Schemas

Introduction
XSD Elements and Attributes
XSD Data Types
XSD Facets
XSD Indicators
Extending XSD

Objectives

- **Need for Schema**
- **What is an XML Schema?**
- **Exploring XSD**
- **How to write XML Schema**
- **Structuring**
  - XSD files
  - Apply to XML files
  - XSD Elements
  - XSD Simple Types Element
  - XSD Data Types
  - XSD Attributes

Objectives

- **Structuring (cont)**
  - XSD Facets
  - XSD Complex Types Element
  - XSD Indicators
  - Extending Complex Types
  - Abstract Types
  - XSD Any Elements
  - XSD Element Substitution
- **Summary**

Schemas
Need for Schema

- The **limitation** of DTDs
  - DTDs are **written** in a **non-XML syntax**
    - DTDs do not use XML notation (they **use EBNF** to present)
    - The EBNF is **difficult** to **write and use**
  - DTDs **do not support namespace**
    - A namespace cannot be used to refer to an element or an entity declaration
    - If the namespace is used, the DTD has to be modified to include any elements taken from the namespace
  - DTDs **do not provide** a means of specifying element and attribute **data types**
    - DTD can only express the data type of attributes in term of explicit enumeration and a few coarse string formats
    - DTDs do not have a facility to describe numbers, dates, etc. …

→ **Schema** is a **candidate** that specially **aimed** at addressing **DTDs' limitation**

Schemas

What is an XML Schema?

- An XML Schema **specifies** the **structure** of **valid XML documents**
  - **Defining a set of elements**, **their relationships** to each other, **and** the **attributes** that they can contain
- The **objective** of XML Schema is the **same purpose with DTD**
  - **Validating XML data = XML document + XML Schema**
  - A schema defines the **valid building block** of an XML document
  - A schema can be considered as **a common vocabulary** that different organizations **can share** to exchange document
- A schema is **considered** a **valid XML document** because the **schema specification** is **defined** using a **DTD**
- An XML Schema defines
  - **Elements** and **Attributes**
  - **Child elements** with the **order** and **number** of child elements
  - Whether an element is an **empty** and can **include text**
  - **Data types** for elements and attributes
  - **Default** and **fixed** values for elements and attributes

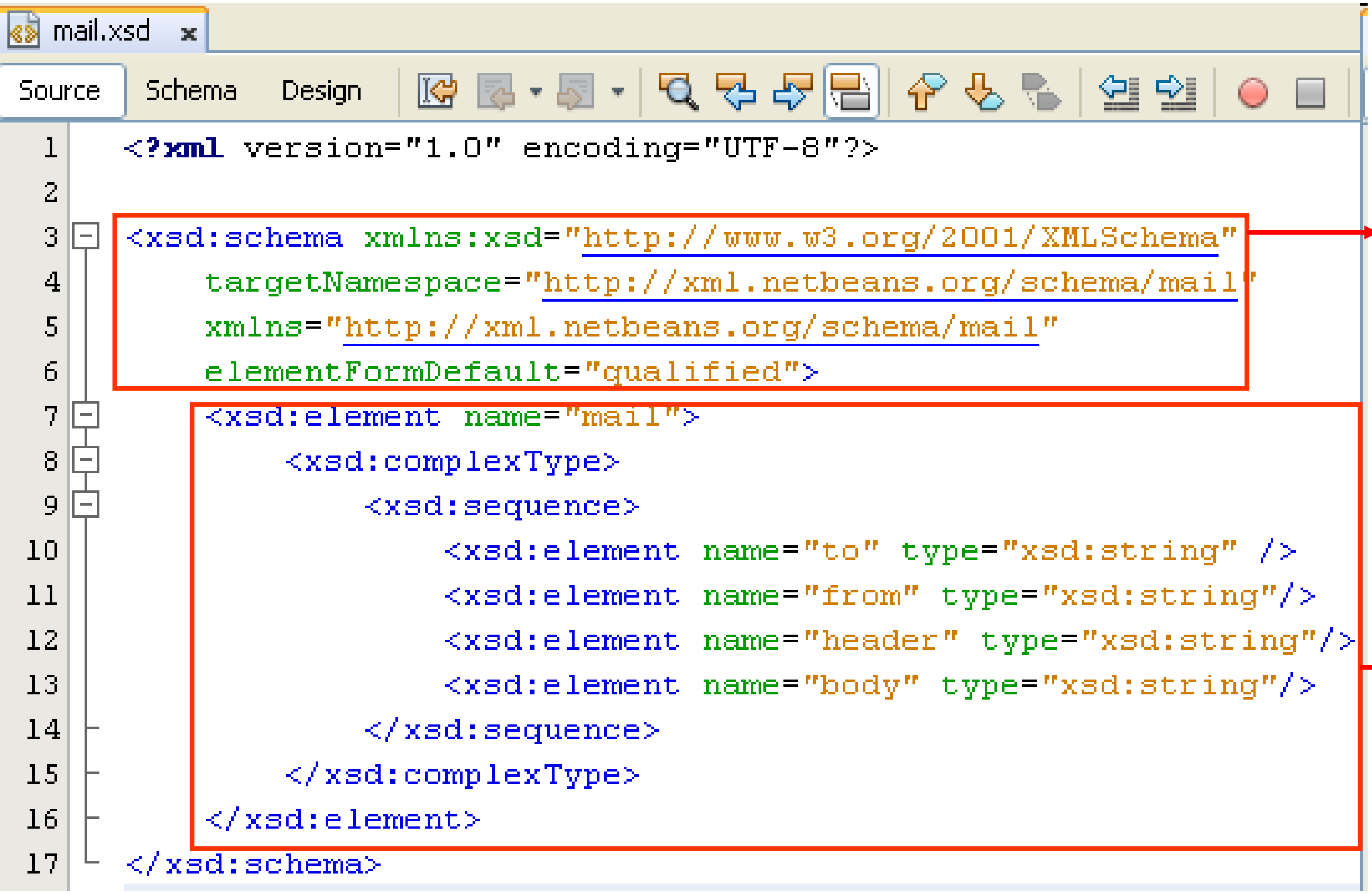
Schemas
What is an XML Schema?

- XML Schema **features**
  - Schema **supports data types** and **ability** to **create** the required **data type**
    - It is easy to **define**, **validate** valid document **content** and **data formats**, and **implement** the **restrictions** on data
    - Richer data types: Schema **defines many data types**
    - Archetypes: allow to **define own name data type from preexisting data types**
    - Attribute grouping: **make relationship** to all attributes
  - Schema is **portable & efficient**: Use **same XML syntax**
  - Schema **secures** data **communications**
    - The **sender** can **specify** the **data in** a **way** that the **receiver will understand**

Schemas
What is an XML Schema?

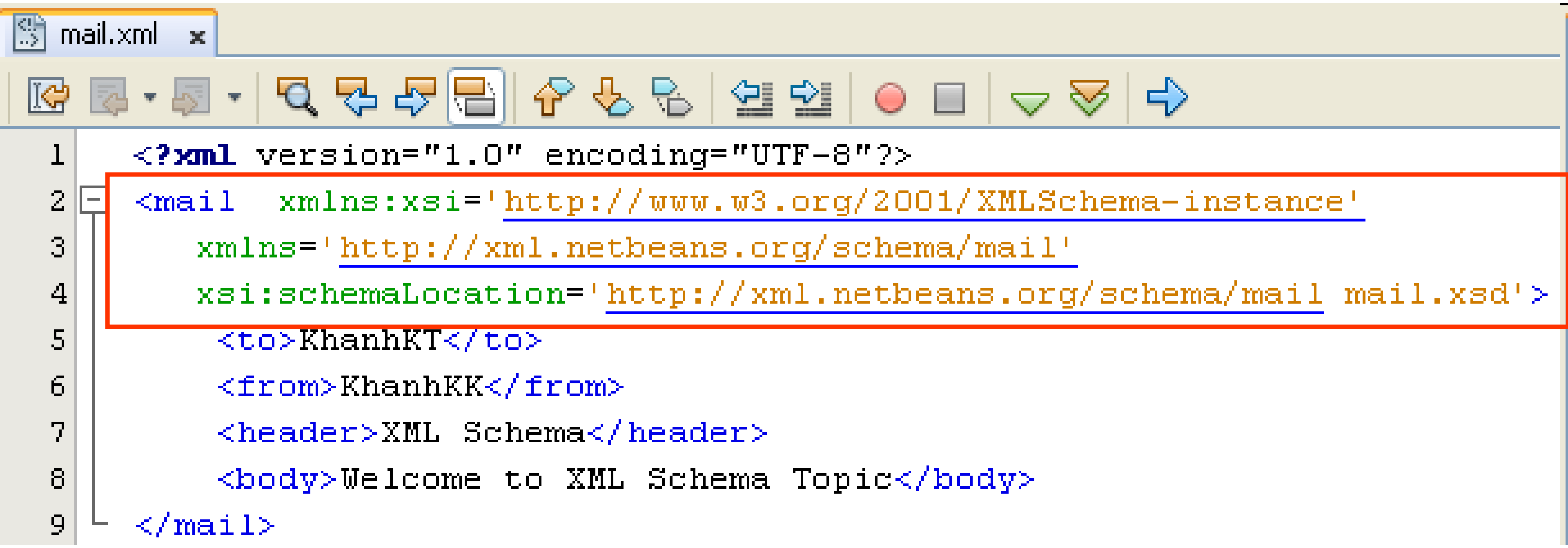
- XML Schema features (cont)
  - Schema is **extendable**
    - **Reusable** and **support reference** of **multiple schemas** in the same document
    - **Refinable archetypes**: allow an “open” content model that elements other than required elements can be presented
  - Schema **catch high-level mistakes**
    - **Checking** a **required** field information is **missing** or in **wrong format**, or an element name is **misspelled after** the XML is **checked** in w**ell-formed**
  - Schema **supports namespace**: **reusable**

Introduction
Exploring XSD



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     targetNamespace="http://xml.netbeans.org/schema/mail"
     xmlns="http://xml.netbeans.org/schema/mail"
     elementFormDefault="qualified">
     <xsd:element name="mail">
         <xsd:complexType>
             <xsd:sequence>
                 <xsd:element name="to" type="xsd:string" />
                 <xsd:element name="from" type="xsd:string"/>
                 <xsd:element name="header" type="xsd:string"/>
                 <xsd:element name="body" type="xsd:string"/>
             </xsd:sequence>
         </xsd:complexType>
     </xsd:element>
</xsd:schema>
```

Introduction
Exploring XSD


mail.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<mail  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/mail'
   xsi:schemaLocation='http://xml.netbeans.org/schema/mail mail.xsd'>
    <to>KhanhKT</to>
    <from>KhanhKK</from>
    <header>XML Schema</header>
    <body>Welcome to XML Schema Topic</body>
</mail>
```

Schemas

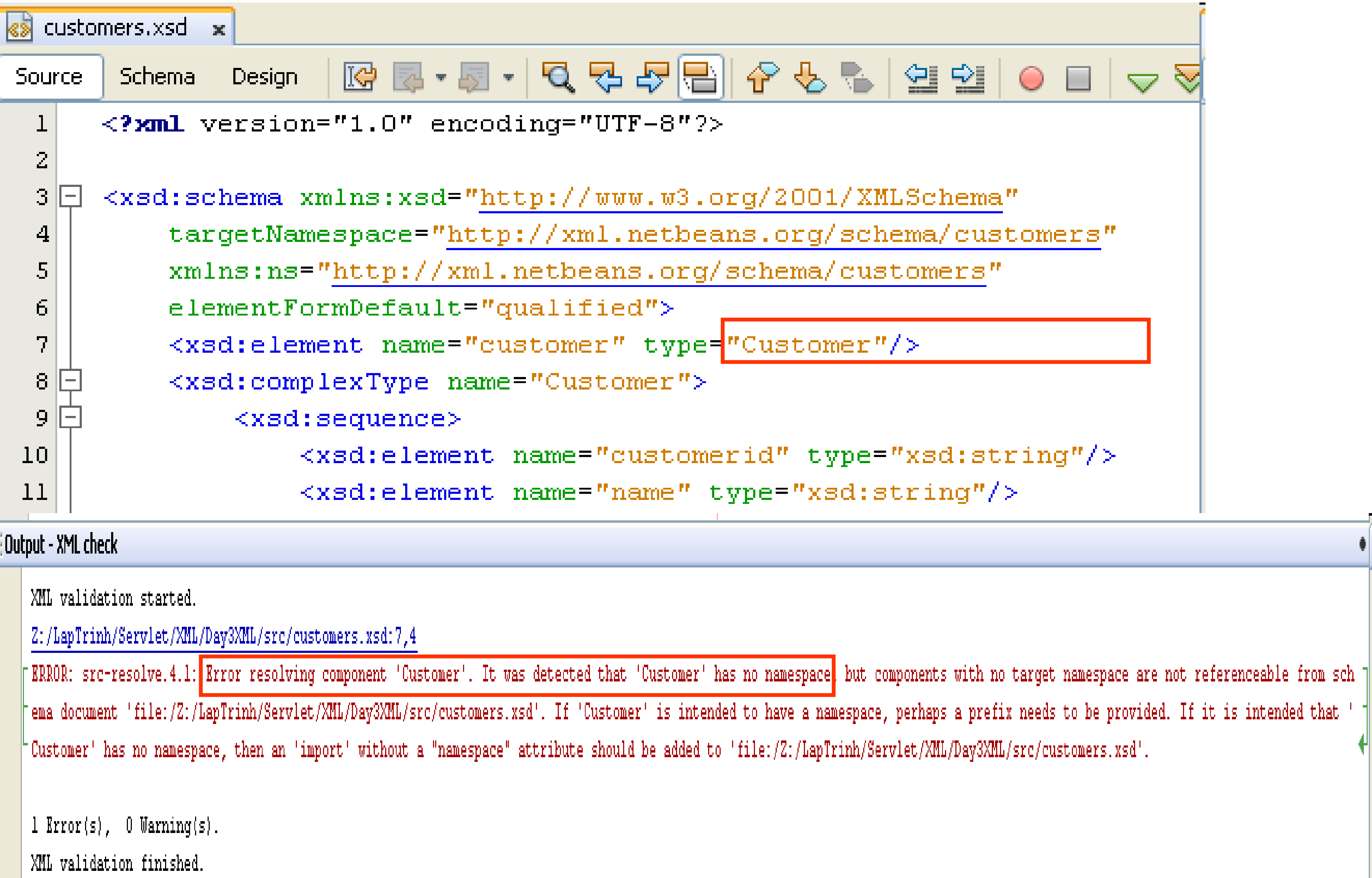
How to write an XML Schema

- An XML Schema is a **text-only document**, and **begins** with a **standard XML declaration**.
- Write **file** is **stored** with **.xsd extension**
  - A **schema declaration** is **typed** in the **beginning**
  - Define the **elements with** their **attributes**
  - Define the **restrictions** and **other constraints** (if any)
- **Reference** the **schema to xml** document
  - **Using schema declaration in xml** document to
    - **Inform** the schema-validator valid the namespace of particular element
    - **Determine instance namespace** applying to XML document
    - **Locate** the XSD file's location (schemaLocation)
  - **Define** XML document's **content**

Schemas
Structuring – XSD files

- The schema element is the root element of every XML Schema
  - **xmlns:xsd**="http://www.w3.org/2001/XMLSchema"
    - **Indicates** that the **elements** and **data types** used in the schema **come from** “http://www.w3.org/2001/XMLSchema” namespace
    - All elements and data types should be prefixed with xsd
  - **targetNamespace**=“namespace_uri"
    - **Indicates** that the **elements defined** by this **schema** (to, from, heading, body) **come from** the “namespace_uri” namespace
  - **xmlns**=“namespace_uri"
    - **Indicates** that the **default namespace** is “namespace_uri“ that is declared
  - elementFormDefault="qualified"
    - **Indicates** that any **elements used** by the XML instance document which were declared in this schema **must be namespace qualified**

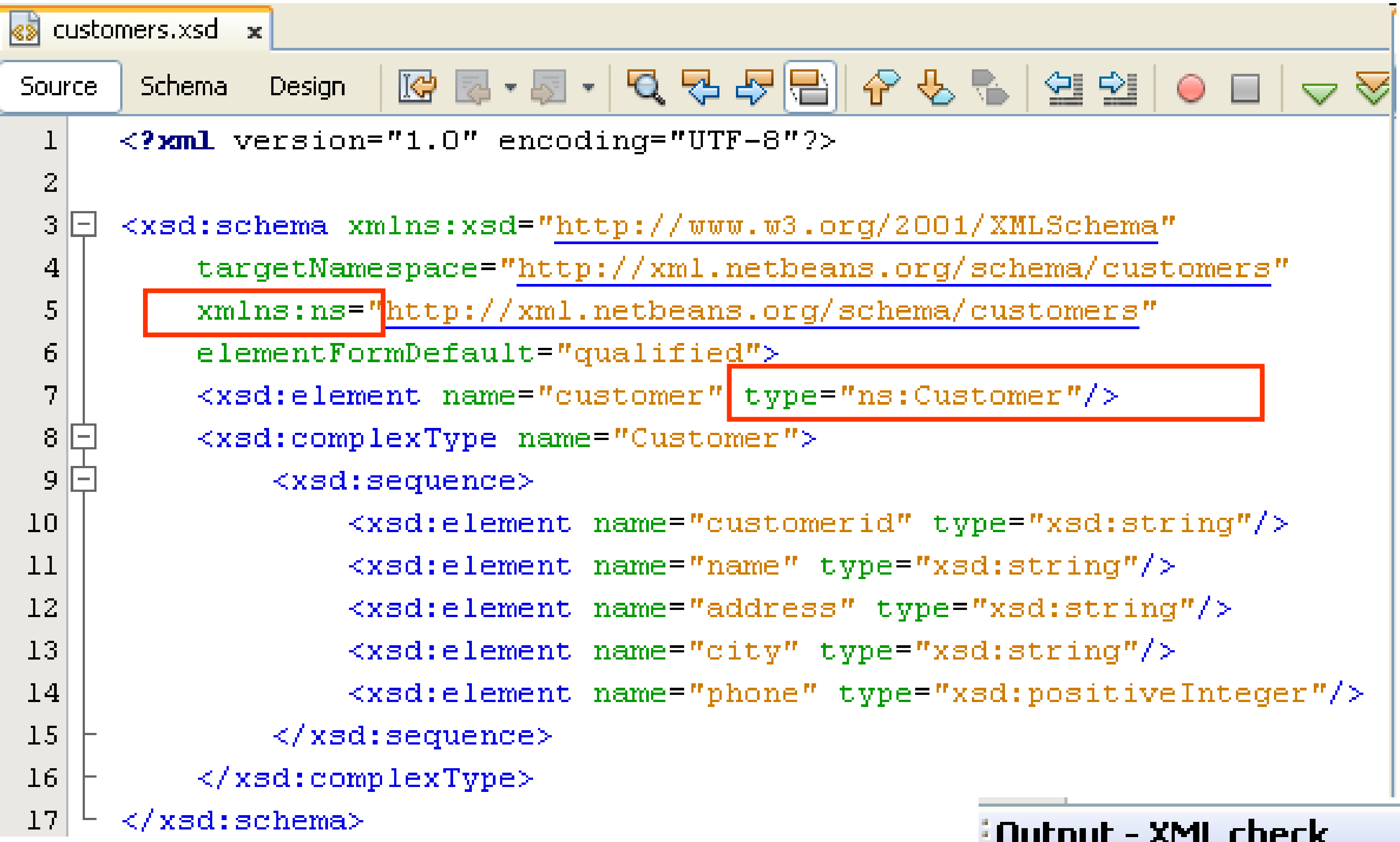
Schemas
Structuring – XSD files – Example



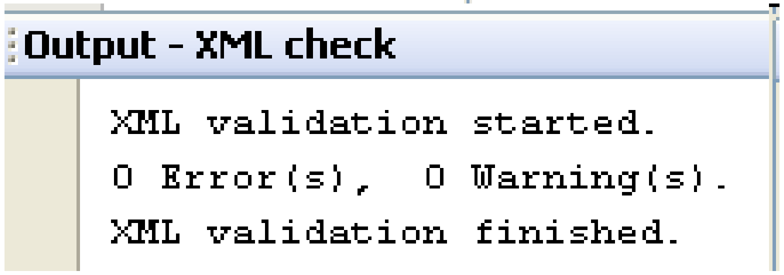
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/customers"
    xmlns:ns="http://xml.netbeans.org/schema/customers"
    elementFormDefault="qualified">
    <xsd:element name="customer" type="Customer"/>
    <xsd:complexType name="Customer">
        <xsd:sequence>
            <xsd:element name="customerid" type="xsd:string"/>
            <xsd:element name="name" type="xsd:string"/>
```

```
XML validation started.
Z:/LapTrinh/Servlet/XML/Day3XML/src/customers.xsd:7,4
ERROR: src-resolve.4.1: Error resolving component 'Customer'. It was detected that 'Customer' has no namespace but components with no target namespace are not referenceable from sch
ema document 'file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/customers.xsd'. If 'Customer' is intended to have a namespace, perhaps a prefix needs to be provided. If it is intended that '
Customer' has no namespace, then an 'import' without a "namespace" attribute should be added to 'file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/customers.xsd'.


1 Error(s),  0 Warning(s).
XML validation finished.
```

Schemas
Structuring – XSD files – Example



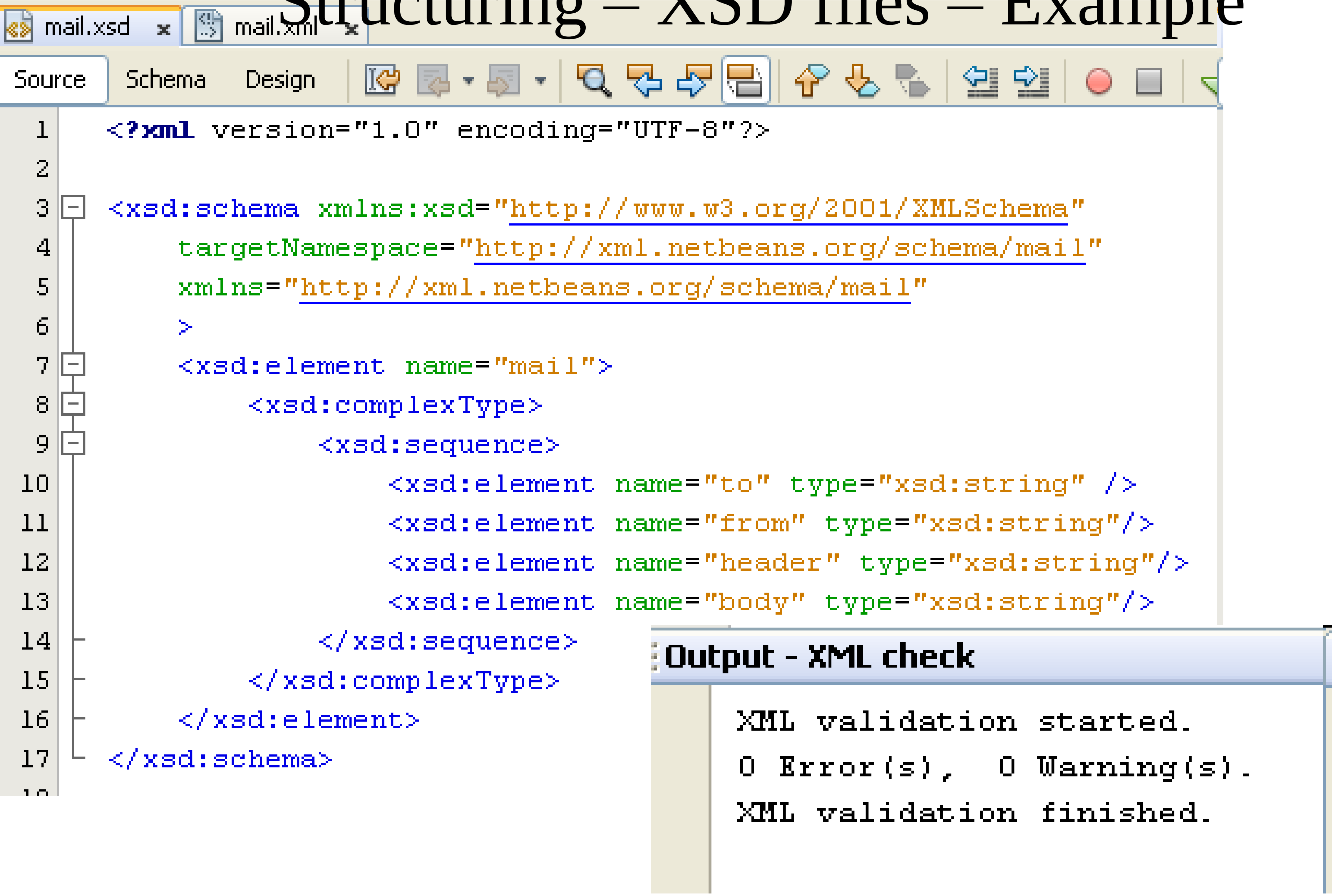
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/customers"
    xmlns:ns="http://xml.netbeans.org/schema/customers"
    elementFormDefault="qualified">
    <xsd:element name="customer" type="ns:Customer"/>
    <xsd:complexType name="Customer">
        <xsd:sequence>
            <xsd:element name="customerid" type="xsd:string"/>
            <xsd:element name="name" type="xsd:string"/>
            <xsd:element name="address" type="xsd:string"/>
            <xsd:element name="city" type="xsd:string"/>
            <xsd:element name="phone" type="xsd:positiveInteger"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```



Output - XML check

