAt)}</td>
}
</table></body></html>
```

NEIN!

</tr>

Aufgabe 1b-2)

```
<html><body><table>
{   for $x in /project/task
    order by $x/description
    return <tr>
        <td>{data($x/description)}</td>
        <td>{data($x/startsAt)}</td>
        <td>{data($x/endsAt)}</td></tr> },
</table></body></html>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
   <body>
      <table><tr>
            <td>Build up the tent</td>
            <td>10:00</td>
            <td>12:00</td>
         </tr><tr>
            <td>Collect rubish</td>
            <td>15:00</td>
            <td>17:00</td>
         </tr><tr>
            <td>Distribute lunch</td>
            <td>12:00</td>
            <td>13:00</td>
         </tr><tr>
            <td> Make a concert</td>
            <td>13:30</td>
            <td>15:00</td>
         </tr><tr>
            <td>Prepair the lunch</td>
            <td>11:00</td>
            <td>12:00</td>
         </tr><tr>
            <td>Transport the food for lunch</td>
            <td>09:30</td>
            <td>10:30</td>
         </tr></table>
   </body>
</html>
```

Ergebnis

Und warum?

Aufgabe 1b)

```
<?xml version="1.0" encoding="UTF-8"?>
<project>
  <task> <description>Build up the tent</description>
    <responsible>Peter</responsible>
    <startsAt>10:00</startsAt>
    <endsAt>12:00</endsAt> </task>
  <task> <description>Transport the food for lunch</description>
    <responsible>Anna</responsible>
    <startsAt>09:30</startsAt>
    <endsAt>10:30</endsAt> </task>
  <task> <description>Prepair the lunch</description>
    <responsible>Tomas</responsible>
    <startsAt>11:00</startsAt>
    <endsAt>12:00</endsAt> </task>
  <task> <description>Distribute lunch</description>
    <responsible>Tomas, Anna</responsible>
    <startsAt>12:00</startsAt>
    <endsAt>13:00</endsAt> </task>
  <task> <description>Make a concert</description>
    <responsible>Peter</responsible>
    <startsAt>13:30</startsAt>
    <endsAt>15:00</endsAt> </task>
  <task> <description>Collect rubish</description>
    <responsible>UNASSIGNED</responsible>
    <startsAt>15:00</startsAt>
    <endsAt>17:00</endsAt> </task>
</project>
```

```
for $x in /project/task
  order by $x/description
```

Welche(r) Knoten ist/sind an $x gebunden?

Welche(r) Knoten ist/sind
an $x gebunden?

ALLE <task>!

… und was machen wir damit?

```
return <tr>
        <td>{data($x/description)}</td>
        <td>{data($x/startsAt)}</td>

        <td>{data($x/endsAt)}</td></tr>
```

Also ...

```
<?xml version="1.0" encoding="UTF-8"?>
<html>
  <body>
    <table><tr>
        <td>Build up the tent</td>
        <td>10:00</td>
        <td>12:00</td>
      </tr><tr>
        <td>Collect rubish</td>
        <td>15:00</td>
        <td>17:00</td>
      </tr><tr>
        <td>Distribute lunch</td>
        <td>12:00</td>
        <td>13:00</td>
      </tr><tr>
        <td> Make a concert</td>
        <td>13:30</td>
        <td>15:00</td>
      </tr><tr>
        <td>Prepair the lunch</td>
        <td>11:00</td>
        <td>12:00</td>
      </tr><tr>
        <td>Transport the food for lunch</td>
        <td>09:30</td>
        <td>10:30</td>
      </tr></table>
  </body>
</html>
```

Aufgabe 1c)

Schreiben Sie eine XQuery-Anfrage, mit dem Sie jene Aufgaben listen, die zu keiner Person zugeordnet sind!

„keiner Person“ === „UNASSIGNED“

„keiner Person“ === kein <responsible>-Objekt?