```
XML validation started.
0 Error(s),  0 Warning(s).
XML validation finished.
```

Schemas
Structuring – XSD files – Example

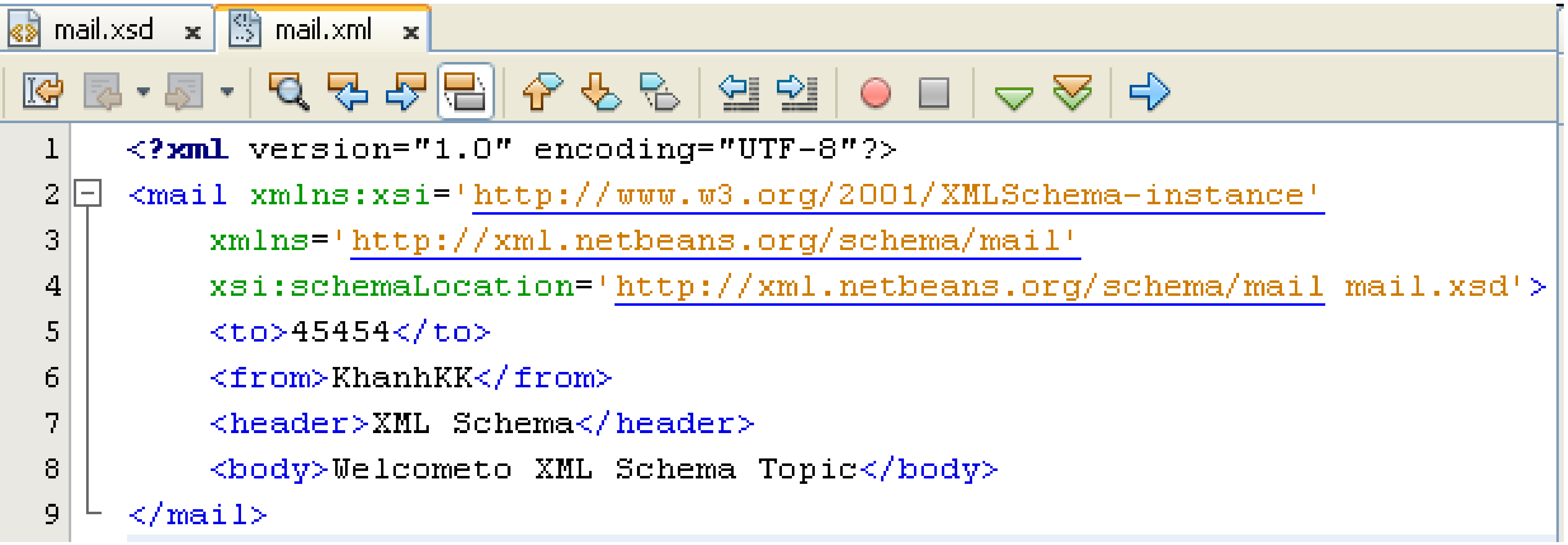


```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/mail"
    xmlns="http://xml.netbeans.org/schema/mail"
    >
    <xsd:element name="mail">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="to" type="xsd:string" />
                <xsd:element name="from" type="xsd:string"/>
                <xsd:element name="header" type="xsd:string"/>
                <xsd:element name="body" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Output - XML check

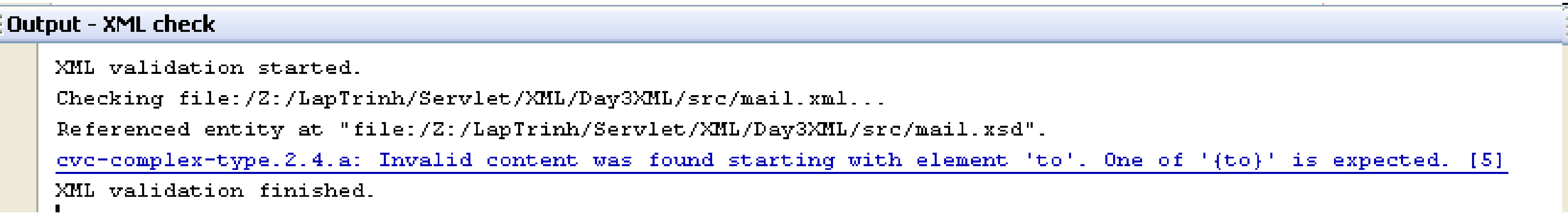
```
XML validation started.
0 Error(s),  0 Warning(s).
XML validation finished.
```

Schemas
Structuring – XSD files – Example



mail.xsd mail.xml

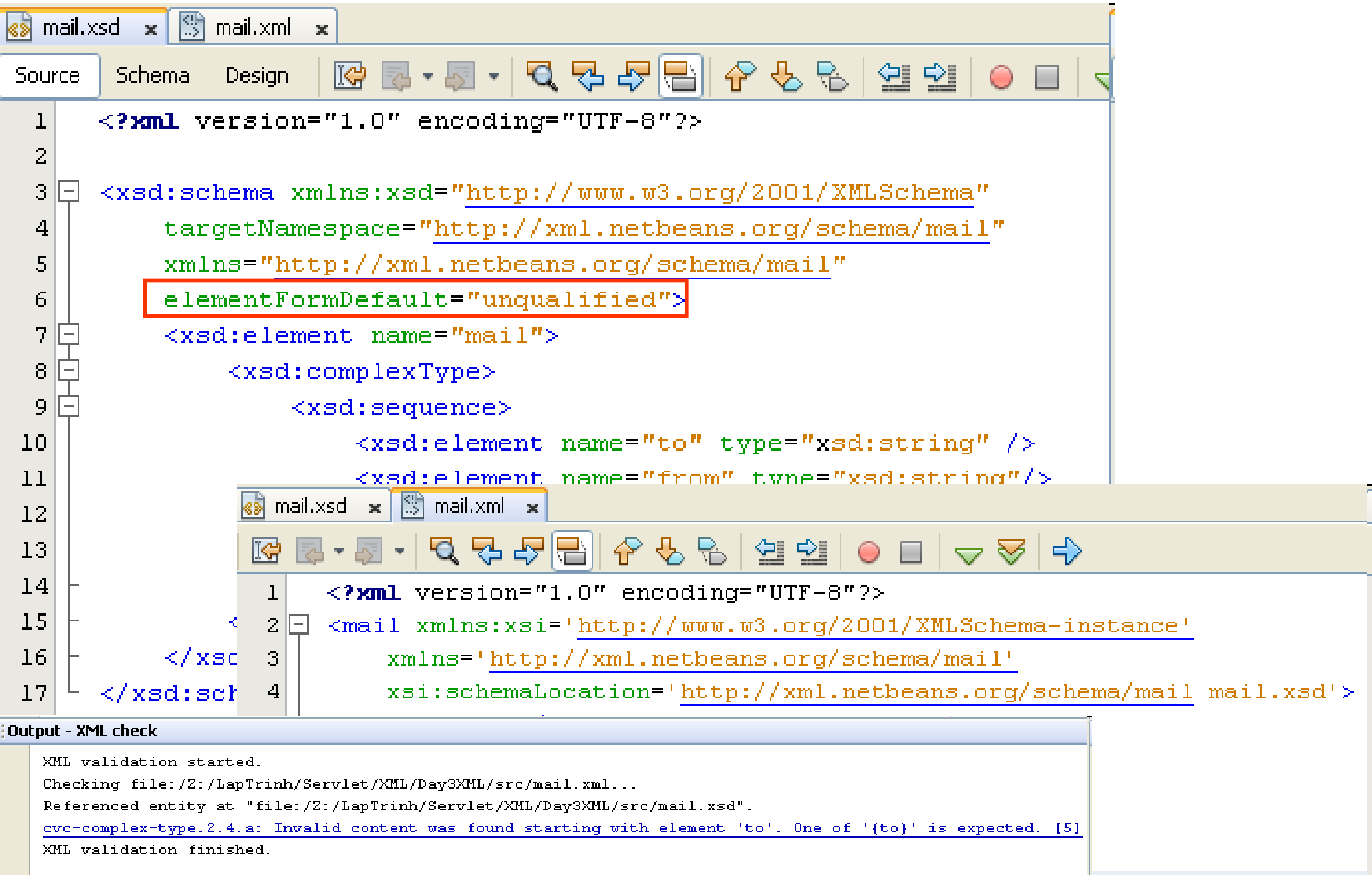
```
<?xml version="1.0" encoding="UTF-8"?>
<mail xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/mail'
    xsi:schemaLocation='http://xml.netbeans.org/schema/mail mail.xsd'>
    <to>45454</to>
    <from>KhanhKK</from>
    <header>XML Schema</header>
    <body>Welcometo XML Schema Topic</body>
</mail>
```



Output - XML check

```
XML validation started.
Checking file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mail.xml...
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mail.xsd".
cvc-complex-type.2.4.a: Invalid content was found starting with element 'to'. One of '{to}' is expected. [5]
XML validation finished.
```

Schemas
Structuring – XSD files – Example



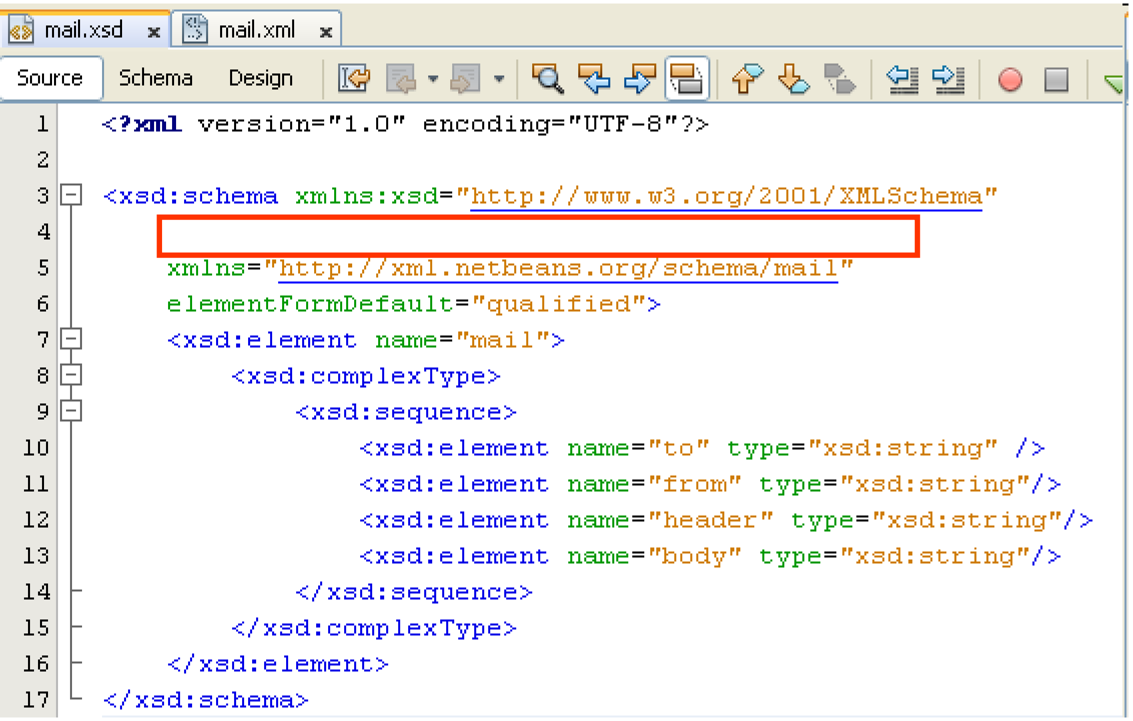
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/mail"
    xmlns="http://xml.netbeans.org/schema/mail"
    elementFormDefault="unqualified">
    <xsd:element name="mail">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="to" type="xsd:string" />
                <xsd:element name="from" type="xsd:string"/>



               <
    </xsd
</xsd:sch
```

```
<?xml version="1.0" encoding="UTF-8"?>
<mail xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/mail'
    xsi:schemaLocation='http://xml.netbeans.org/schema/mail mail.xsd'>
```

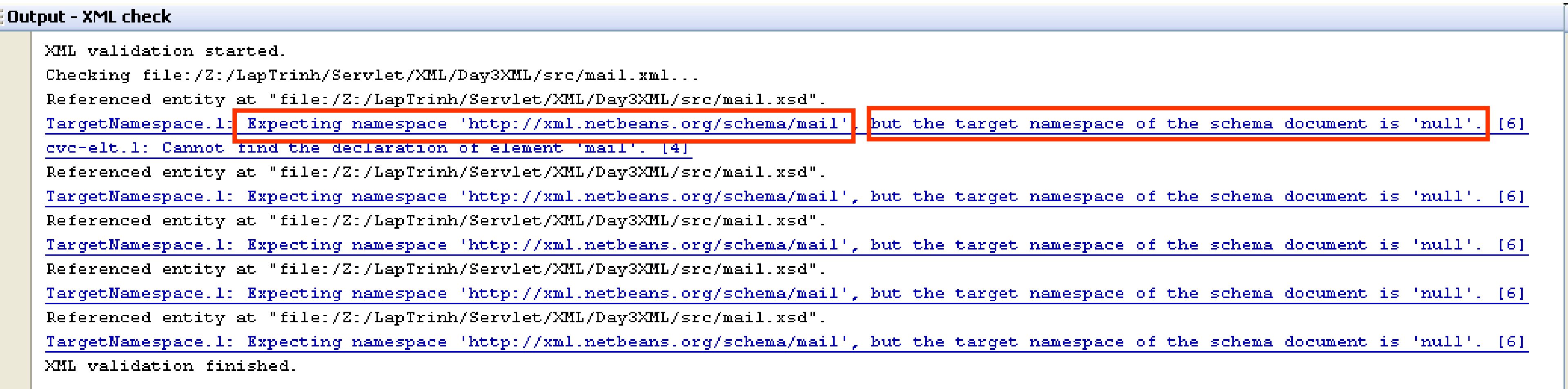
Output - XML check

```
XML validation started.
Checking file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mail.xml...
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mail.xsd".
cvc-complex-type.2.4.a: Invalid content was found starting with element 'to'. One of '{to}' is expected. [5]
XML validation finished.
```

Schemas
Structuring – XSD files – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"

    xmlns="http://xml.netbeans.org/schema/mail"
    elementFormDefault="qualified">
    <xsd:element name="mail">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="to" type="xsd:string" />
                <xsd:element name="from" type="xsd:string"/>
                <xsd:element name="header" type="xsd:string"/>
                <xsd:element name="body" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

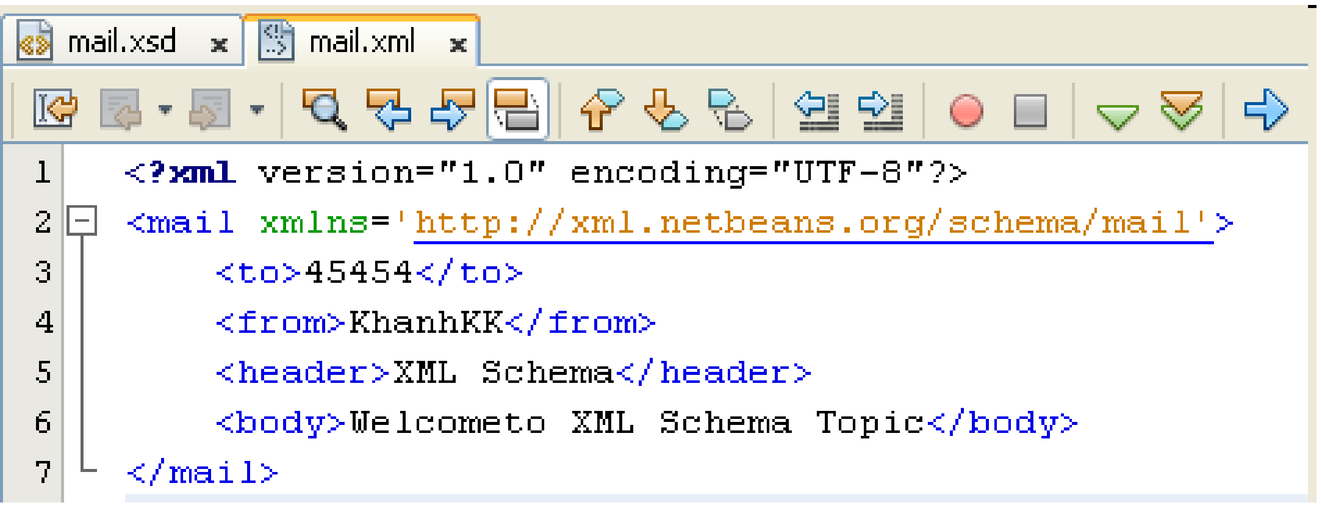


```
XML validation started.
Checking file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mail.xml...
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mail.xsd".
TargetNamespace.1: Expecting namespace 'http://xml.netbeans.org/schema/mail', but the target namespace of the schema document is 'null'. [6]
cvc-elt.1: Cannot find the declaration of element 'mail'. [4]
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mail.xsd".
TargetNamespace.1: Expecting namespace 'http://xml.netbeans.org/schema/mail', but the target namespace of the schema document is 'null'. [6]
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mail.xsd".
TargetNamespace.1: Expecting namespace 'http://xml.netbeans.org/schema/mail', but the target namespace of the schema document is 'null'. [6]
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mail.xsd".
TargetNamespace.1: Expecting namespace 'http://xml.netbeans.org/schema/mail', but the target namespace of the schema document is 'null'. [6]
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mail.xsd".
TargetNamespace.1: Expecting namespace 'http://xml.netbeans.org/schema/mail', but the target namespace of the schema document is 'null'. [6]
XML validation finished.
```

Schemas
Structuring – Apply to XML files

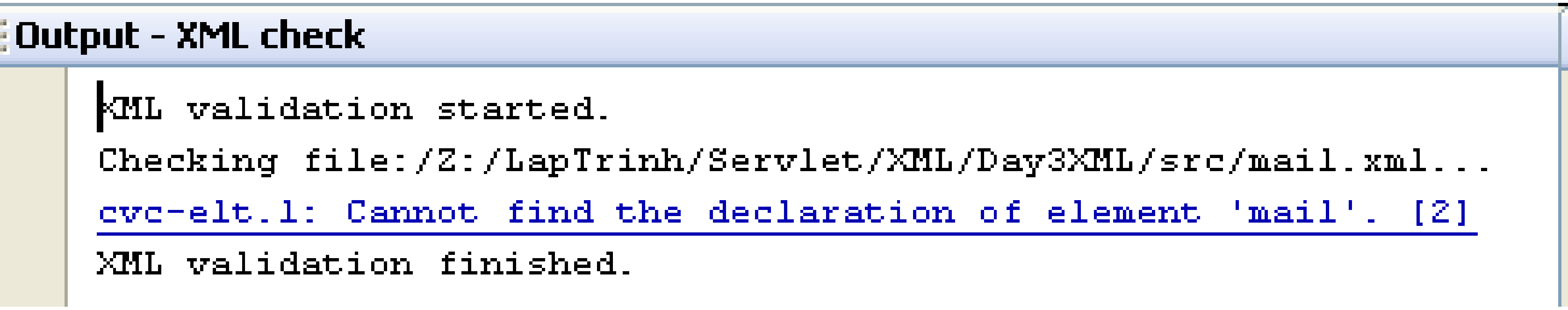
- The root element is defined with some attributes
  - **xmlns:xsi**= “http://www.w3.org/2001/XMLSchema-instance”
    - XML Schema Definition Language: **Structures defines** several **attributes for direct use in** any **XML** documents. They are defined in Schema Instance Namespace
  - **xmlns**=“http://xml.netbeans.org/schema/mail”
    - **Specifies** the **default namespace declaration**.
    - This declaration **tells** the **schema-validator** that all the **elements used** in this XML document **are declared** in the “http://xml.netbeans.org/schema/mail” namespace
  - **xsi:schemaLocation**=“namespace_uri **instance**”>
    - The **first** value is the **namespace** to use.
    - The **second** value is the **location** of the XML **schema** to use for that namespace

Schemas
Structuring – Apply to XML files – Example



mail.xsd | mail.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<mail xmlns='http://xml.netbeans.org/schema/mail'>
    <to>45454</to>
    <from>KhanhKK</from>
    <header>XML Schema</header>
    <body>Welcometo XML Schema Topic</body>
</mail>
```

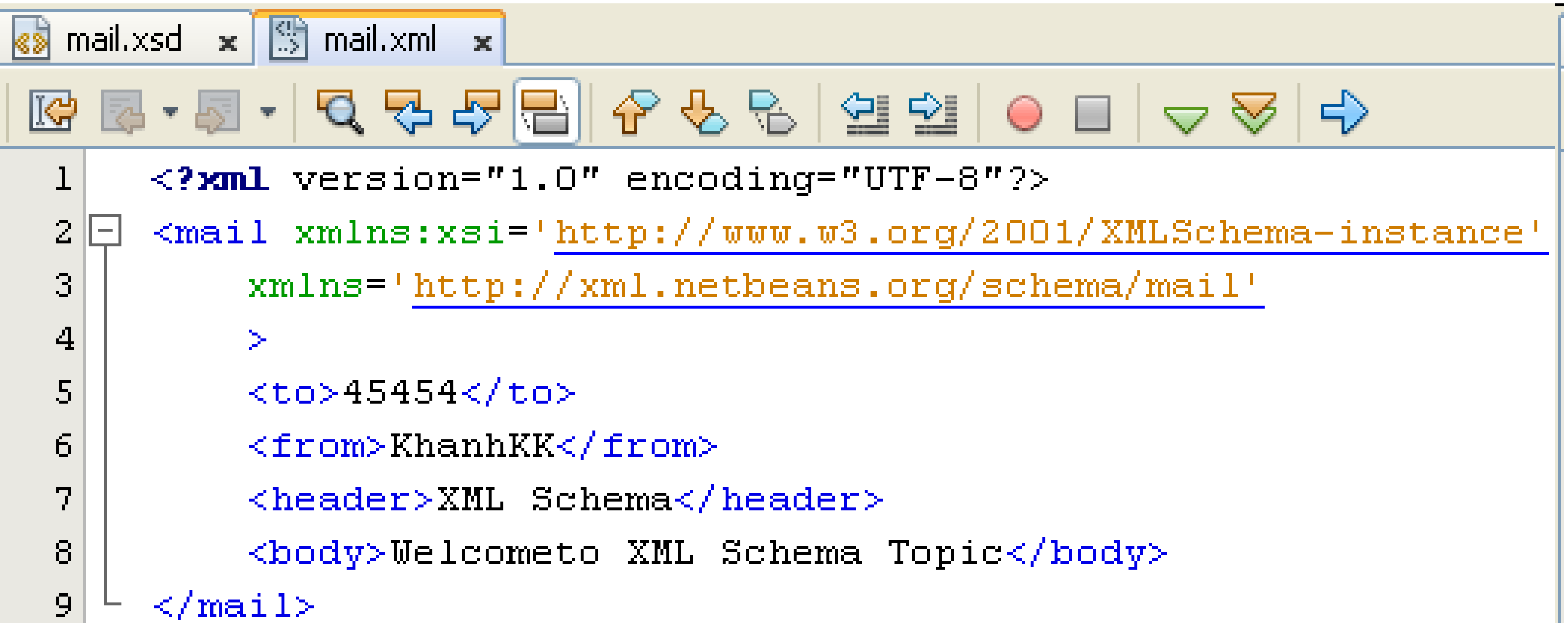


Output - XML check

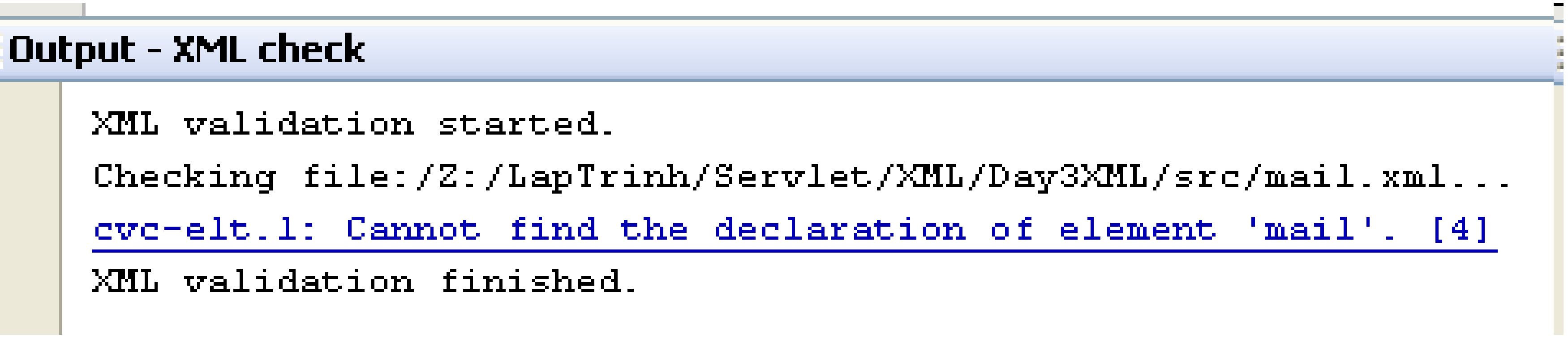
```
XML validation started.
Checking file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mail.xml...
cvc-elt.1: Cannot find the declaration of element 'mail'. [2]
XML validation finished.
```

Schemas

Structuring – Apply to XML files – Example



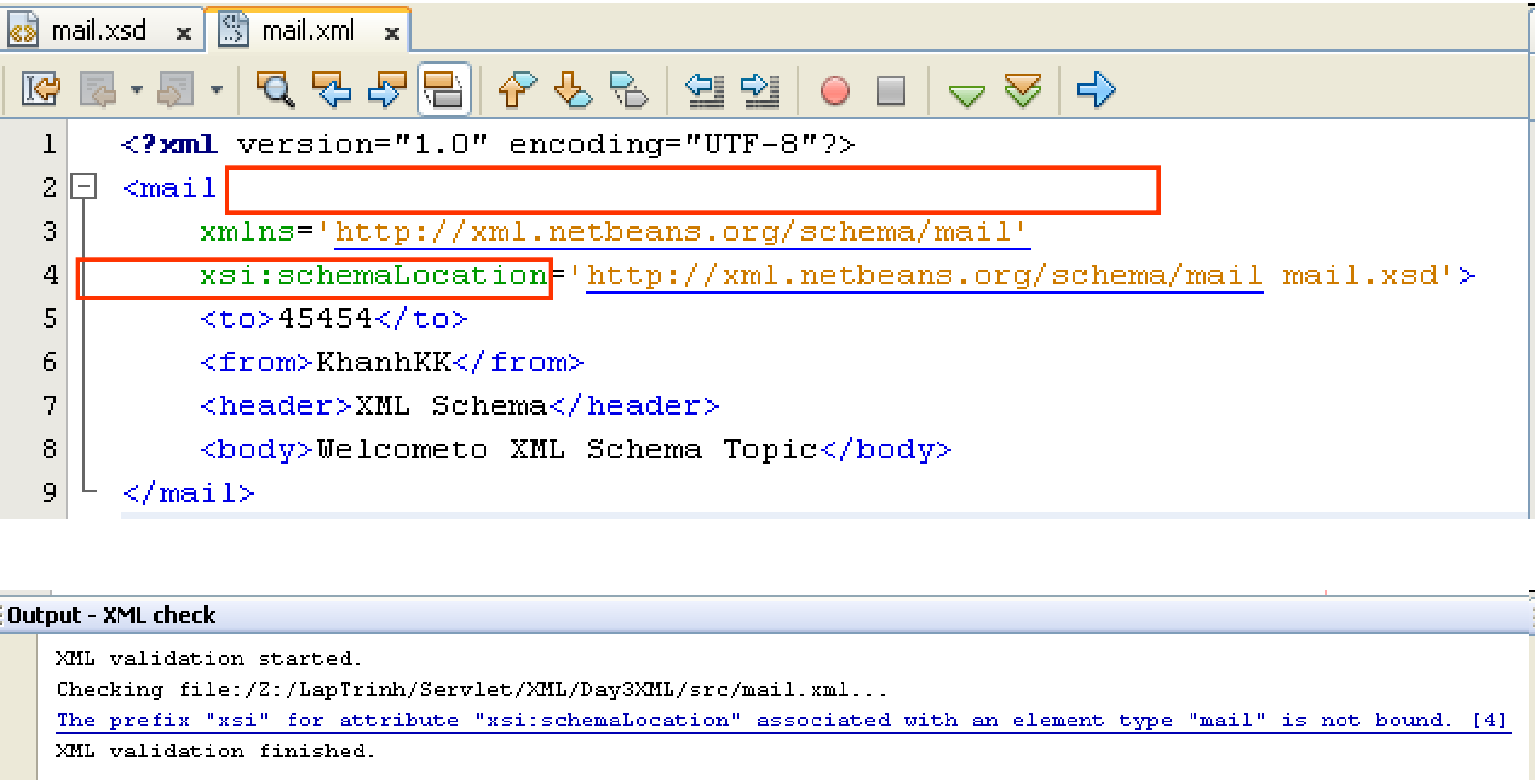
```xml
<?xml version="1.0" encoding="UTF-8"?>
<mail xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
      xmlns='http://xml.netbeans.org/schema/mail'
      >
      <to>45454</to>
      <from>KhanhKK</from>
      <header>XML Schema</header>
      <body>Welcometo XML Schema Topic</body>
</mail>
```



```
XML validation started.
Checking file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mail.xml...
cvc-elt.1: Cannot find the declaration of element 'mail'. [4]
XML validation finished.
```

Schemas

Structuring – Apply to XML files – Example

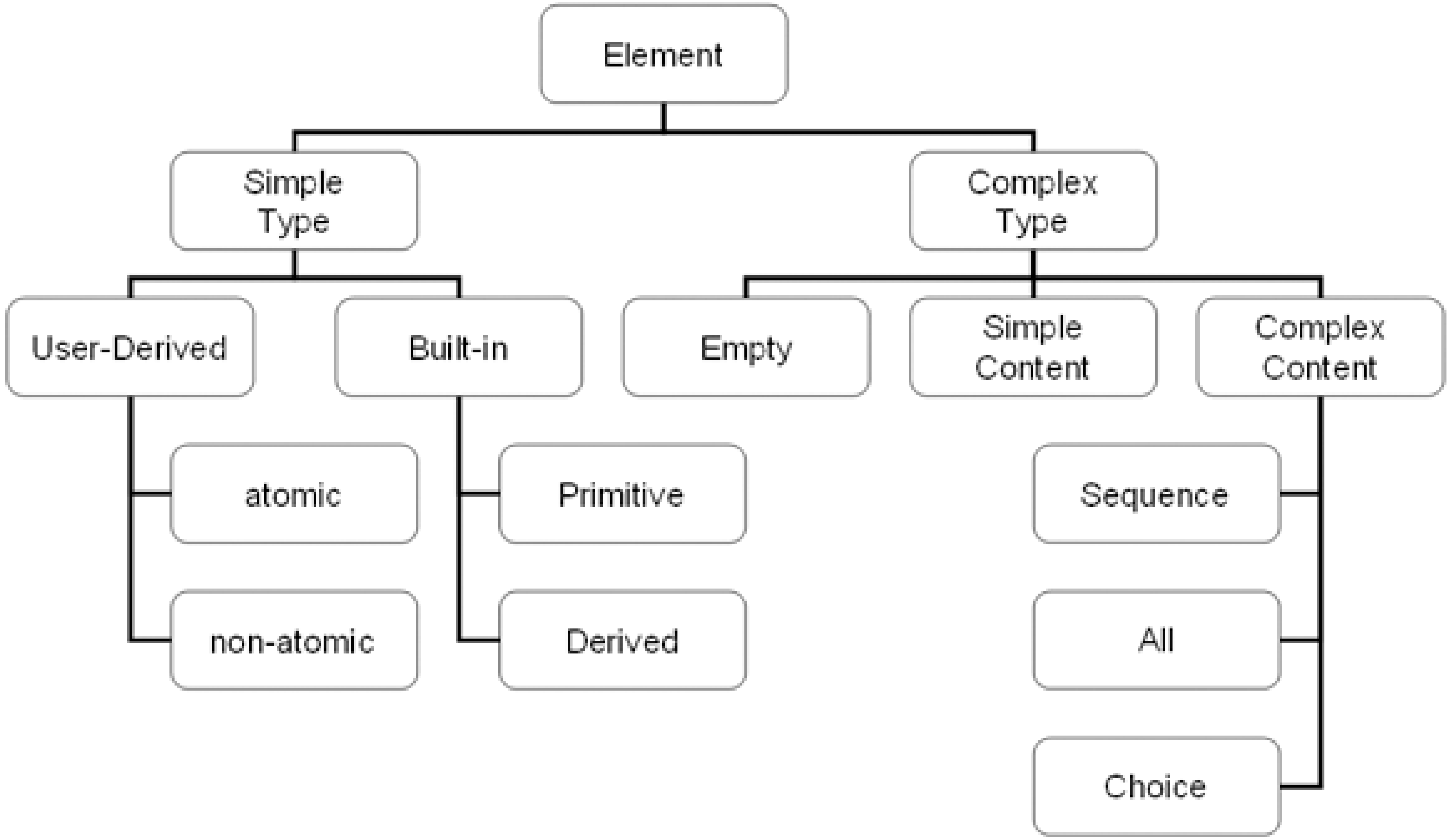


```
<?xml version="1.0" encoding="UTF-8"?>
<mail
    xmlns='http://xml.netbeans.org/schema/mail'
    xsi:schemaLocation='http://xml.netbeans.org/schema/mail mail.xsd'>
    <to>45454</to>
    <from>KhanhKK</from>
    <header>XML Schema</header>
    <body>Welcometo XML Schema Topic</body>
</mail>
```

Output - XML check

```
XML validation started.
Checking file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mail.xml...
The prefix "xsi" for attribute "xsi:schemaLocation" associated with an element type "mail" is not bound. [4]
XML validation finished.
```

Schemas
Structuring – XSD Elements



Schemas
Structuring – XSD Simple Type Elements

- Is an XML element that **contains only text**.
- **Cannot** contain any other elements or attributes
- **Syntax**: **declaration** vs. **reference** to global element
  - <xsd:element name=“element_name" type=“data_type"/>
  - <xsd:element ref=“referenced_element_name”/>
- Default, Fixed, and Nil Values
  - A **default** value is **automatically assigned** to the element **when no** other **value** is **specified**

<xsd:element name=“element_name" type=“data_type“ default=“default_value”/>

  - A **fixed** value is also **automatically assigned** to the element, and it **cannot specify another value**

<xsd:element name=“element_name" type=“data_type“ fixed=“fixed_value”/>

  - A **default and fixed value are not existed** in the **same time**
  - When an optional element is left out of an XML instance, it has no clear meaning
    - elements in XML can be set to nil by setting the xsi:nil attribute to true

<xsd:element name=“element_name" type=“data_type“ nillable=“true|false”/>

Schemas
Structuring – XSD Data Types

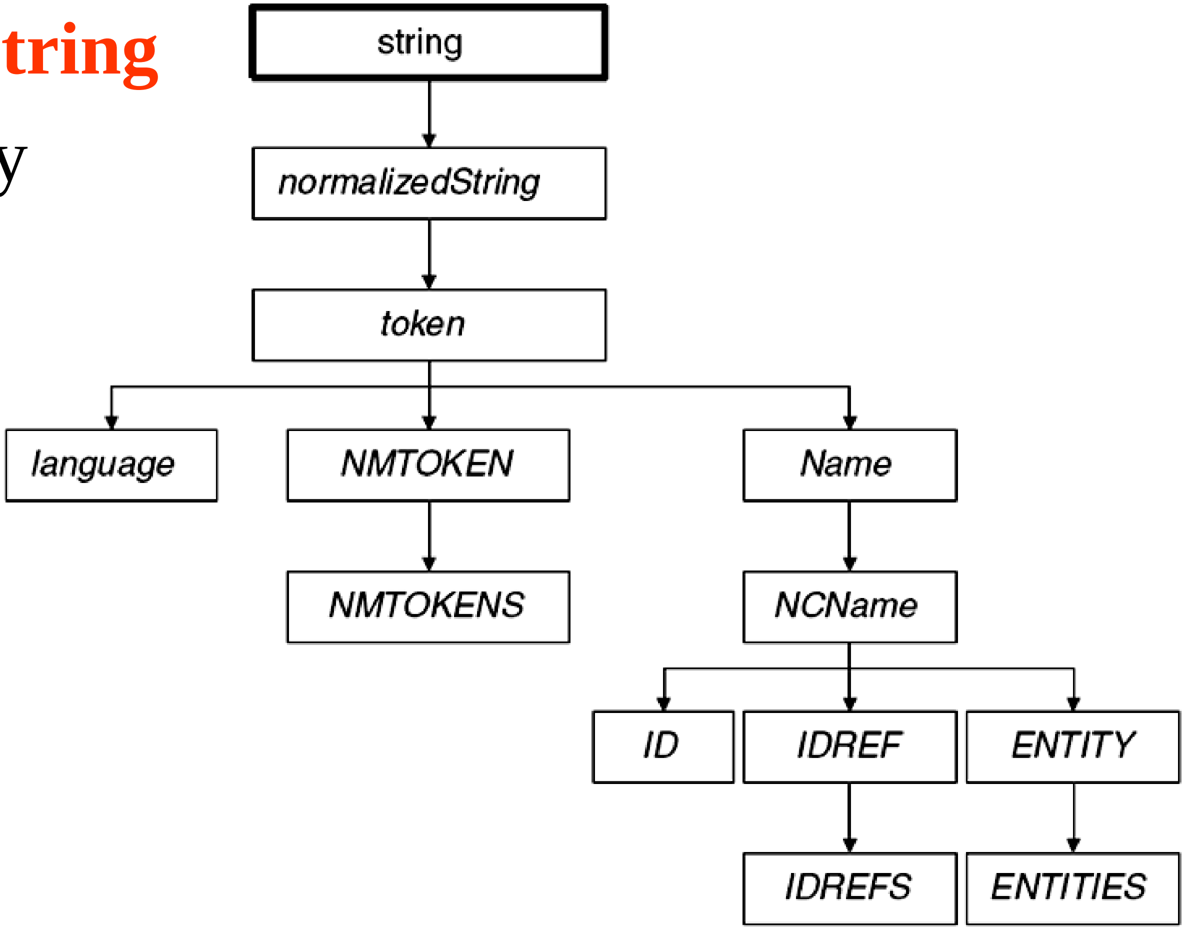
- XML Schema defines 02 sort of data types
  - **Built-in** data types
    - Are **available** to all XML Schema authors
    - Should be **implemented** by a **conforming** processor
    - Can be used to **specify** and **validate** the **intended data type** of the content
    - **Allow** a user to **create a user-defined data by extending** the built-in data types **using facets (→ derived types)**
    - Are specified into groups as
      - string, boolean, numeric, data and time, binary, anyURI
  - **User-derived** data types
    - Are **defined** in individual **schema instances**
    - Are particular to that schema

Schemas
Structuring – XSD Data Types

- Are defined how the values are represented as character in XML following
  - **Basic** type
    - A **value space**
      - The set of values the type can hold
    - A **lexical spaces**
      - The set of lexical is used to declare the represented name
  - **User-derived** type
    - A **basic** type
    - **Canonical representation:** using correctly syntax
    - List of **facets**

Schemas
Structuring – XSD Data Types – string

- A group of characters
  - **Contains** characters, line feeds, carriage returns, and tab characters
  - **Consists** of a **combination** of Unicode characters
- **Syntax**: **xsd:string**
- Type hierarchy



Schemas
Structuring – XSD Data Types – string

Type	Description
normalizedString	- Contains characters, but the XML processor will **remove** line feeds, carriage returns, and tab characters. - **Syntax: xsd:normalizedString**
token	- Contains characters, but the XML processor will **remove** line feeds, carriage returns, tabs, leading and trailing spaces, and multiple spaces - **Syntax: xsd:token**
language	- A string that contains a valid language id. **Syntax: xsd:language**
NMTOKEN	A string that represents the NMTOKEN attribute in XML (only used with schema attributes). **Syntax: xsd:NMTOKEN**
NMTOKENS	List of NMTOKEN values separated by whitespace (only used with schema attributes). **Syntax: xsd:NMTOKENS**
Name	A string that contains a valid XML name. **Syntax: xsd:Name**
QName	- Represent qualified names according to Namespace (order:Order) - **Syntax: xsd:Qname**

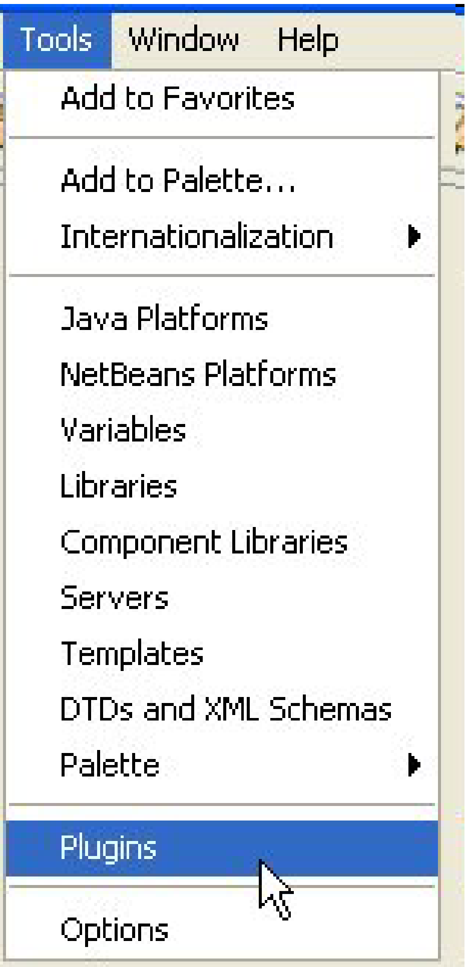
Schemas
Structuring – XSD Data Types – string

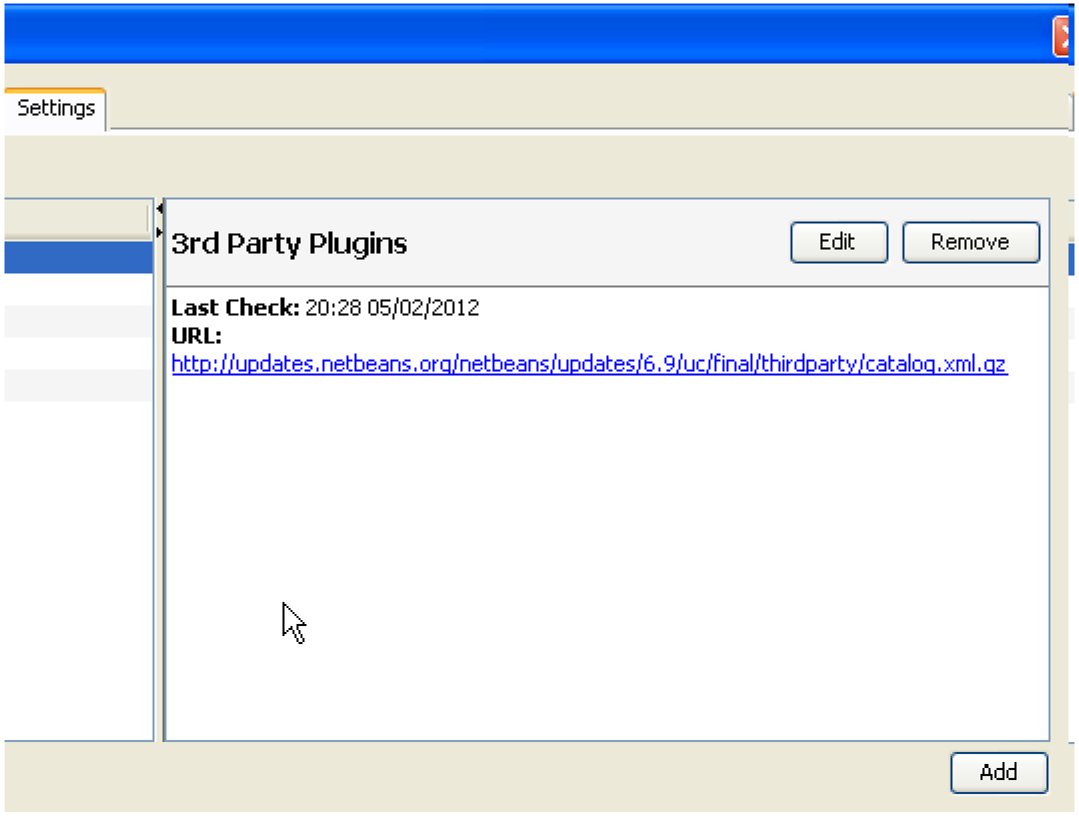
Type	Description
NCNAME	-An XML non-colonized name, typically used for the local names of namespace-qualified elements and attributes; the part after the prefix and the colon (**e.g. a name that is not qualified with a namespace-related prefix**) -**Syntax: xsd:NCName**
ID	A string that represents the ID attribute in XML (only used with schema attributes). **Syntax: xsd:ID**
IDREF	A string must match an ID value elsewhere in XML document (only used with schema attributes). **Syntax: xsd:IDREF**
IDREFS	List of ID values separated by white space (only used with schema attributes). **Syntax: xsd:IDREFS**
ENTITY	A String must match the name of an unparsed entity declared (only used with schema attributes). Syntax: **xsd:ENTITY**
ENTITIES	List of ENTITY names separated by white space (only used with schema attributes). Syntax: **xsd:ENTITIES**

Schemas

Add Plugin support Schema in Netbeans

- Open menu Tools, choose Plugin
  - Then choose tag Settings, then click Add button





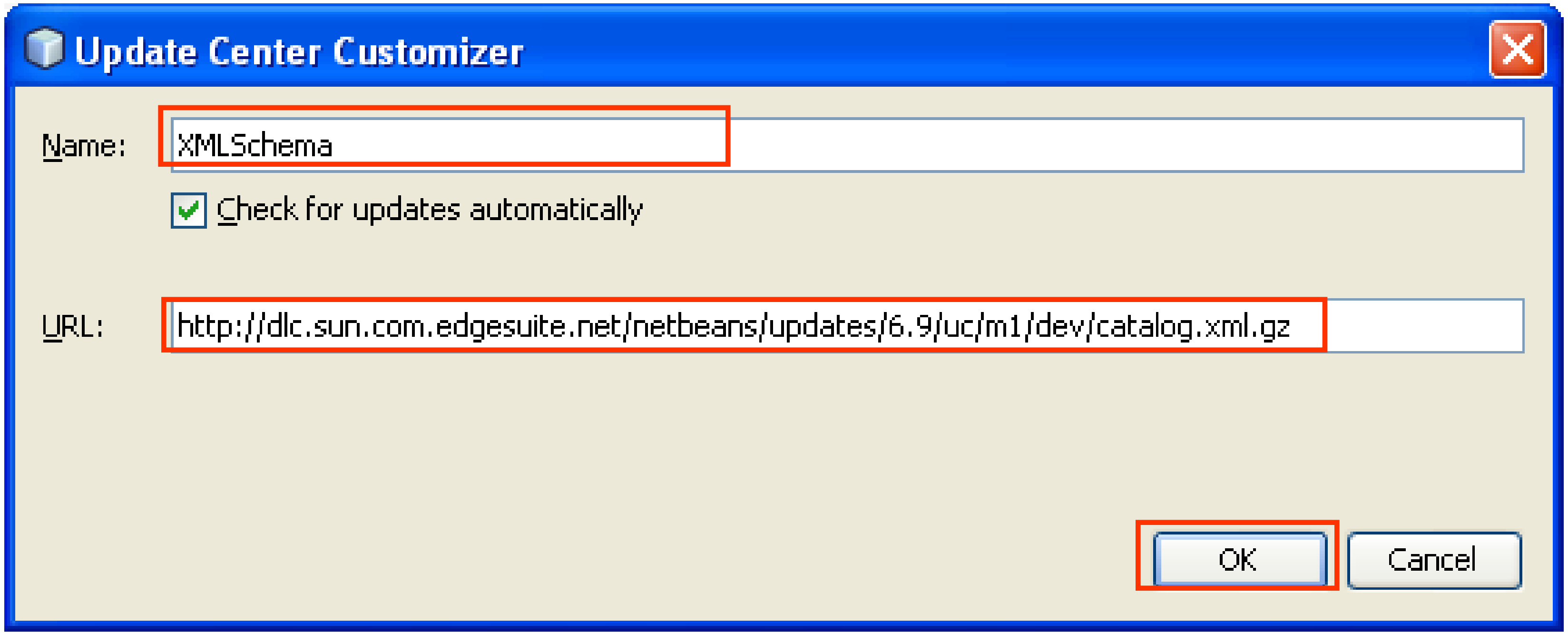
3rd Party Plugins

Last Check: 20:28 05/02/2012
URL:
http://updates.netbeans.org/netbeans/updates/6.9/uc/final/thirdparty/catalog.xml.gz

Fpt University

Schemas

Add Plugin support Schema in Netbeans



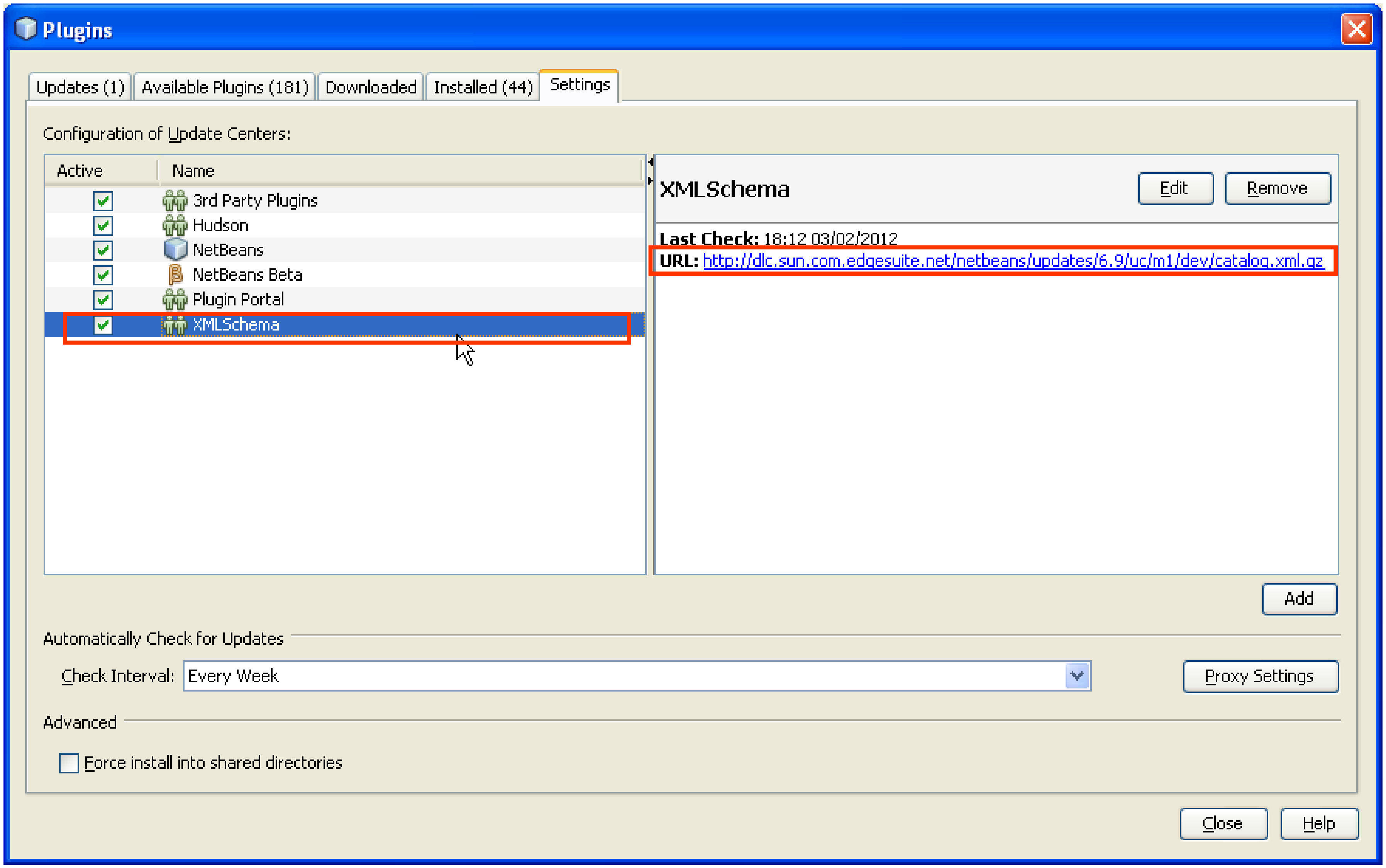
Name: XMLSchema

[✓] Check for updates automatically

URL: http://dlc.sun.com.edgesuite.net/netbeans/updates/6.9/uc/m1/dev/catalog.xml.gz