```
for $ x in
    //task[responsible/text()="UNASSIGNED"]
return $x
```

```
for $ x in
    //task[not(exists(responsible))]
return $x
```

```
for $ x in
    //task[string(responsible)=""]
return $x
```

Aufgabe 1c)

die Beschreibung der Aufgabe, wann die Aufgabe anfängt, bis wann sie dauert!

```
for $ x in
    //task[responsible/text()="UNASSIGNED"]
return <r>{$x/description,
          $x/startsAt,
          $x/endsAt}
          </r>
```

```
for $ x in
//task[responsible="UNASSIGNED"]
return concat(string($x/description), string($x/startsAt), string($x/endsAt))
```

Aufgabe 2)

Schreiben Sie eine XQuery-Anfrage, welche als Ergebnis die Namen und Preisen von Produkten liefert. Die Produkte sollen dabei nach dem Preis aufsteigend sortiert werden.

```
<?xml version="1.1"?>
<products>
   <product name="Bread" price="1.20"/>
   <product name="Eggs" quantity="10" price="1.49"/>
   <product name="Bio-Eggs" quantity="6" price="1.29"/>
   <product name="Milk" price="0.60"/>
   <product name="Orange Juice 50%" price="0.59"/>
   <product name="Orange Juice 100%" price="0.89"/>
</products>
```

Aufgabe 2a)

```
<?xml version="1.1"?>
<products>
   <product name="Bread" price="1.20"/>
   <product name="Eggs" quantity="10" price="1.49"/>
   <product name="Bio-Eggs" quantity="6" price="1.29"/>
   <product name="Milk" price="0.60"/>
   <product name="Orange Juice 50%" price="0.59"/>
   <product name="Orange Juice 100%" price="0.89"/>
</products>
```

```
for $x
in //product
order by
    $x/@price
    return $x
```

Aufgabe 2b)

Jene Produkte, die teurer, als 1,00 sind, sollen zwischen den Tags „<teuer_produkt>“ bzw. </teuer_produkt> stehen

```
for $x
in //product
order by
    $x/@price
    return
        if($x/@price > 1.00)
        then
        <teuer>
        {string($x/@name)}
        </teuer>
        else
        <billig/>
```

```
<?xml version="1.1"?>
<products>
   <product name="Bread" price="1.20"/>
   <product name="Eggs" quantity="10" price="1.49"/>
   <product name="Bio-Eggs" quantity="6" price="1.29"/>
   <product name="Milk" price="0.60"/>
   <product name="Orange Juice 50%" price="0.59"/>
   <product name="Orange Juice 100%" price="0.89"/>
</products>
```

0. Allerlei

1. Übung

2. Hands on sme things

2.1 Saxon

2.2 Corba

Software (XML Parser und XML Prozessoren)

… und wer keinen XML Editor hat, geht den Weg zu Fuß …

… z. B. Mit Saxon (www.saxonica.com)
→ aktuelle Version 9.2 (vom 26.4.2010)

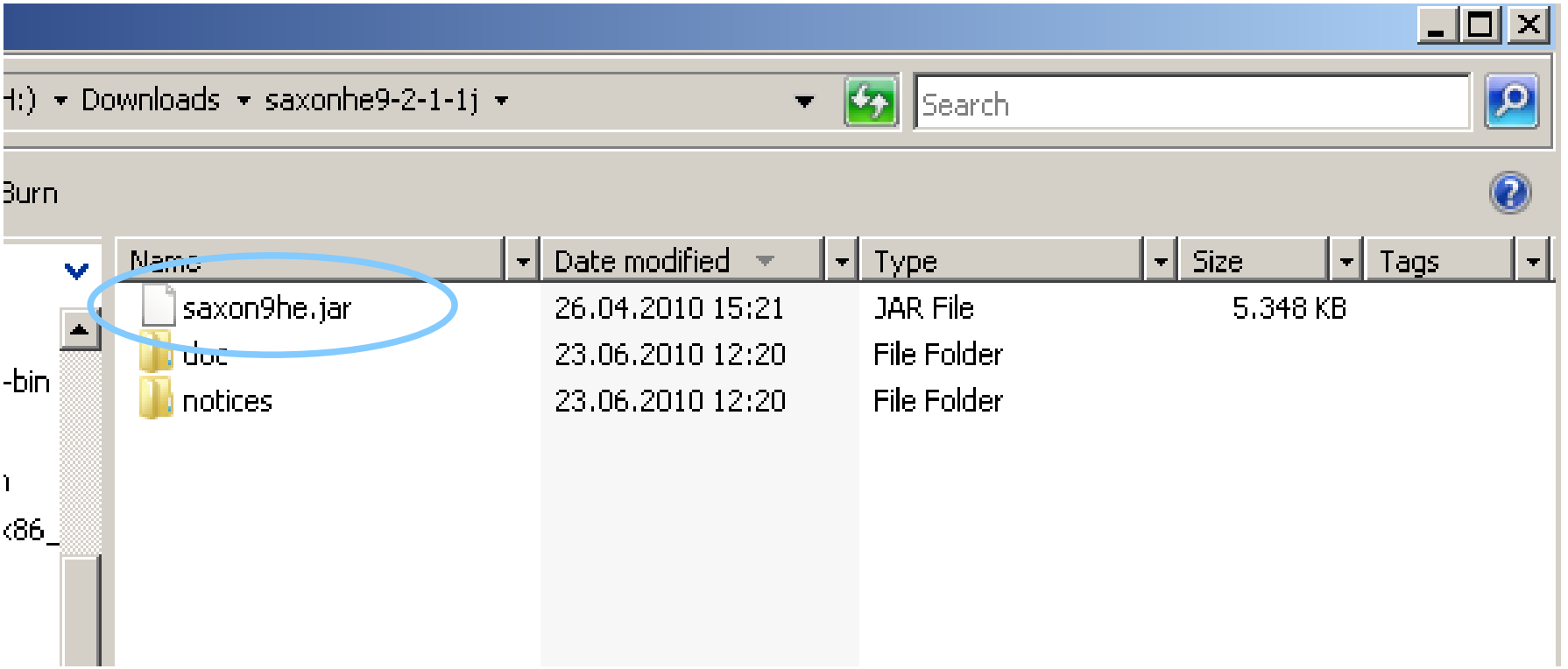
Document:

```
<?xml version="1.1"?>
<test>
   <p>Hallo, Welt!</p>
   <div>Wie geht es dir?</div>
</test>
```

XQuery:

```
//p
```

Nach dem Download von Saxon...



Kann man Saxon benutzen ;-)

```
H:\saxon>java -cp .;saxon9he.jar net.sf.saxon.Query\
     -q:query1.xquery -s:test.xml
<?xml version="1.0" encoding="UTF-8"?>
<p>Hallo, Welt!</p>
H:\saxon>
```

Saxon als Script (Windows bat-file):

```
@echo off
set SAXON=h:\saxon
java -cp .;%SAXON%\saxon9he.jar net.sf.saxon.Query %*
```

Saxon als Script ausführen:

```
H:\is2-tut>saxon -qs:"//div" -s:test.xml
```

… und 'rumspielen ...

Vorbedingung:
- saxon[.bat] muss im PATH liegen
- java.exe auch

0. Allerlei

1. Übung

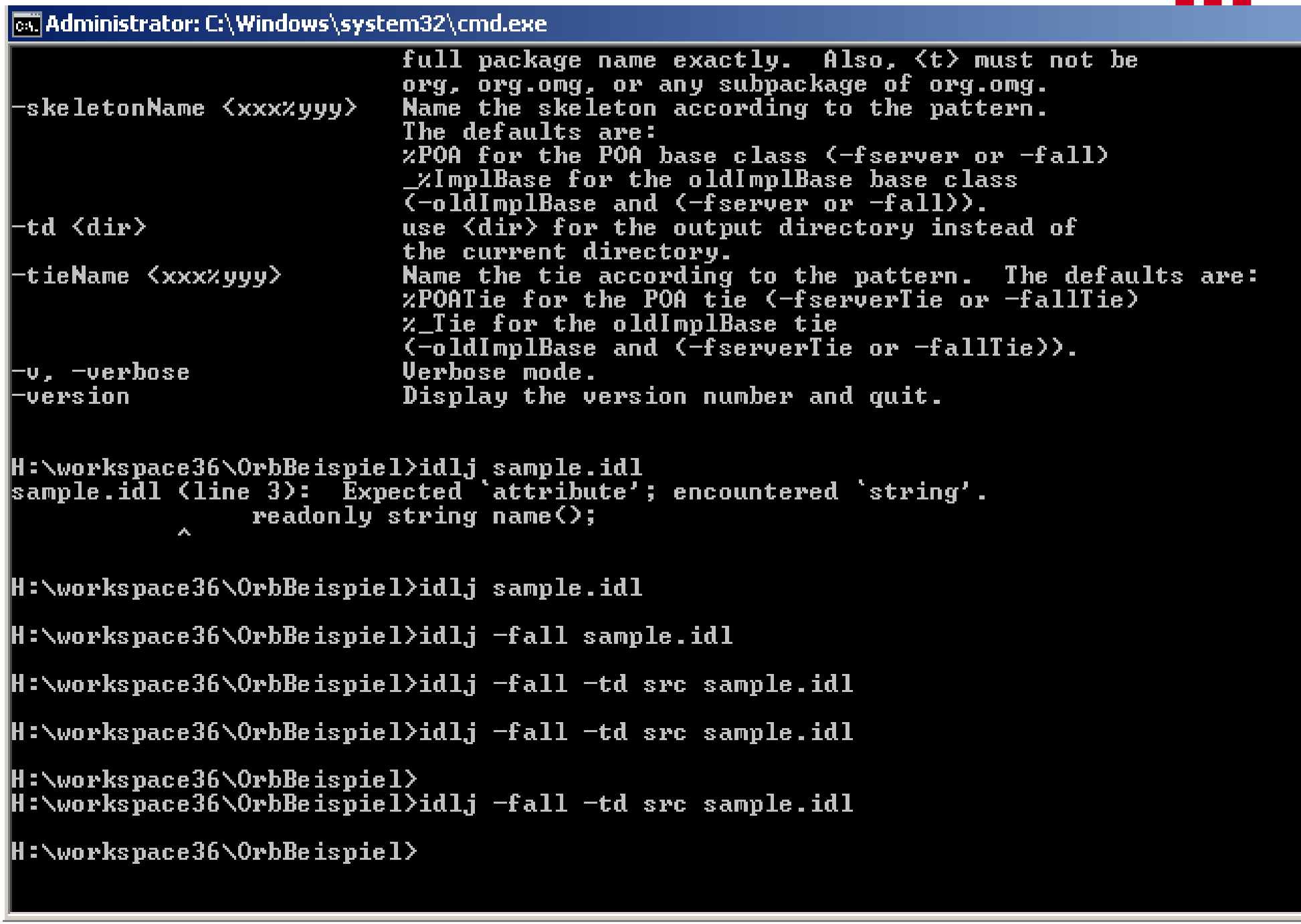
2. Hands on sme things

2.1 Saxon

2.2 Corba

Corba