Schemas

Add Plugin support Schema in Netbeans



Configuration of Update Centers:

Active	Name
✓	3rd Party Plugins
✓	Hudson
✓	NetBeans
✓	NetBeans Beta
✓	Plugin Portal
✓	XMLSchema

XMLSchema

Last Check: 18:12 03/02/2012

URL: http://dlc.sun.com.edgesuite.net/netbeans/updates/6.9/uc/m1/dev/catalog.xml.gz

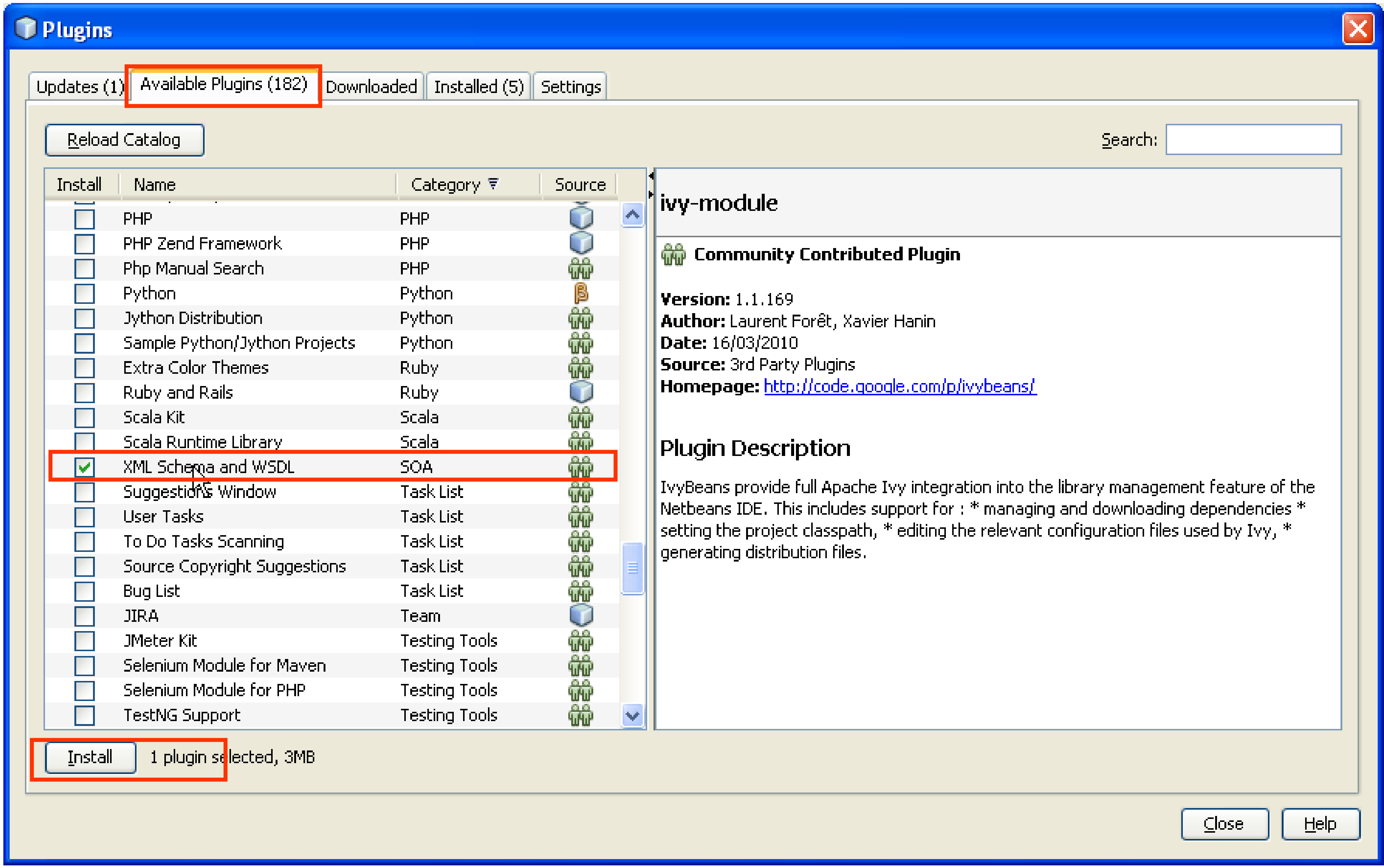
Automatically Check for Updates

Check Interval: Every Week

Advanced

☐ Force install into shared directories

Schemas
Add Plugin support Schema in Netbeans



Plugins

Updates (1) | Available Plugins (182) | Downloaded | Installed (5) | Settings

Reload Catalog

Search:

Install	Name	Category	Source
☐	PHP	PHP	
☐	PHP Zend Framework	PHP	
☐	Php Manual Search	PHP	
☐	Python	Python	
☐	Jython Distribution	Python	
☐	Sample Python/Jython Projects	Python	
☐	Extra Color Themes	Ruby	
☐	Ruby and Rails	Ruby	
☐	Scala Kit	Scala	
☐	Scala Runtime Library	Scala	
✓	XML Schema and WSDL	SOA	
☐	Suggestions Window	Task List	
☐	User Tasks	Task List	
☐	To Do Tasks Scanning	Task List	
☐	Source Copyright Suggestions	Task List	
☐	Bug List	Task List	
☐	JIRA	Team	
☐	JMeter Kit	Testing Tools	
☐	Selenium Module for Maven	Testing Tools	
☐	Selenium Module for PHP	Testing Tools	
☐	TestNG Support	Testing Tools	

ivy-module

Community Contributed Plugin

Version: 1.1.169
Author: Laurent Forêt, Xavier Hanin
Date: 16/03/2010
Source: 3rd Party Plugins
Homepage: http://code.google.com/p/ivybeans/

Plugin Description

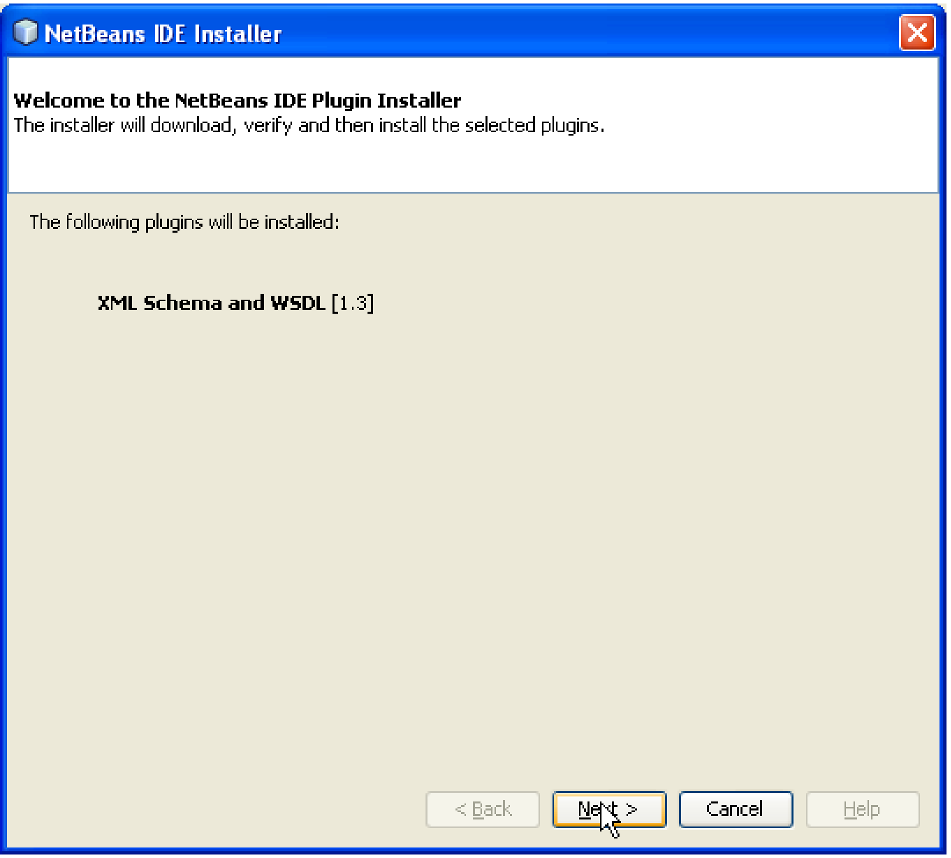
IvyBeans provide full Apache Ivy integration into the library management feature of the Netbeans IDE. This includes support for : * managing and downloading dependencies * setting the project classpath, * editing the relevant configuration files used by Ivy, * generating distribution files.

Install   1 plugin selected, 3MB

Close   Help

Schemas

Add Plugin support Schema in Netbeans

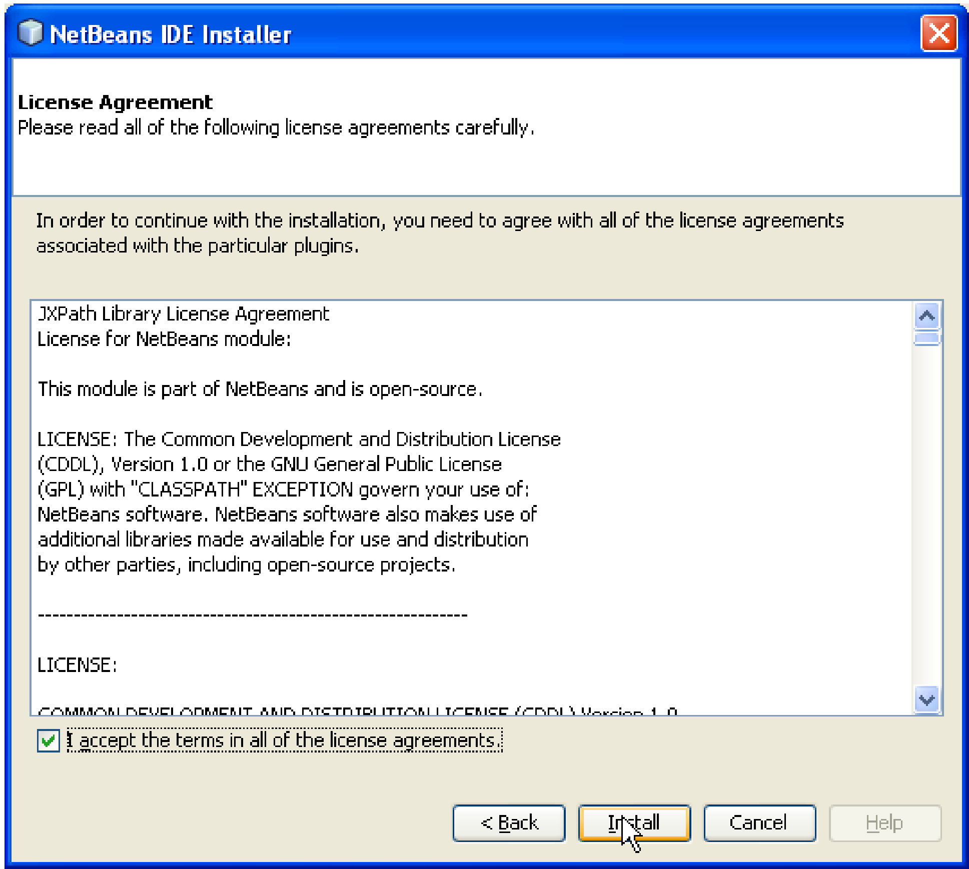


NetBeans IDE Installer

Welcome to the NetBeans IDE Plugin Installer
The installer will download, verify and then install the selected plugins.

The following plugins will be installed:

XML Schema and WSDL [1.3]



NetBeans IDE Installer

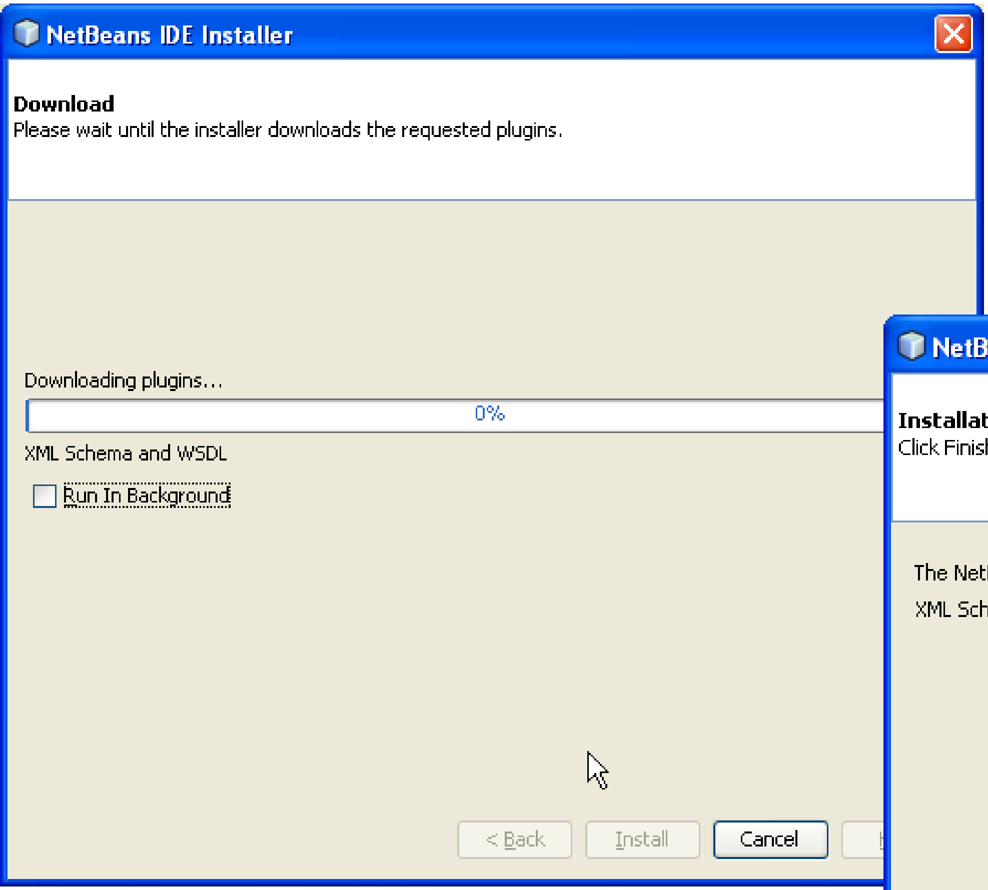
License Agreement
Please read all of the following license agreements carefully.

In order to continue with the installation, you need to agree with all of the license agreements associated with the particular plugins.

```
JXPath Library License Agreement
License for NetBeans module:

This module is part of NetBeans and is open-source.

LICENSE: The Common Development and Distribution License
(CDDL), Version 1.0 or the GNU General Public License
(GPL) with "CLASSPATH" EXCEPTION govern your use of:
NetBeans software. NetBeans software also makes use of
additional libraries made available for use and distribution
by other parties, including open-source projects.

------------------------------------------------------------

LICENSE:

COMMON DEVELOPMENT AND DISTRIBUTION LICENSE (CDDL) Version 1.0
```

[✓] I accept the terms in all of the license agreements.

Schemas
Add Plugin support Schema in Netbeans



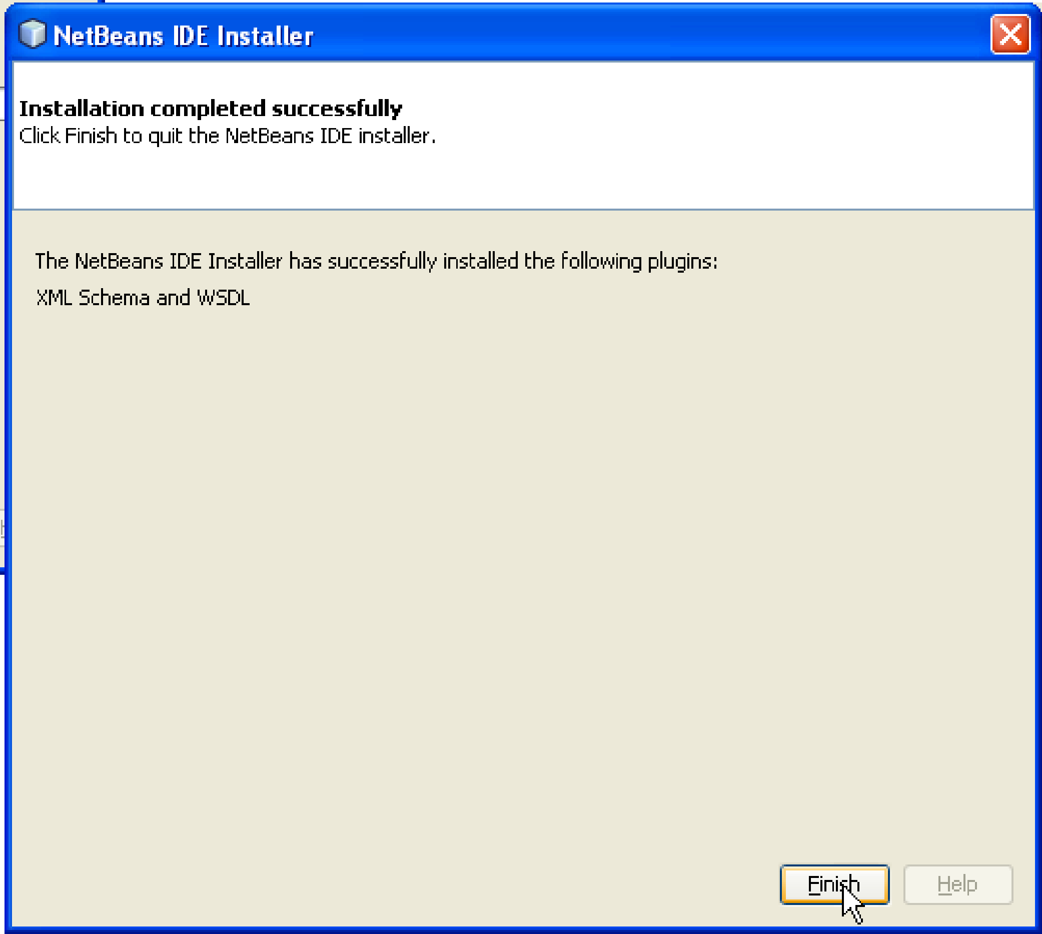
NetBeans IDE Installer

Download
Please wait until the installer downloads the requested plugins.

Downloading plugins...

0%

XML Schema and WSDL

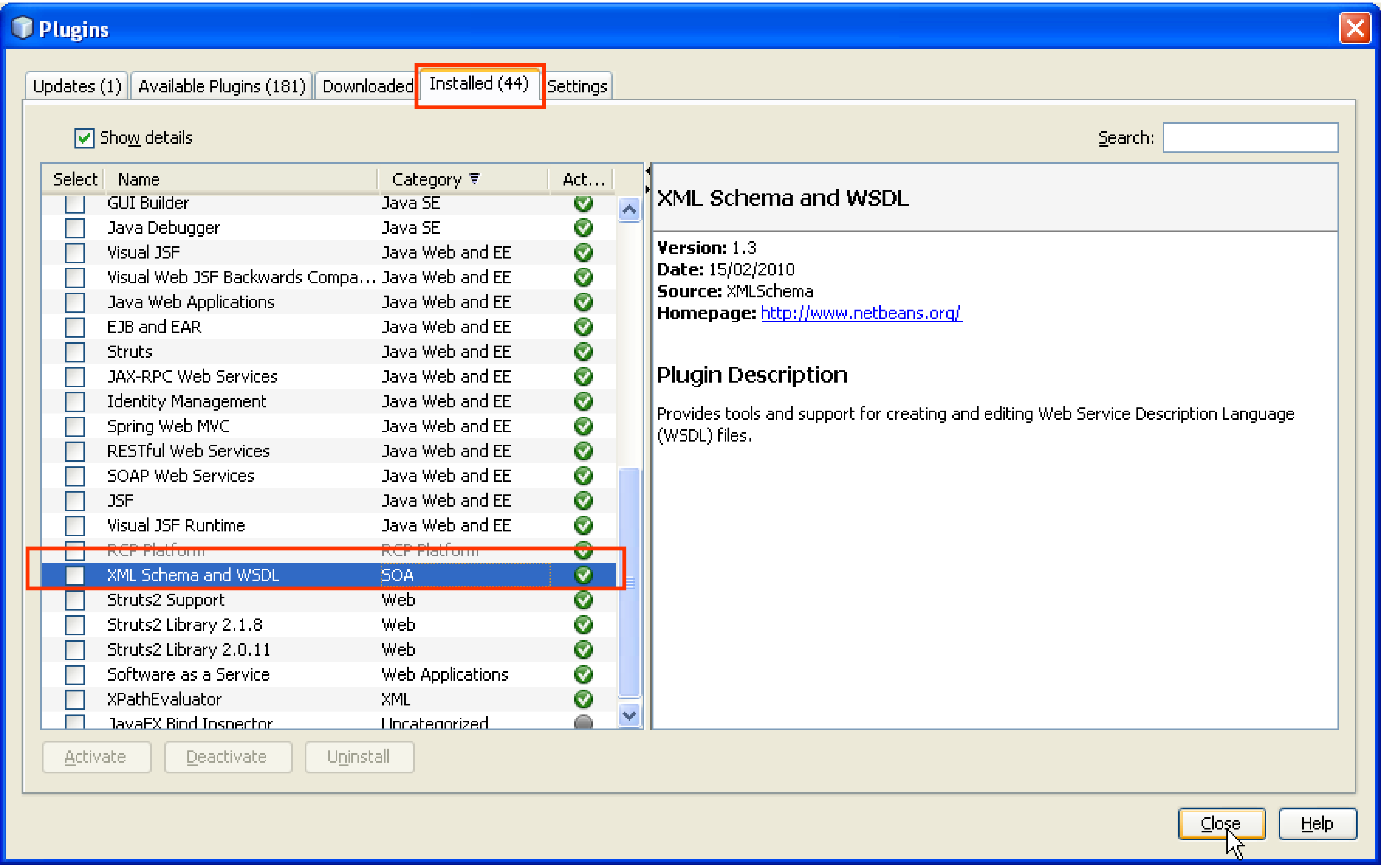


NetBeans IDE Installer

Installation completed successfully
Click Finish to quit the NetBeans IDE installer.