```
idlj -fall sample.idl
```


Administrator: C:\Windows\system32\cmd.exe
full package name exactly. Also, <t> must not be
org, org.omg, or any subpackage of org.omg.
-skeletonName <xxx%yyy> Name the skeleton according to the pattern.
The defaults are:
%POA for the POA base class (-fserver or -fall)
_%ImplBase for the oldImplBase base class
(-oldImplBase and (-fserver or -fall)).
-td <dir> use <dir> for the output directory instead of
the current directory.
-tieName <xxx%yyy> Name the tie according to the pattern. The defaults are:
%POATie for the POA tie (-fserverTie or -fallTie)
%_Tie for the oldImplBase tie
(-oldImplBase and (-fserverTie or -fallTie)).
-v, -verbose Verbose mode.
-version Display the version number and quit.
H:\workspace36\OrbBeispiel>idlj sample.idl
sample.idl (line 3): Expected `attribute'; encountered `string'.
readonly string name();
^
H:\workspace36\OrbBeispiel>idlj sample.idl
H:\workspace36\OrbBeispiel>idlj -fall sample.idl
H:\workspace36\OrbBeispiel>idlj -fall -td src sample.idl
H:\workspace36\OrbBeispiel>idlj -fall -td src sample.idl
H:\workspace36\OrbBeispiel>
H:\workspace36\OrbBeispiel>idlj -fall -td src sample.idl
H:\workspace36\OrbBeispiel>

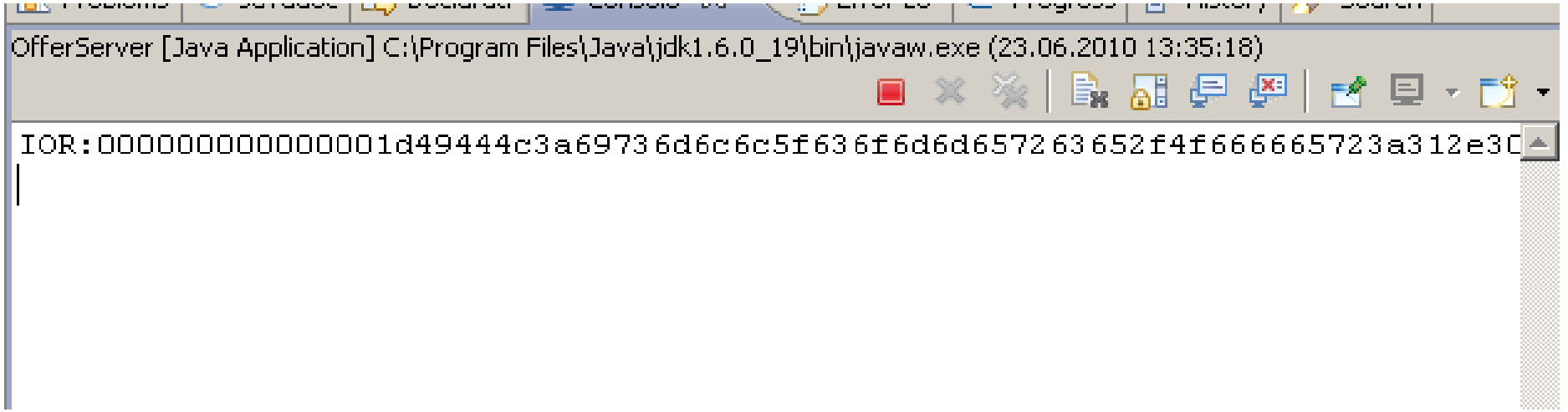
Corba


OrbBeispiel
src
client
ismll_commerce
server
JRE System Library [jdk1.6.0_19]
sample.idl

termine.txt | pruefungsfragen.txt | DummyWebService.java | OfferClient.java

```
package client;

import ismll_commerce.Offer;

public class OfferClient {
    public static void main(String args[]) {
        try {
            // a. connect to the ORB infrastructure:
            String[] argsOrb = new String[] { "-ORBInitialPort", "9090",
                    "-ORBInitialHost", "localhost" };
            ORB orb = ORB.init(argsOrb, null);

            // b. retrieve application object by reference (here: args[0]
            // command line):
            String refString = args[0];
            org.omg.CORBA.Object ref = orb.string_to_object(refString);
            Offer offer_pc = OfferHelper.narrow(ref);

            // c. do something with them:
            System.out.println("Obtained a handle on server object: "
                    + offer_pc);
            System.out.println("name: " + offer_pc.name());
            System.out.println("price: " + offer_pc.price());
        } catch (Exception e) {
            System.out.println("ERROR : " + e);
            e.printStackTrace(System.out);
        }
    }
}
```

Start Server



Start Client