The NetBeans IDE Installer has successfully installed the following plugins:
XML Schema and WSDL

Schemas
Add Plugin support Schema in Netbeans



Select	Name	Category	Act...
☐	GUI Builder	Java SE	✓
☐	Java Debugger	Java SE	✓
☐	Visual JSF	Java Web and EE	✓
☐	Visual Web JSF Backwards Compa...	Java Web and EE	✓
☐	Java Web Applications	Java Web and EE	✓
☐	EJB and EAR	Java Web and EE	✓
☐	Struts	Java Web and EE	✓
☐	JAX-RPC Web Services	Java Web and EE	✓
☐	Identity Management	Java Web and EE	✓
☐	Spring Web MVC	Java Web and EE	✓
☐	RESTful Web Services	Java Web and EE	✓
☐	SOAP Web Services	Java Web and EE	✓
☐	JSF	Java Web and EE	✓
☐	Visual JSF Runtime	Java Web and EE	✓
☐	RCP Platform	RCP Platform	✓
☐	XML Schema and WSDL	SOA	✓
☐	Struts2 Support	Web	✓
☐	Struts2 Library 2.1.8	Web	✓
☐	Struts2 Library 2.0.11	Web	✓
☐	Software as a Service	Web Applications	✓
☐	XPathEvaluator	XML	✓
☐	JavaFX Bind Inspector	Uncategorized	●

XML Schema and WSDL

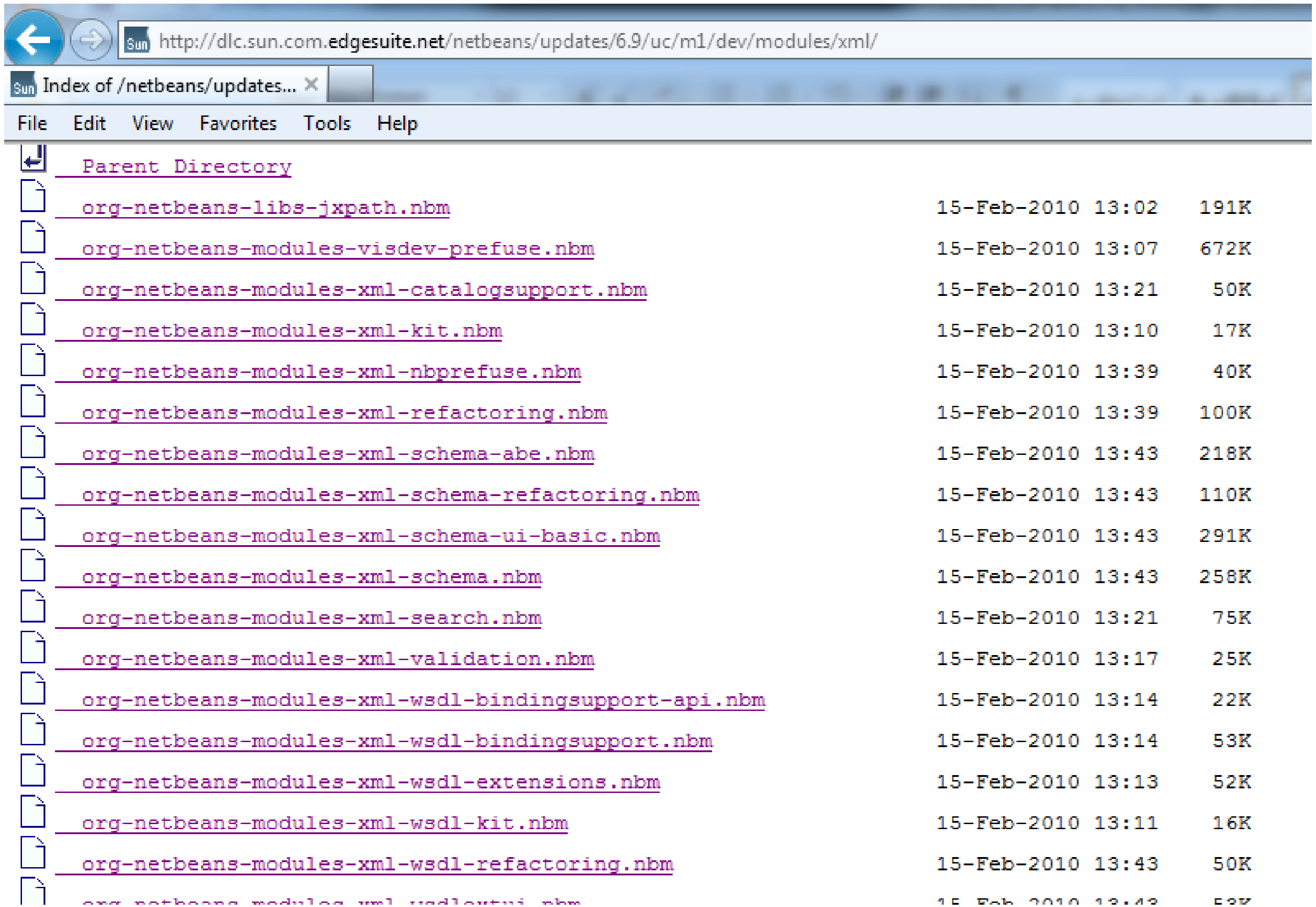
Version: 1.3
Date: 15/02/2010
Source: XMLSchema
Homepage: http://www.netbeans.org/

Plugin Description

Provides tools and support for creating and editing Web Service Description Language (WSDL) files.

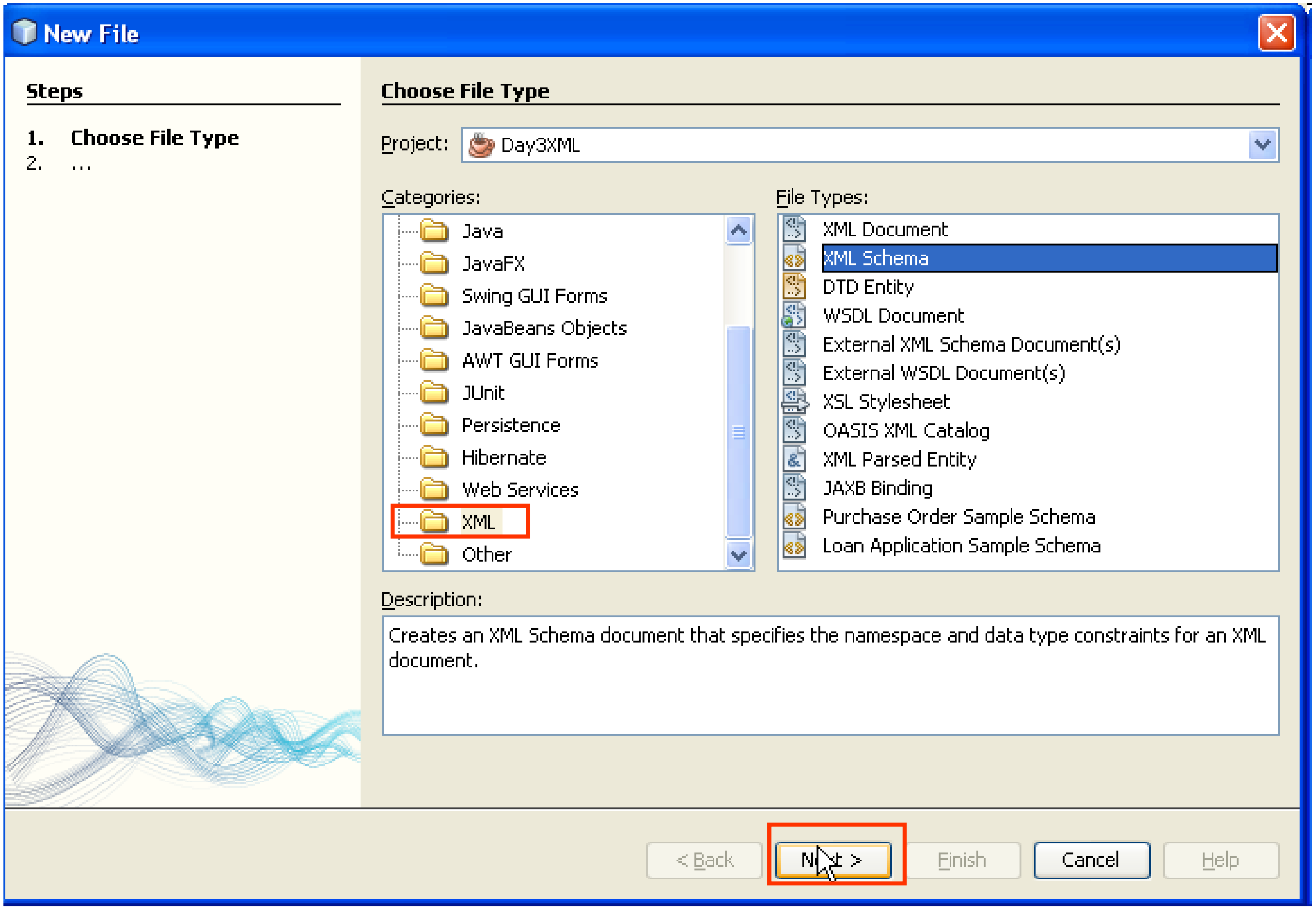
Schemas
Add Plugin support Schema in Netbeans

- Other ways
  - Download file from http://dlc.sun.com.edgesuite.net/netbeans/updates/6.9/uc/m1/dev/modules/xml/
  - Then, copy all to the .netbeans\6.9\update\download, and restart the Netbeans


http://dlc.sun.com.edgesuite.net/netbeans/updates/6.9/uc/m1/dev/modules/xml/
Index of /netbeans/updates...
File Edit View Favorites Tools Help

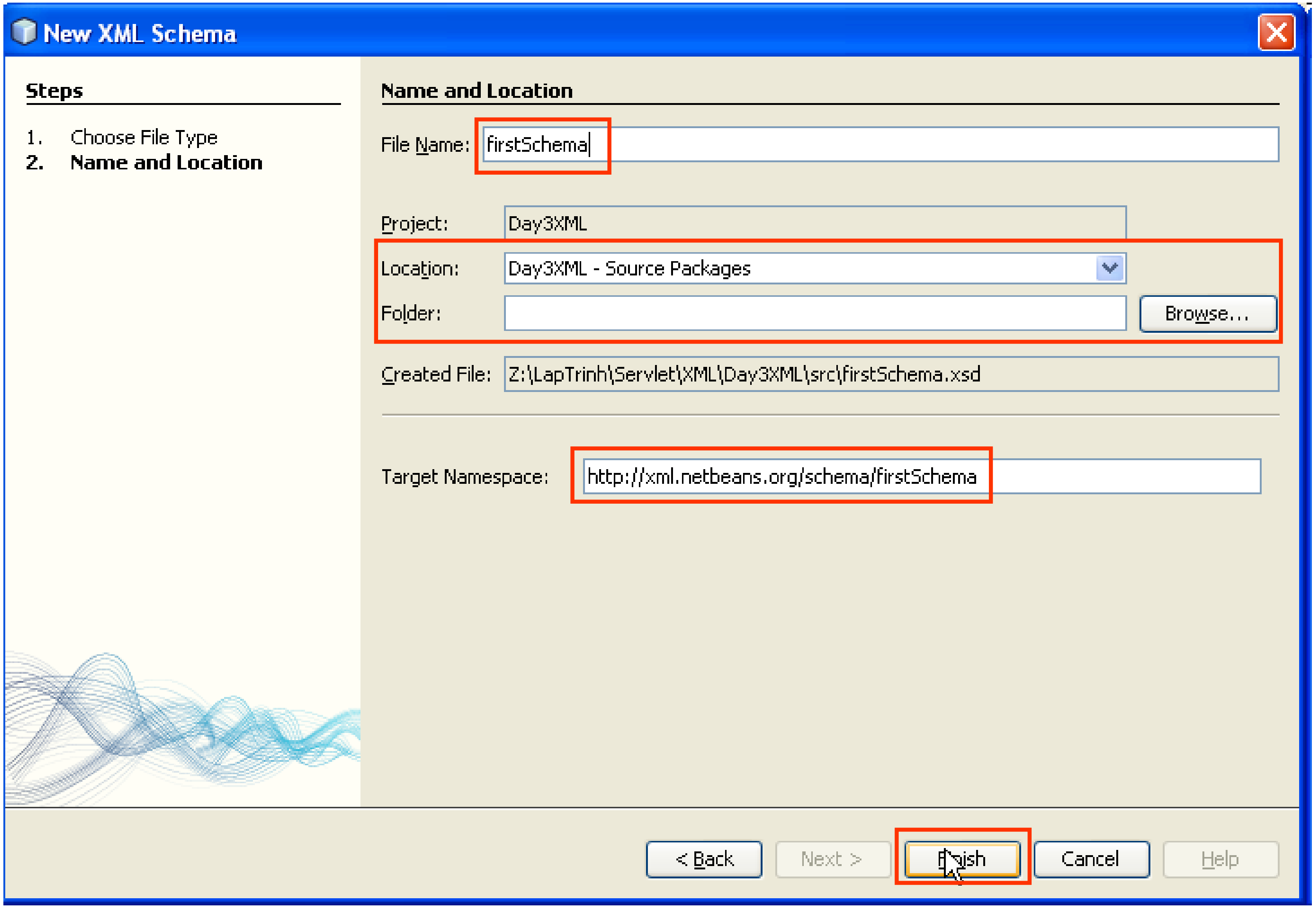
```
Parent Directory
org-netbeans-libs-jxpath.nbm                          15-Feb-2010 13:02   191K
org-netbeans-modules-visdev-prefuse.nbm               15-Feb-2010 13:07   672K
org-netbeans-modules-xml-catalogsupport.nbm           15-Feb-2010 13:21    50K
org-netbeans-modules-xml-kit.nbm                      15-Feb-2010 13:10    17K
org-netbeans-modules-xml-nbprefuse.nbm                15-Feb-2010 13:39    40K
org-netbeans-modules-xml-refactoring.nbm              15-Feb-2010 13:39   100K
org-netbeans-modules-xml-schema-abe.nbm               15-Feb-2010 13:43   218K
org-netbeans-modules-xml-schema-refactoring.nbm       15-Feb-2010 13:43   110K
org-netbeans-modules-xml-schema-ui-basic.nbm          15-Feb-2010 13:43   291K
org-netbeans-modules-xml-schema.nbm                   15-Feb-2010 13:43   258K
org-netbeans-modules-xml-search.nbm                   15-Feb-2010 13:21    75K
org-netbeans-modules-xml-validation.nbm               15-Feb-2010 13:17    25K
org-netbeans-modules-xml-wsdl-bindingsupport-api.nbm  15-Feb-2010 13:14    22K
org-netbeans-modules-xml-wsdl-bindingsupport.nbm      15-Feb-2010 13:14    53K
org-netbeans-modules-xml-wsdl-extensions.nbm          15-Feb-2010 13:13    52K
org-netbeans-modules-xml-wsdl-kit.nbm                 15-Feb-2010 13:11    16K
org-netbeans-modules-xml-wsdl-refactoring.nbm         15-Feb-2010 13:43    50K
org-netbeans-modules-xml-wsdlextui.nbm                15-Feb-2010 13:43    53K
```

Schemas
How to write Schema in Netbeans



Creates an XML Schema document that specifies the namespace and data type constraints for an XML document.

Schemas
How to write Schema in Netbeans



New XML Schema

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name: firstSchema

Project: Day3XML

Location: Day3XML - Source Packages

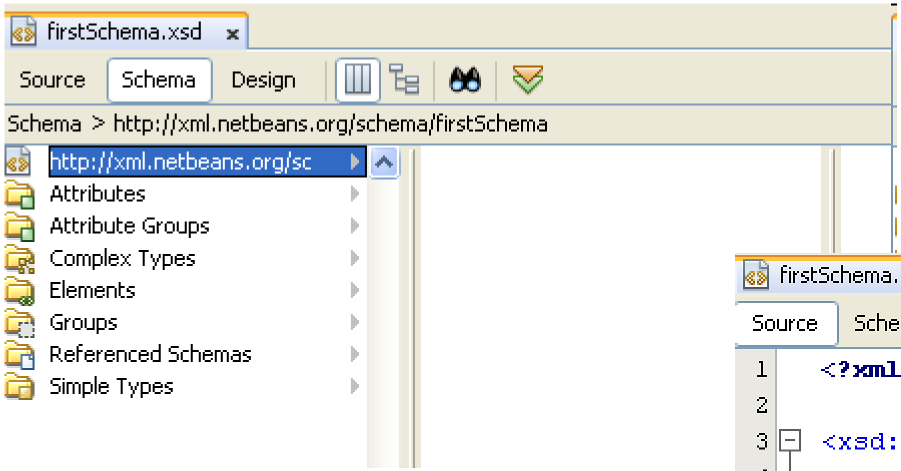
Folder:

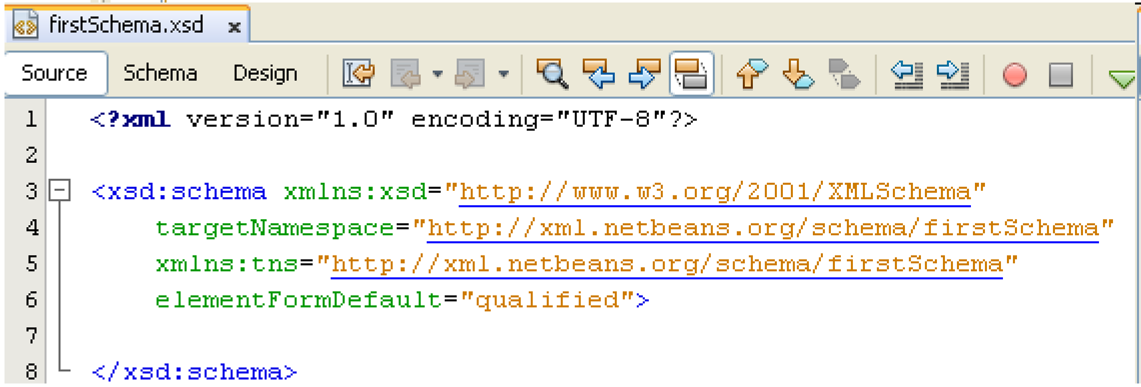
Created File: Z:\LapTrinh\Servlet\XML\Day3XML\src\firstSchema.xsd

Target Namespace: http://xml.netbeans.org/schema/firstSchema

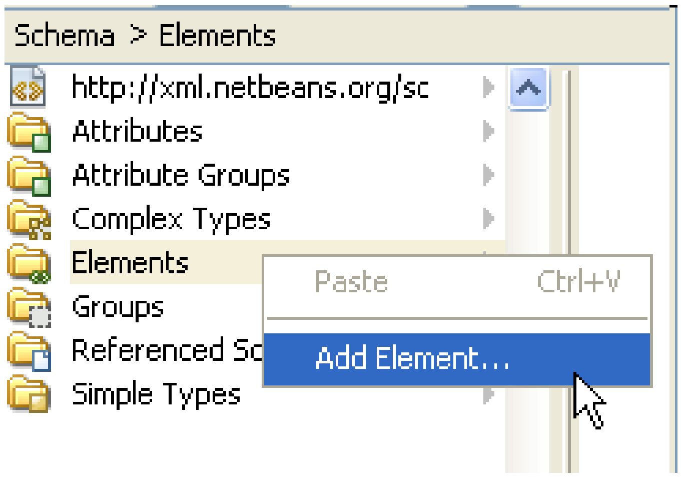
Schemas

How to write Schema in Netbeans



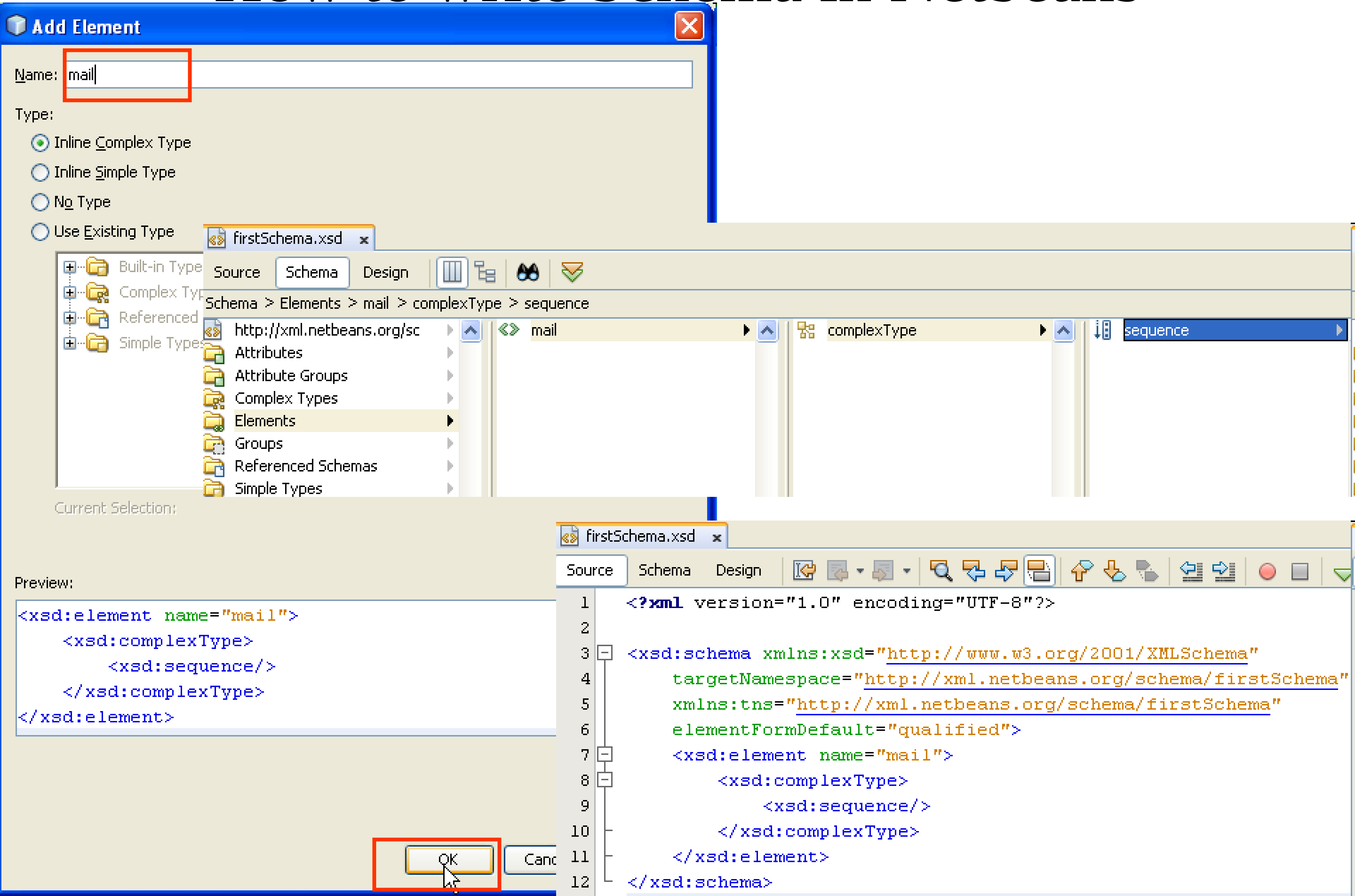


```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/firstSchema"
    xmlns:tns="http://xml.netbeans.org/schema/firstSchema"
    elementFormDefault="qualified">

</xsd:schema>
```



Click right mouse

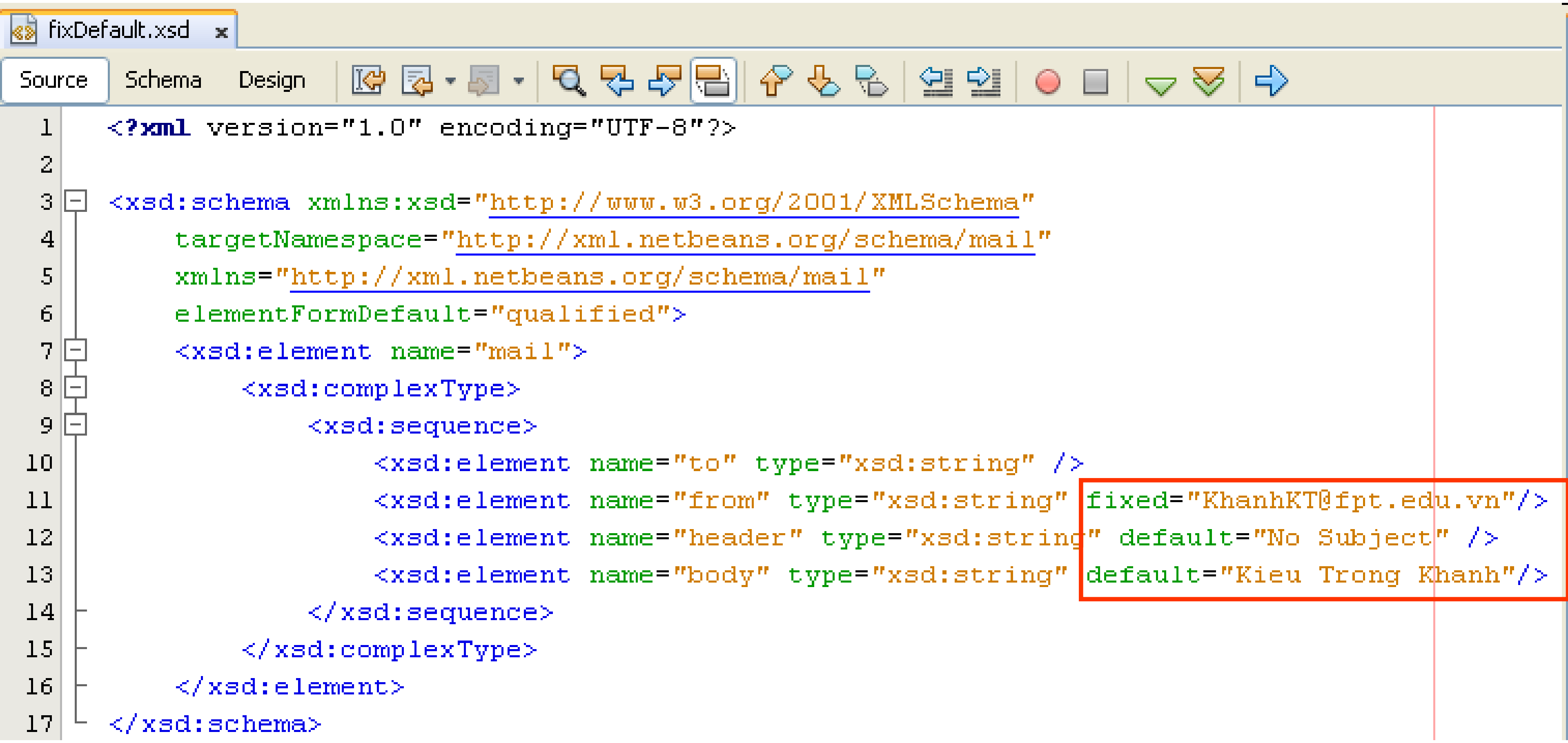
Schemas
How to write Schema in Netbeans



```
<xsd:element name="mail">
    <xsd:complexType>
        <xsd:sequence/>
    </xsd:complexType>
</xsd:element>
```

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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/firstSchema"
    xmlns:tns="http://xml.netbeans.org/schema/firstSchema"
    elementFormDefault="qualified">
    <xsd:element name="mail">
        <xsd:complexType>
            <xsd:sequence/>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

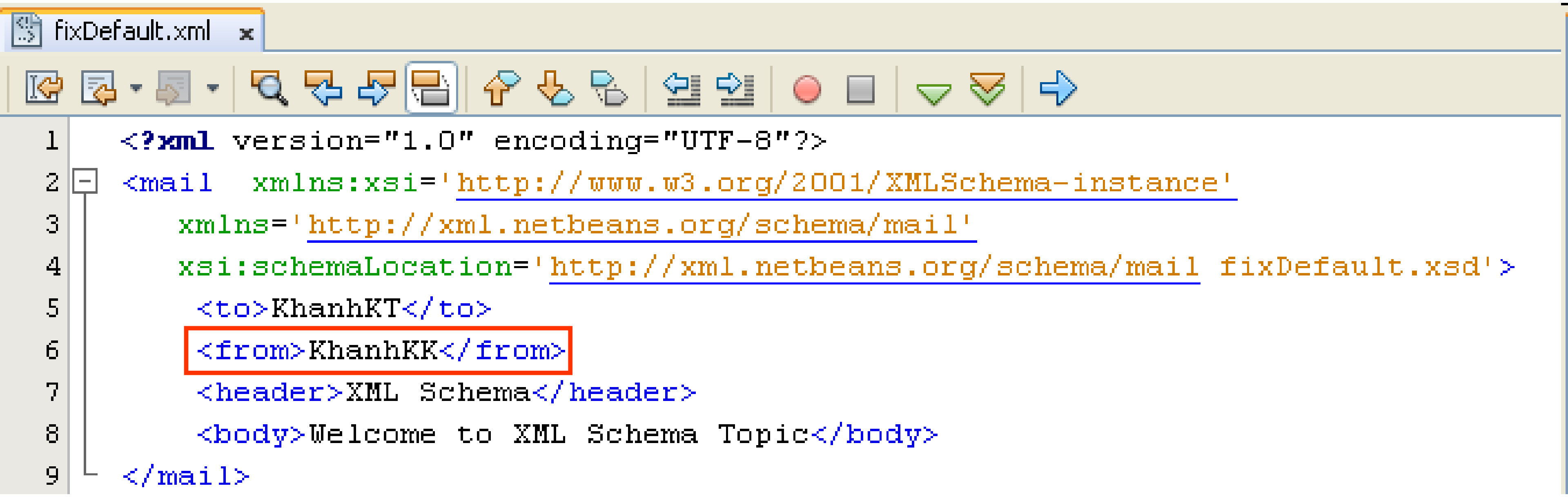
Schemas
Structuring–XSD Data Types–string–Example


fixDefault.xsd

Source Schema Design

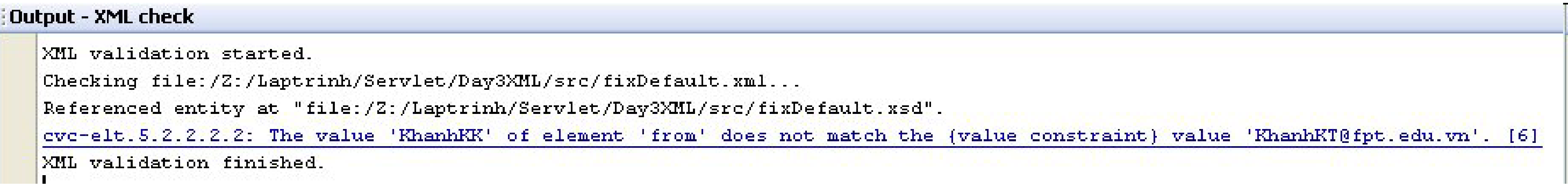
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/mail"
    xmlns="http://xml.netbeans.org/schema/mail"
    elementFormDefault="qualified">
    <xsd:element name="mail">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="to" type="xsd:string" />
                <xsd:element name="from" type="xsd:string" fixed="KhanhKT@fpt.edu.vn"/>
                <xsd:element name="header" type="xsd:string" default="No Subject" />
                <xsd:element name="body" type="xsd:string" default="Kieu Trong Khanh"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas
Structuring–XSD Data Types–string–Example



fixDefault.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<mail  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/mail'
    xsi:schemaLocation='http://xml.netbeans.org/schema/mail fixDefault.xsd'>
     <to>KhanhKT</to>
     <from>KhanhKK</from>
     <header>XML Schema</header>
     <body>Welcome to XML Schema Topic</body>
</mail>
```

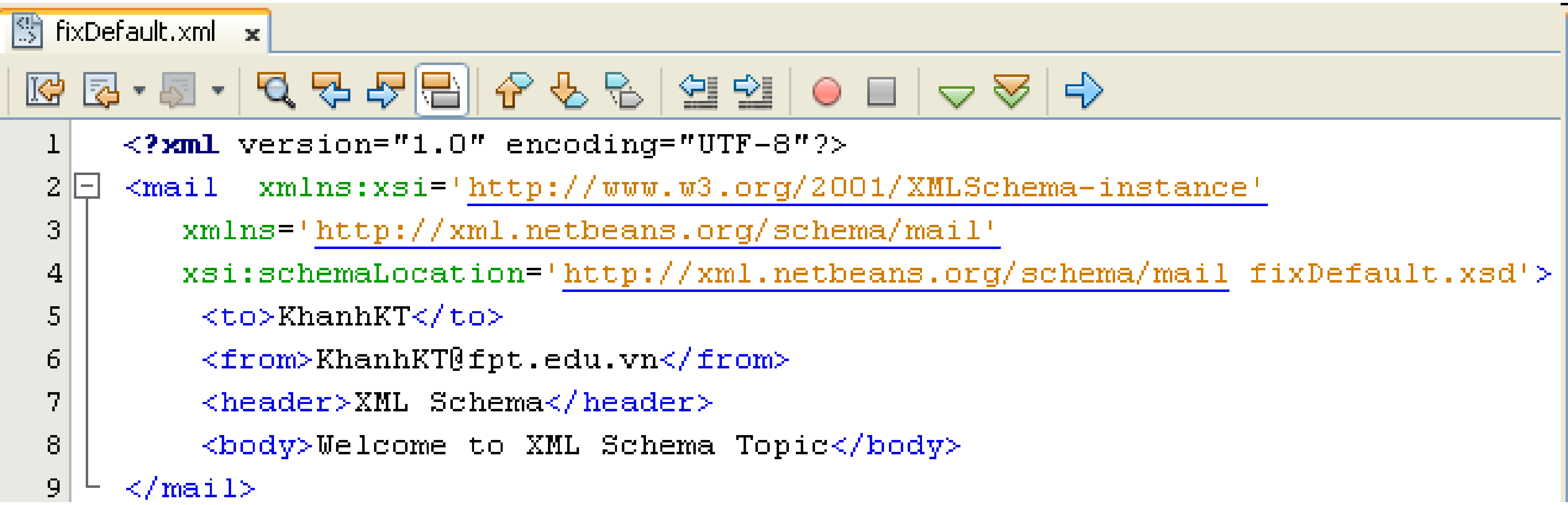


Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/fixDefault.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/fixDefault.xsd".
cvc-elt.5.2.2.2.2: The value 'KhanhKK' of element 'from' does not match the {value constraint} value 'KhanhKT@fpt.edu.vn'. [6]
XML validation finished.
```

Schemas

Structuring–XSD Data Types–string–Example

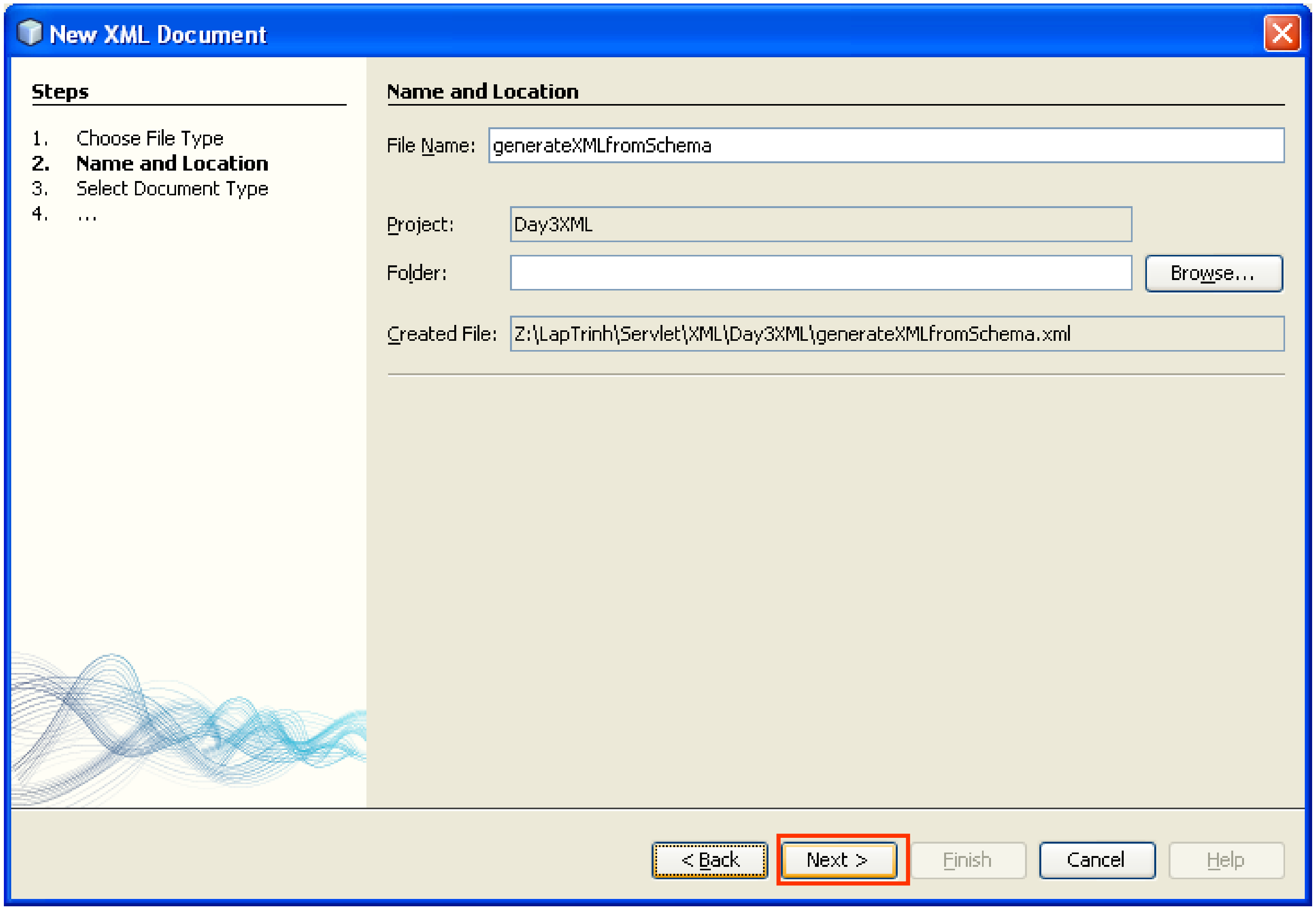


fixDefault.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<mail  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/mail'
    xsi:schemaLocation='http://xml.netbeans.org/schema/mail fixDefault.xsd'>
     <to>KhanhKT</to>
     <from>KhanhKT@fpt.edu.vn</from>
     <header>XML Schema</header>
     <body>Welcome to XML Schema Topic</body>
</mail>
```

Schemas

How to generate XML from Schema in Netbeans


New XML Document

Steps

1. Choose File Type
2. **Name and Location**
3. Select Document Type
4. ...

Name and Location

File Name: generateXMLfromSchema

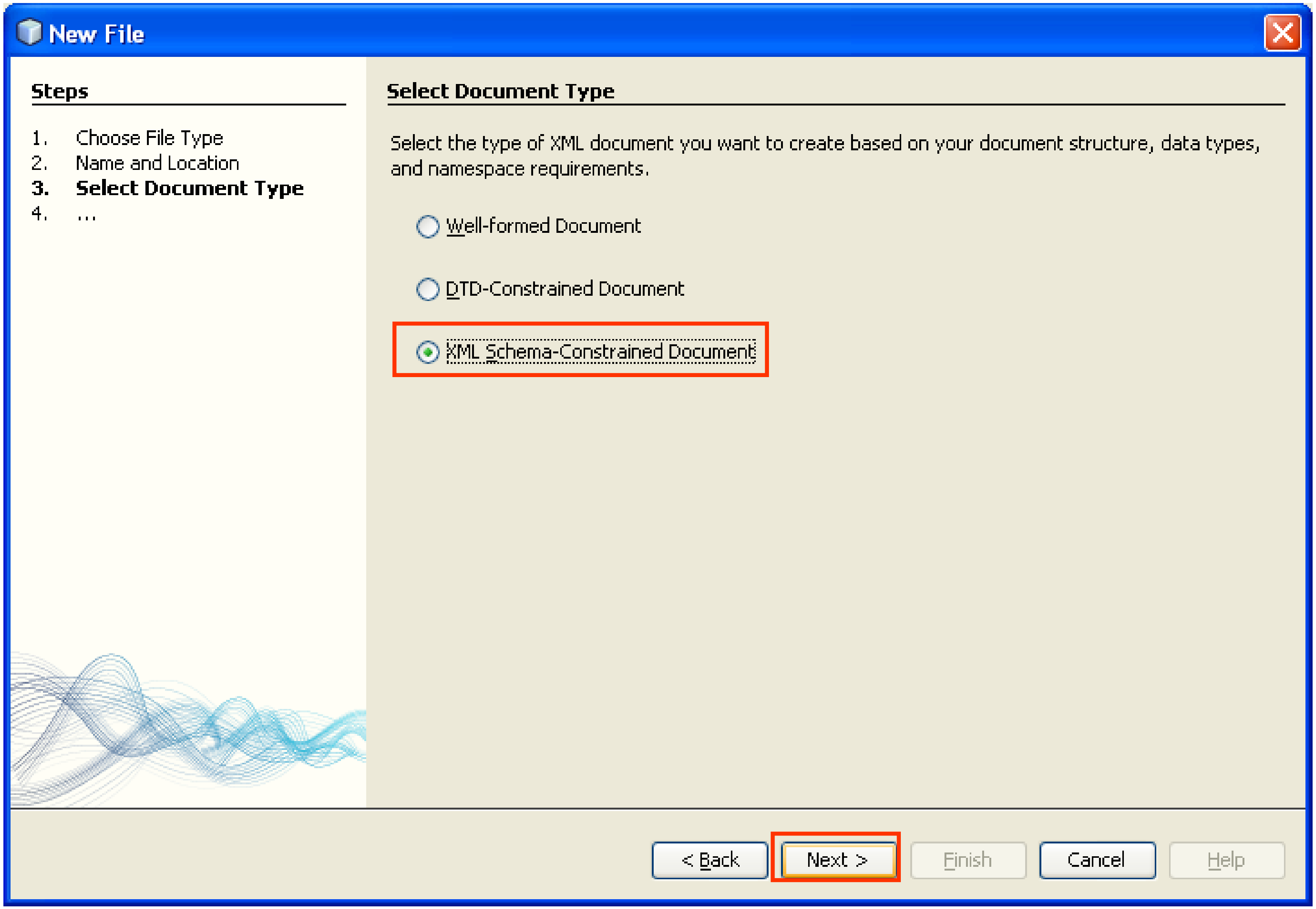
Project: Day3XML

Folder:

Created File: Z:\LapTrinh\Servlet\XML\Day3XML\generateXMLfromSchema.xml

Schemas

How to generate XML from Schema in Netbeans



New File

Steps

1. Choose File Type
2. Name and Location
3. **Select Document Type**
4. ...

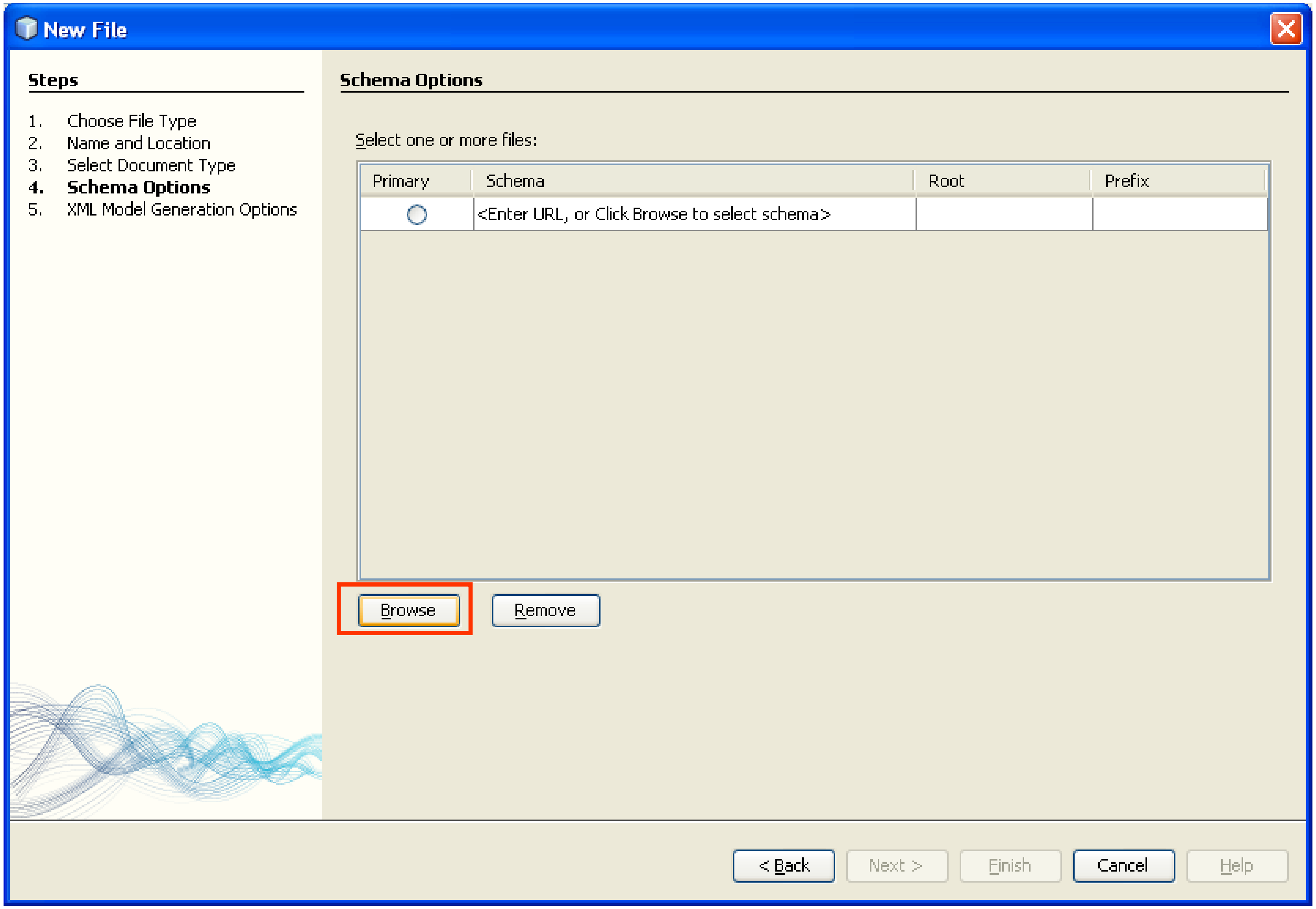
Select Document Type

Select the type of XML document you want to create based on your document structure, data types, and namespace requirements.

- ◯ Well-formed Document
- ◯ DTD-Constrained Document
- ● XML Schema-Constrained Document

Schemas

How to generate XML from Schema in Netbeans


New File

Steps

1. Choose File Type
2. Name and Location
3. Select Document Type
4. **Schema Options**
5. XML Model Generation Options

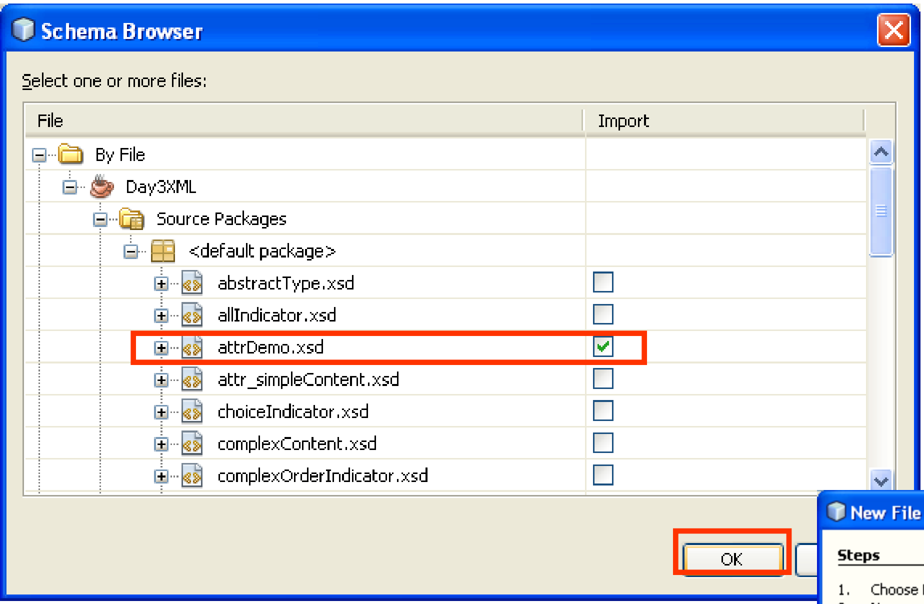
Schema Options

Select one or more files:

Primary	Schema	Root	Prefix
○	<Enter URL, or Click Browse to select schema>		

Schemas

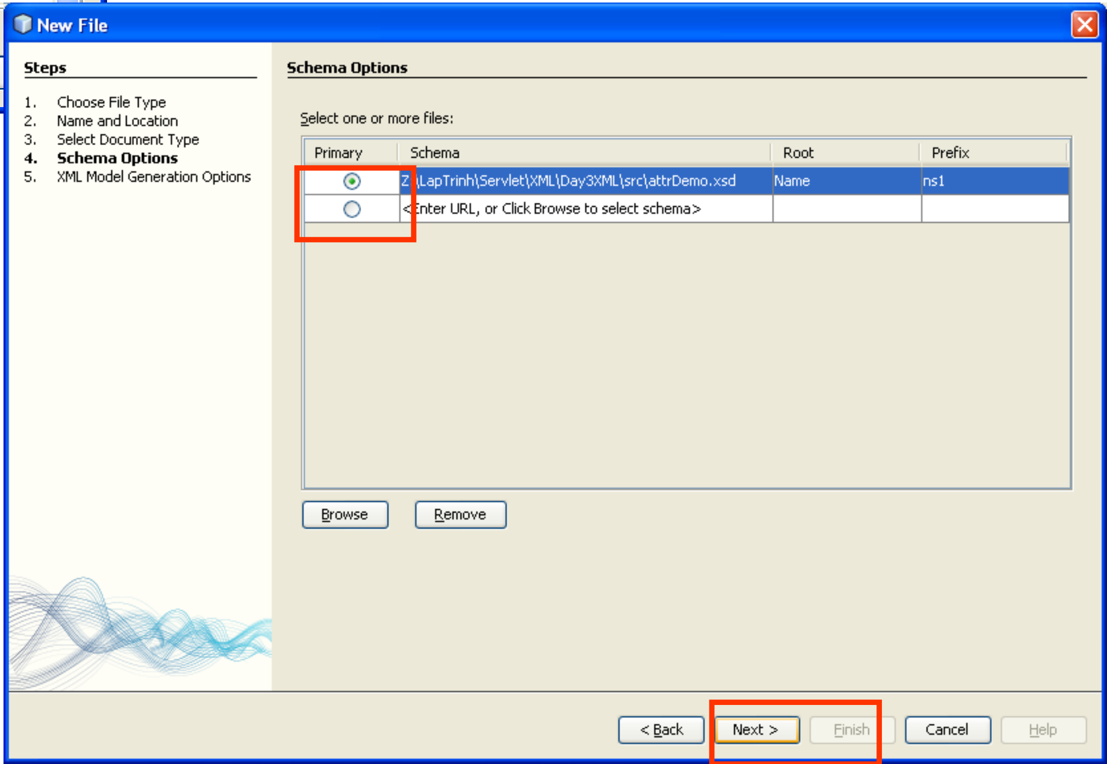
How to generate XML from Schema in Netbeans



Schema Browser

Select one or more files:

File	Import
By File	
Day3XML	
Source Packages	
<default package>	
abstractType.xsd	☐
allIndicator.xsd	☐
attrDemo.xsd	✓
attr_simpleContent.xsd	☐
choiceIndicator.xsd	☐
complexContent.xsd	☐
complexOrderIndicator.xsd	☐



New File

Steps

1. Choose File Type
2. Name and Location
3. Select Document Type
4. **Schema Options**
5. XML Model Generation Options

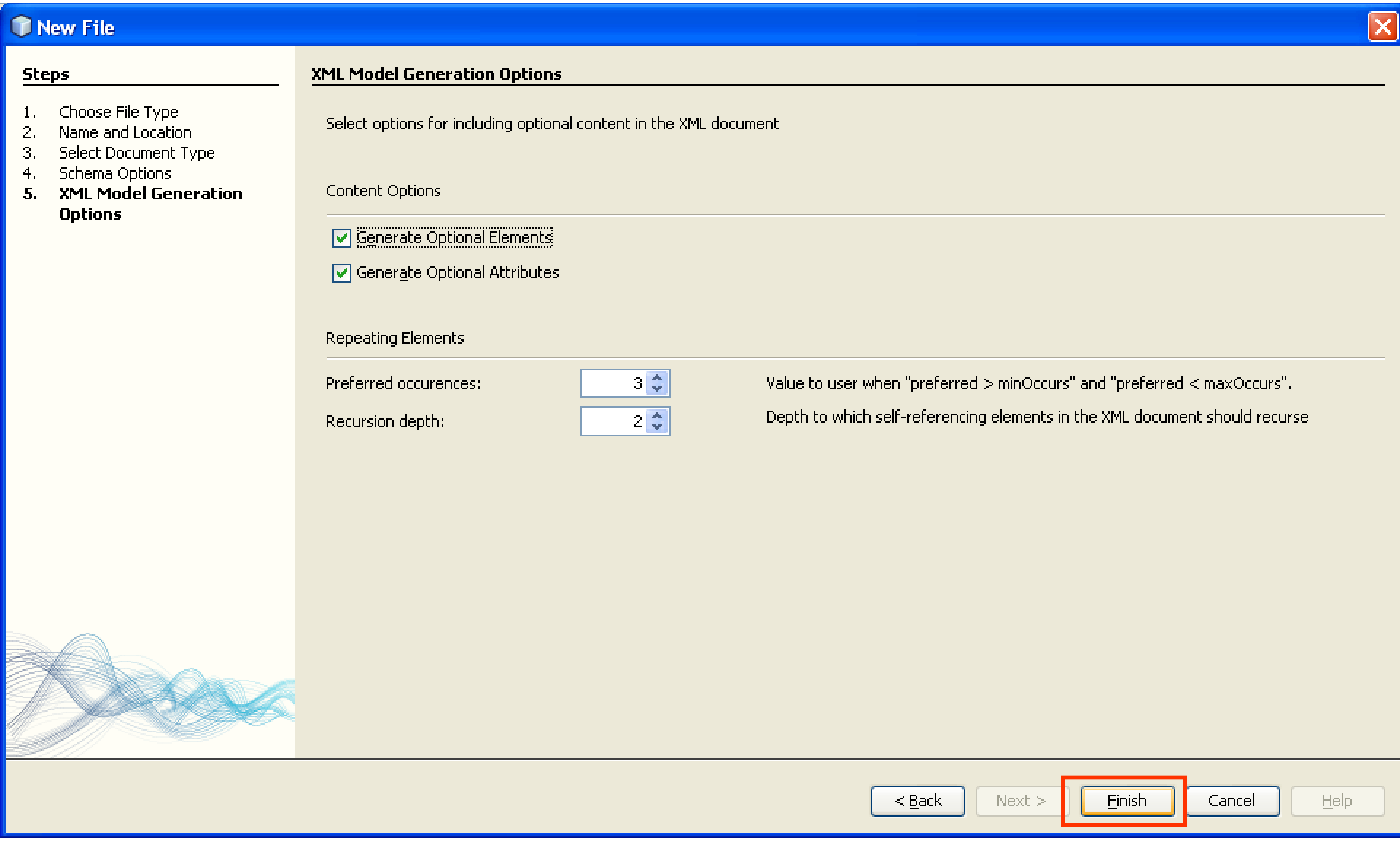
Schema Options

Select one or more files:

Primary	Schema	Root	Prefix
●	Z:\LapTrinh\Servlet\XML\Day3XML\src\attrDemo.xsd	Name	ns1
○	<Enter URL, or Click Browse to select schema>		

Schemas

How to generate XML from Schema in Netbeans



New File

Steps

1. Choose File Type
2. Name and Location
3. Select Document Type
4. Schema Options
5. **XML Model Generation Options**

XML Model Generation Options

Select options for including optional content in the XML document

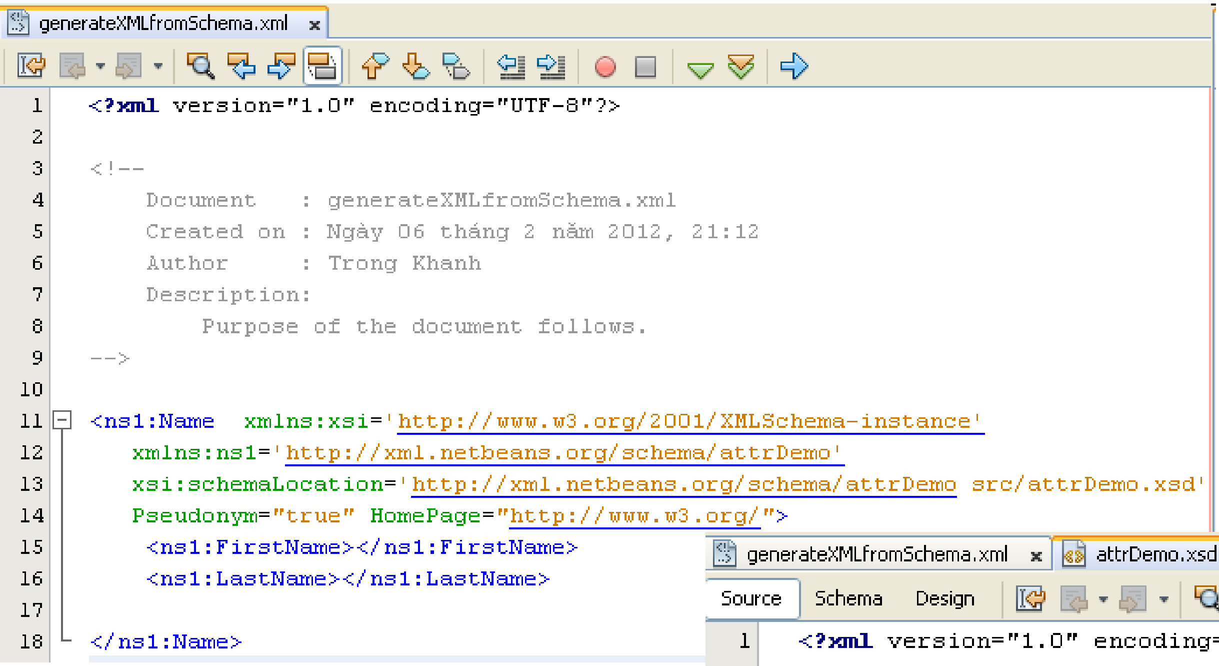
Content Options

- [✓] Generate Optional Elements
- [✓] Generate Optional Attributes

Repeating Elements

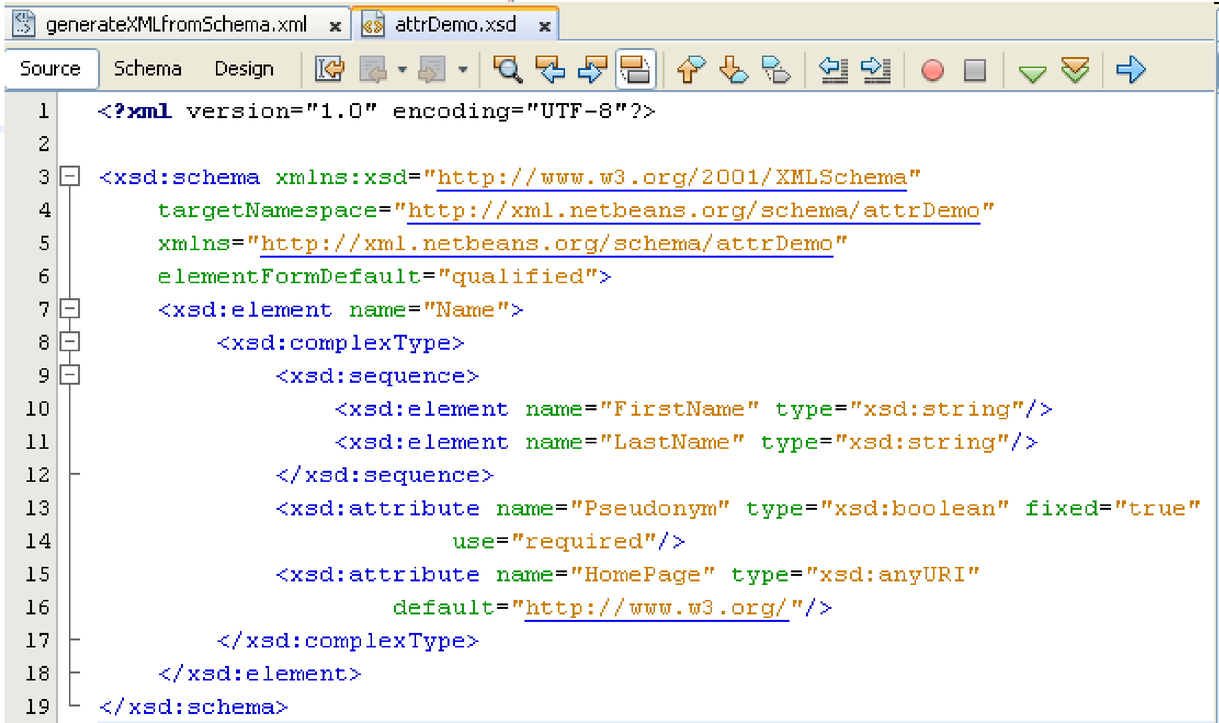
Preferred occurences:	3	Value to user when "preferred > minOccurs" and "preferred < maxOccurs".
Recursion depth:	2	Depth to which self-referencing elements in the XML document should recurse

Schemas
How to generate XML from Schema in Netbeans



generateXMLfromSchema.xml

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

<!--
    Document   : generateXMLfromSchema.xml
    Created on : Ngày 06 tháng 2 năm 2012, 21:12
    Author     : Trong Khanh
    Description:
        Purpose of the document follows.
-->

<ns1:Name  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns:ns1='http://xml.netbeans.org/schema/attrDemo'
   xsi:schemaLocation='http://xml.netbeans.org/schema/attrDemo src/attrDemo.xsd'
   Pseudonym="true" HomePage="http://www.w3.org/">
    <ns1:FirstName></ns1:FirstName>
    <ns1:LastName></ns1:LastName>

</ns1:Name>
```

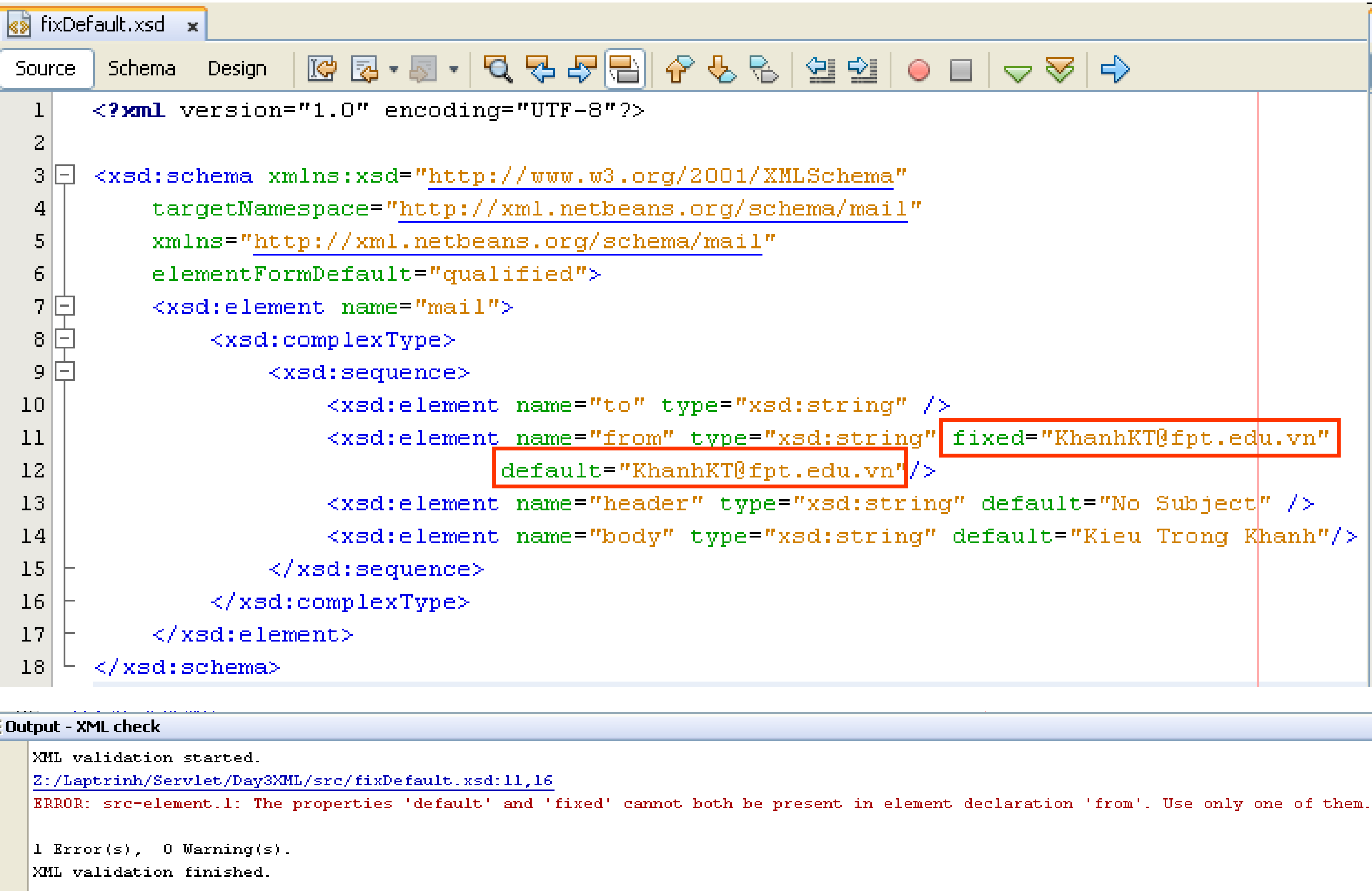


generateXMLfromSchema.xml attrDemo.xsd

Source Schema Design

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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/attrDemo"
    xmlns="http://xml.netbeans.org/schema/attrDemo"
    elementFormDefault="qualified">
    <xsd:element name="Name">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="FirstName" type="xsd:string"/>
                <xsd:element name="LastName" type="xsd:string"/>
            </xsd:sequence>
            <xsd:attribute name="Pseudonym" type="xsd:boolean" fixed="true"
                        use="required"/>
            <xsd:attribute name="HomePage" type="xsd:anyURI"
                    default="http://www.w3.org/"/>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

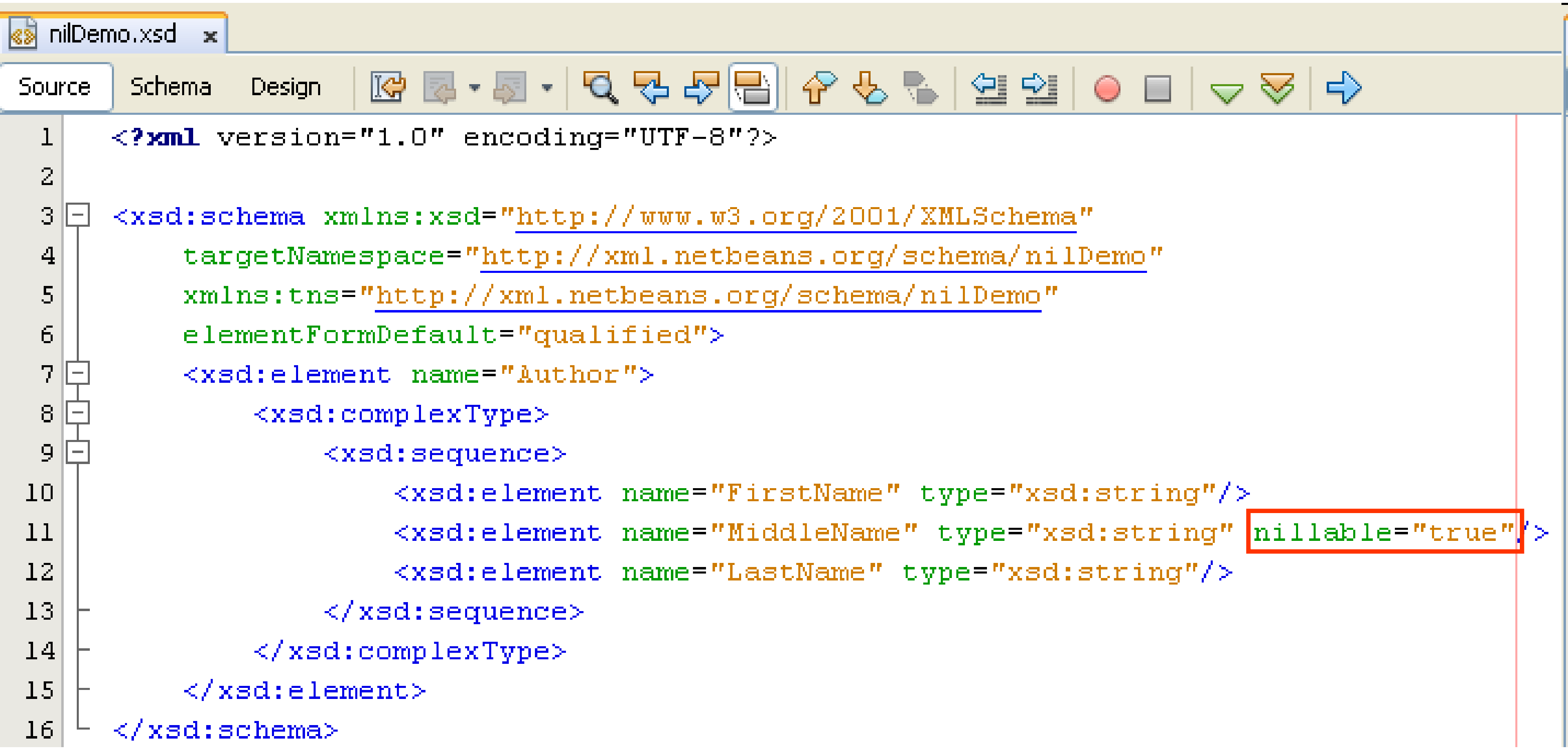
Schemas
Structuring–XSD Data Types–string–Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/mail"
    xmlns="http://xml.netbeans.org/schema/mail"
    elementFormDefault="qualified">
    <xsd:element name="mail">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="to" type="xsd:string" />
                <xsd:element name="from" type="xsd:string" fixed="KhanhKT@fpt.edu.vn"
                        default="KhanhKT@fpt.edu.vn"/>
                <xsd:element name="header" type="xsd:string" default="No Subject" />
                <xsd:element name="body" type="xsd:string" default="Kieu Trong Khanh"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

```
XML validation started.
Z:/Laptrinh/Servlet/Day3XML/src/fixDefault.xsd:11,16
ERROR: src-element.1: The properties 'default' and 'fixed' cannot both be present in element declaration 'from'. Use only one of them.

1 Error(s),  0 Warning(s).
XML validation finished.
```

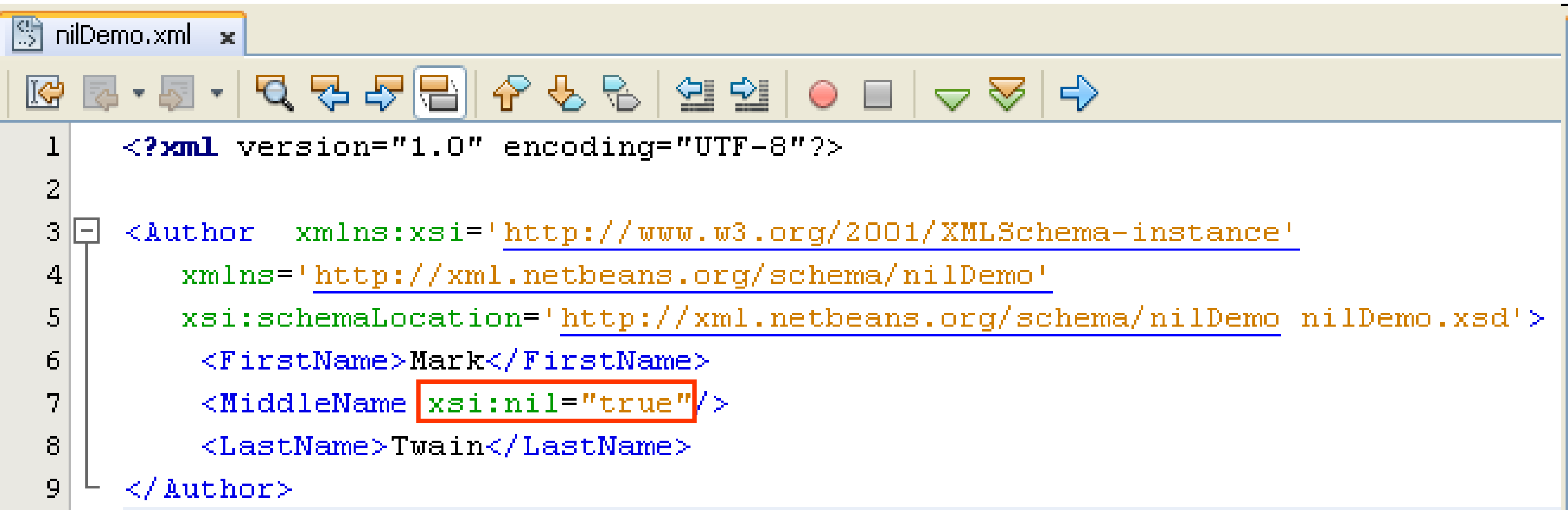
Schemas
Structuring–XSD Data Types–string–Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/nilDemo"
    xmlns:tns="http://xml.netbeans.org/schema/nilDemo"
    elementFormDefault="qualified">
    <xsd:element name="Author">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="FirstName" type="xsd:string"/>
                <xsd:element name="MiddleName" type="xsd:string" nillable="true"/>
                <xsd:element name="LastName" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

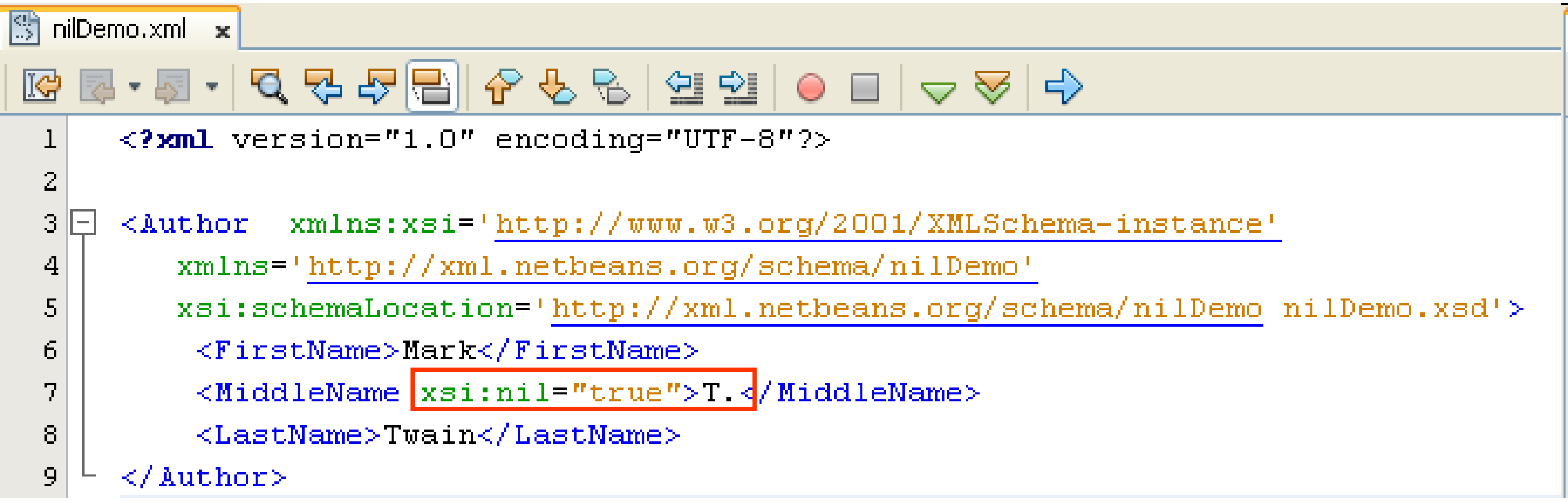
Schemas

Structuring–XSD Data Types–string–Example



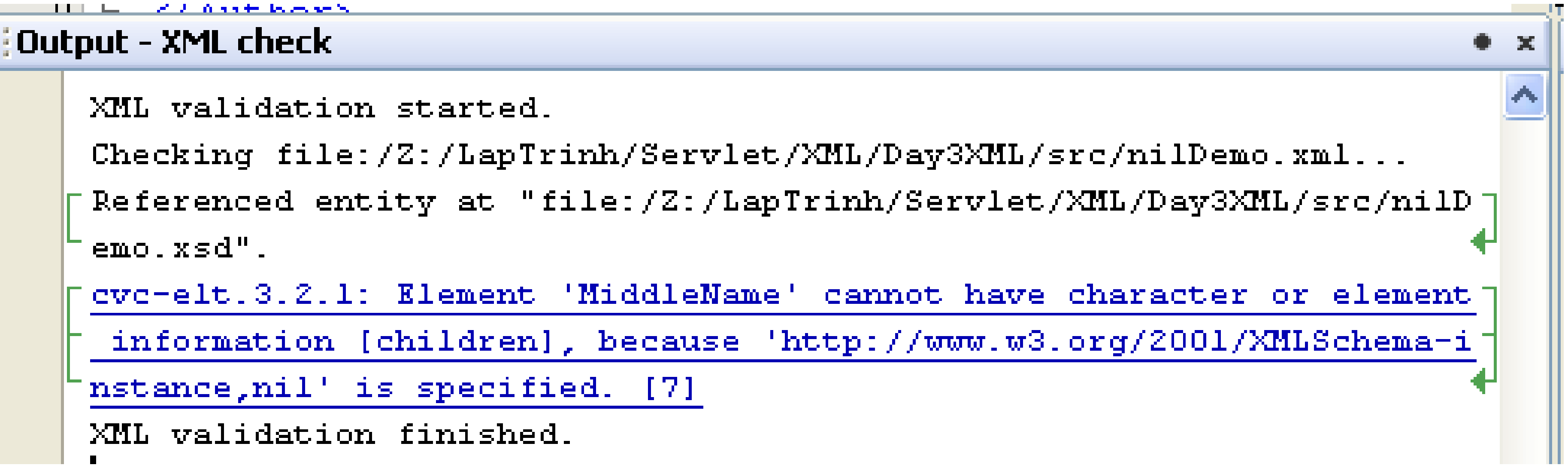
```
<?xml version="1.0" encoding="UTF-8"?>

<Author  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/nilDemo'
    xsi:schemaLocation='http://xml.netbeans.org/schema/nilDemo nilDemo.xsd'>
     <FirstName>Mark</FirstName>
     <MiddleName xsi:nil="true"/>
     <LastName>Twain</LastName>
</Author>
```

Schemas
Structuring–XSD Data Types–string–Example



nilDemo.xml

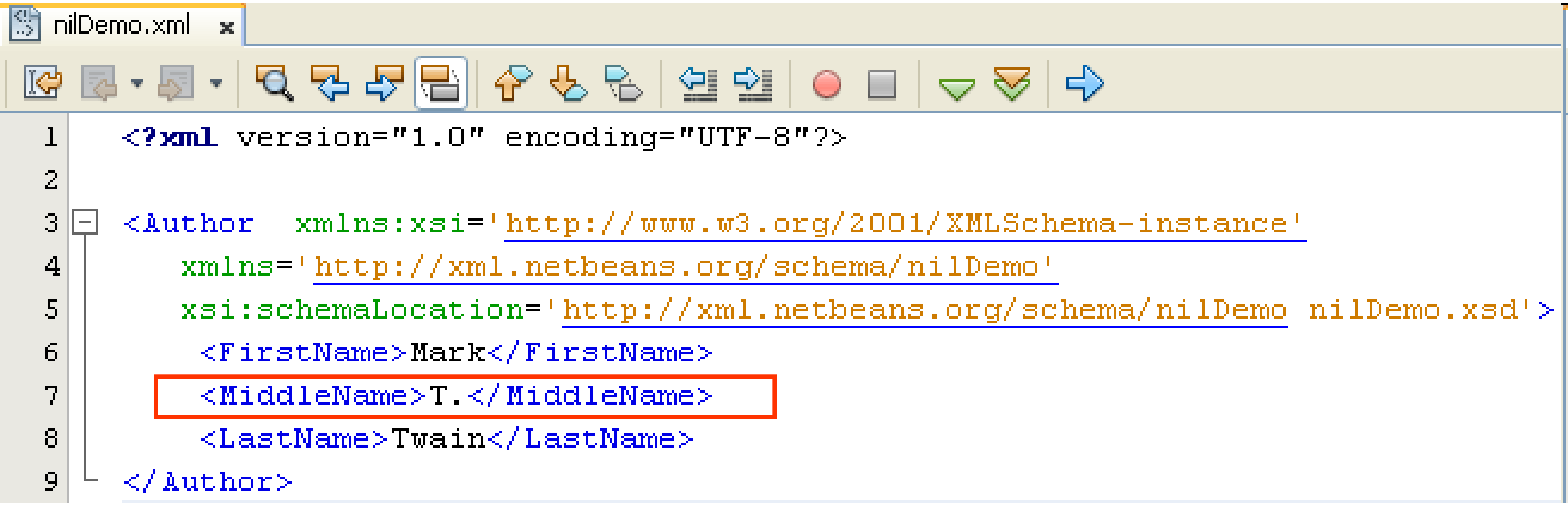
```
<?xml version="1.0" encoding="UTF-8"?>

<Author  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/nilDemo'
    xsi:schemaLocation='http://xml.netbeans.org/schema/nilDemo nilDemo.xsd'>
     <FirstName>Mark</FirstName>
     <MiddleName xsi:nil="true">T.</MiddleName>
     <LastName>Twain</LastName>
</Author>
```



Output - XML check

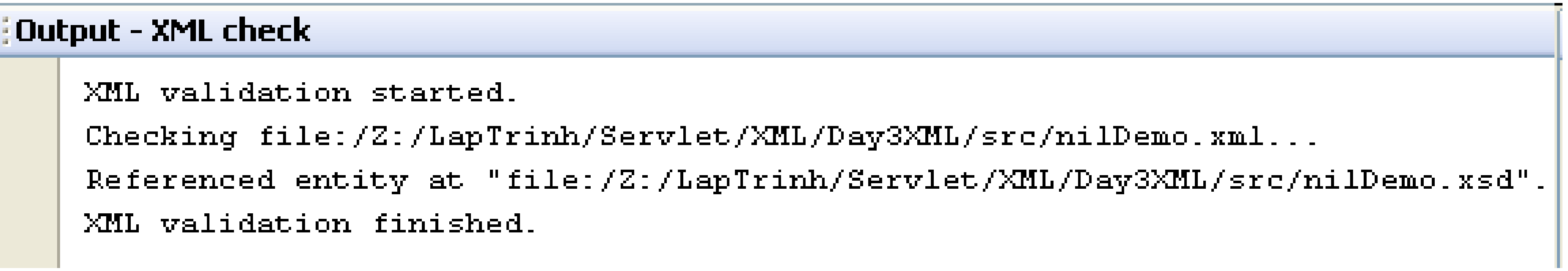
```
XML validation started.
Checking file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/nilDemo.xml...
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/nilD
emo.xsd".
cvc-elt.3.2.1: Element 'MiddleName' cannot have character or element
 information [children], because 'http://www.w3.org/2001/XMLSchema-i
nstance,nil' is specified. [7]
XML validation finished.
```

Schemas
Structuring–XSD Data Types–string–Example



nilDemo.xml

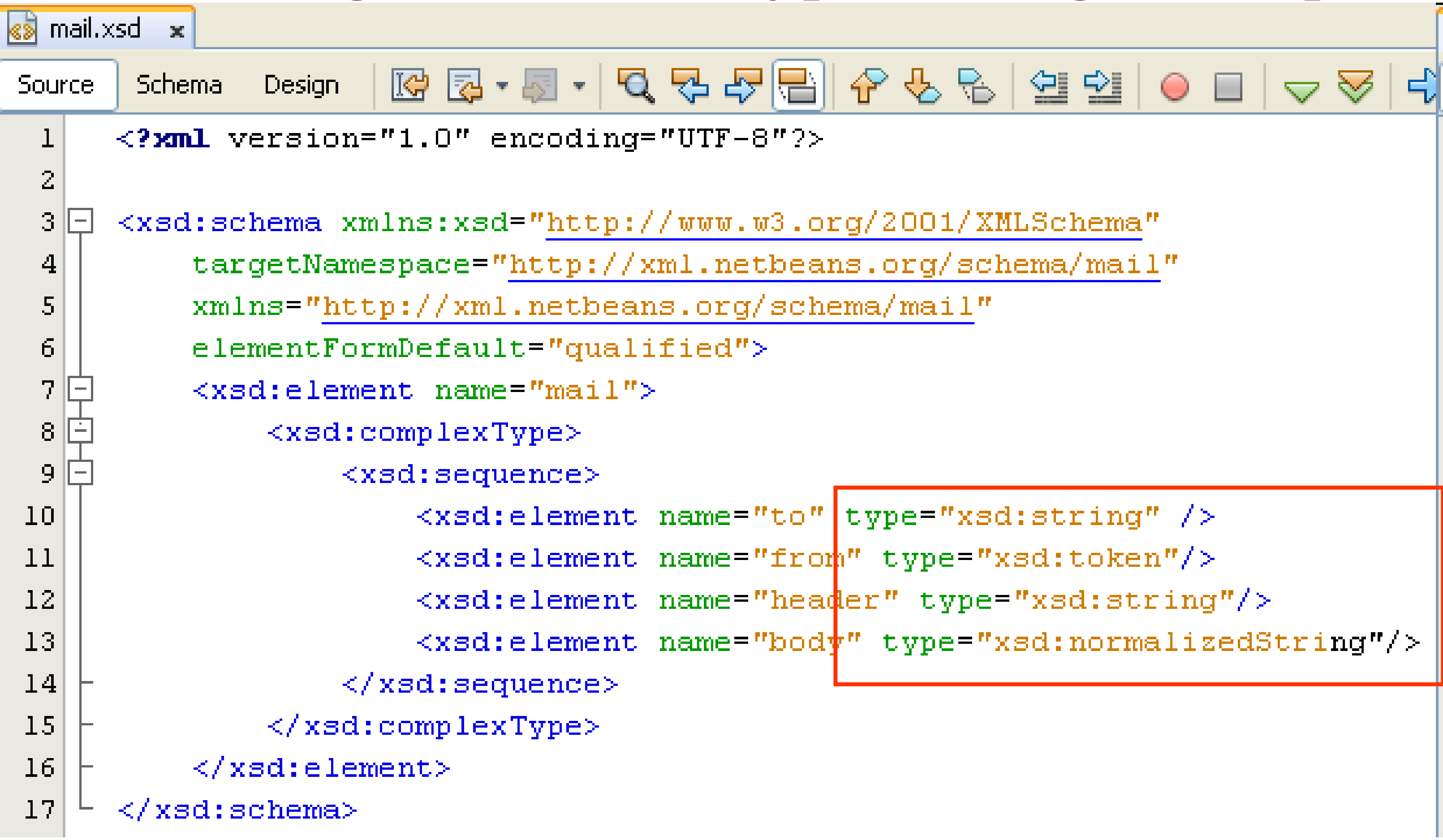
```
<?xml version="1.0" encoding="UTF-8"?>

<Author  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/nilDemo'
    xsi:schemaLocation='http://xml.netbeans.org/schema/nilDemo nilDemo.xsd'>
     <FirstName>Mark</FirstName>
     <MiddleName>T.</MiddleName>
     <LastName>Twain</LastName>
</Author>
```



Output - XML check

```
XML validation started.
Checking file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/nilDemo.xml...
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/nilDemo.xsd".
XML validation finished.
```

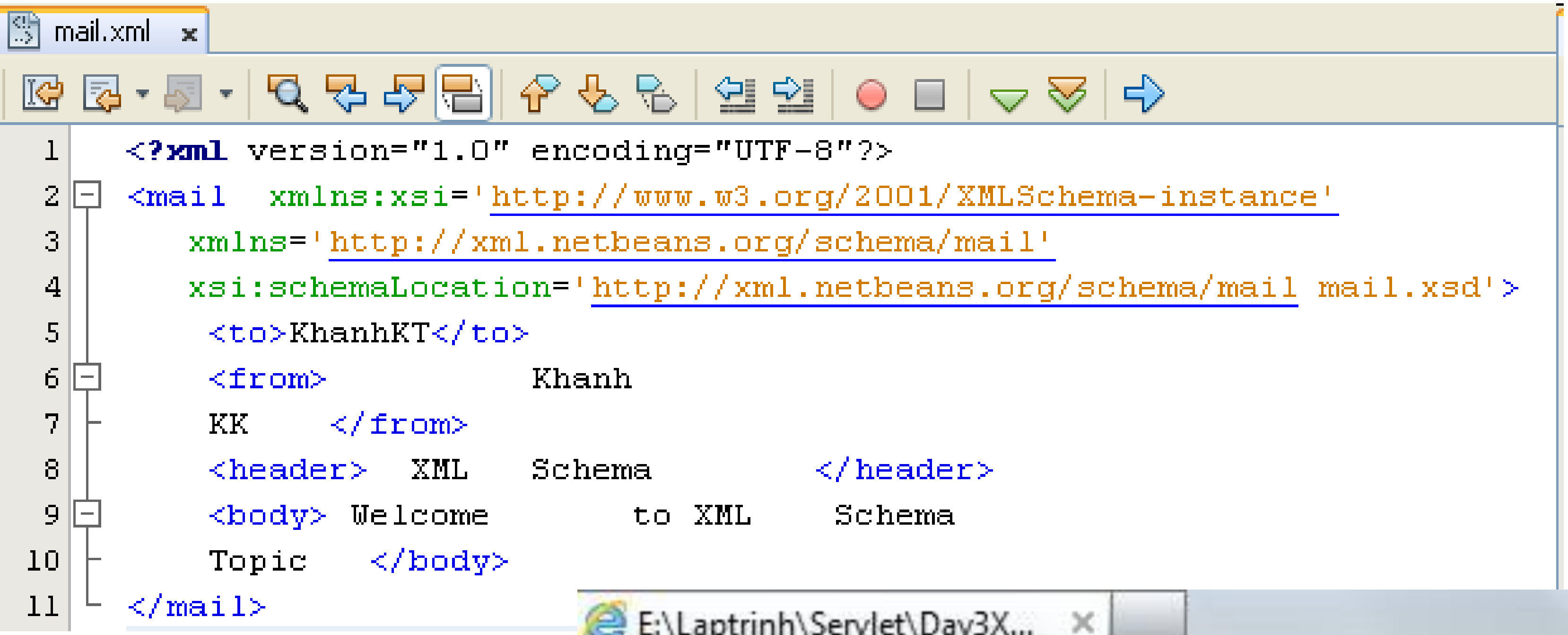
Schemas
Structuring–XSD Data Types–string–Example



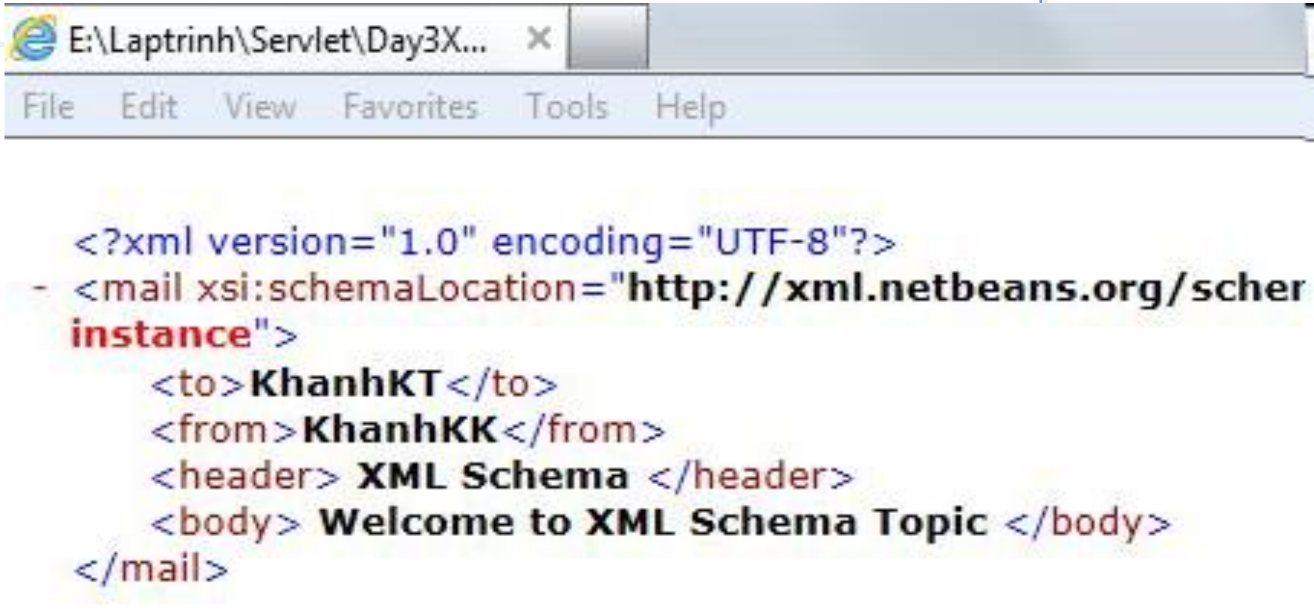
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/mail"
    xmlns="http://xml.netbeans.org/schema/mail"
    elementFormDefault="qualified">
    <xsd:element name="mail">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="to" type="xsd:string" />
                <xsd:element name="from" type="xsd:token"/>
                <xsd:element name="header" type="xsd:string"/>
                <xsd:element name="body" type="xsd:normalizedString"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas

Structuring–XSD Data Types–string–Example



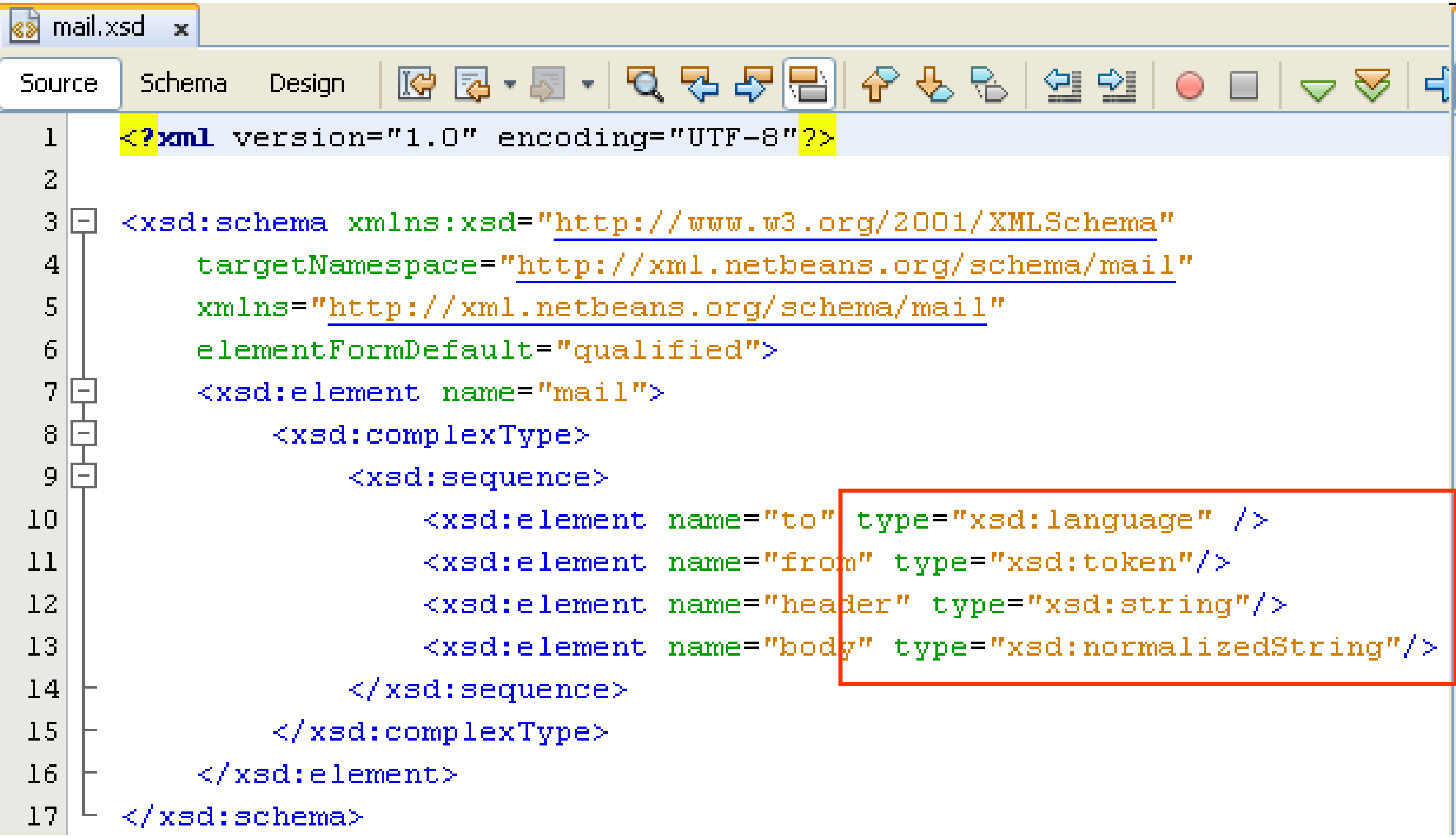
```
<?xml version="1.0" encoding="UTF-8"?>
<mail  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/mail'
   xsi:schemaLocation='http://xml.netbeans.org/schema/mail mail.xsd'>
    <to>KhanhKT</to>
    <from>          Khanh
    KK    </from>
    <header>  XML    Schema        </header>
    <body> Welcome       to XML     Schema
    Topic    </body>
</mail>
```



E:\Laptrinh\Servlet\Day3X...
File Edit View Favorites Tools Help

```
<?xml version="1.0" encoding="UTF-8"?>
- <mail xsi:schemaLocation="http://xml.netbeans.org/scher
instance">
     <to>KhanhKT</to>
     <from>KhanhKK</from>
     <header> XML Schema </header>
     <body> Welcome to XML Schema Topic </body>
</mail>
```

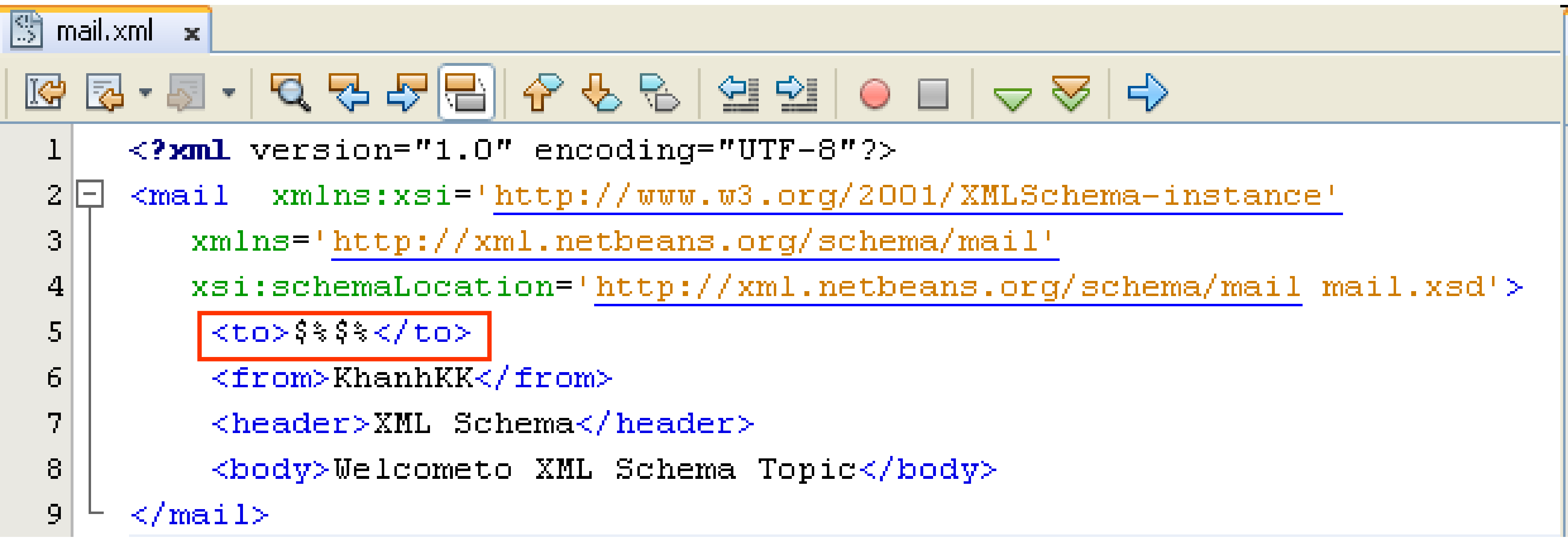
Schemas
Structuring–XSD Data Types–string–Example



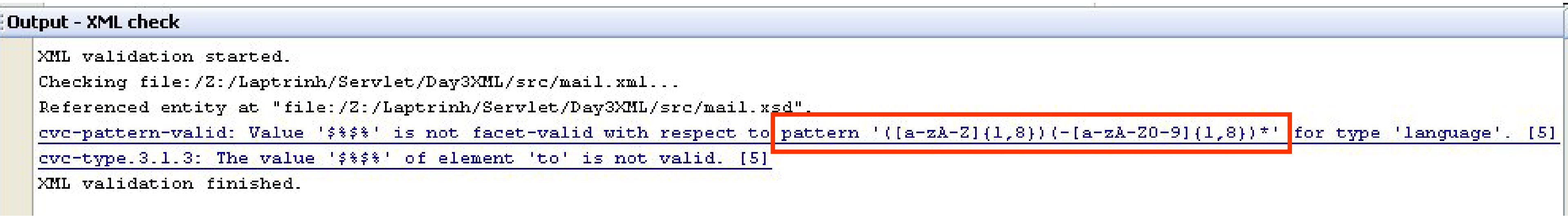
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/mail"
    xmlns="http://xml.netbeans.org/schema/mail"
    elementFormDefault="qualified">
    <xsd:element name="mail">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="to" type="xsd:language" />
                <xsd:element name="from" type="xsd:token"/>
                <xsd:element name="header" type="xsd:string"/>
                <xsd:element name="body" type="xsd:normalizedString"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas

Structuring–XSD Data Types–string–Example



```
<?xml version="1.0" encoding="UTF-8"?>
<mail  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/mail'
    xsi:schemaLocation='http://xml.netbeans.org/schema/mail mail.xsd'>
     <to>$%$%</to>
     <from>KhanhKK</from>
     <header>XML Schema</header>
     <body>Welcometo XML Schema Topic</body>
</mail>
```



Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/mail.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/mail.xsd".
cvc-pattern-valid: Value '$%$%' is not facet-valid with respect to pattern '([a-zA-Z]{1,8})(-[a-zA-Z0-9]{1,8})*' for type 'language'. [5]
cvc-type.3.1.3: The value '$%$%' of element 'to' is not valid. [5]
XML validation finished.
```

Schemas
Structuring–XSD Data Types–Miscellaneous

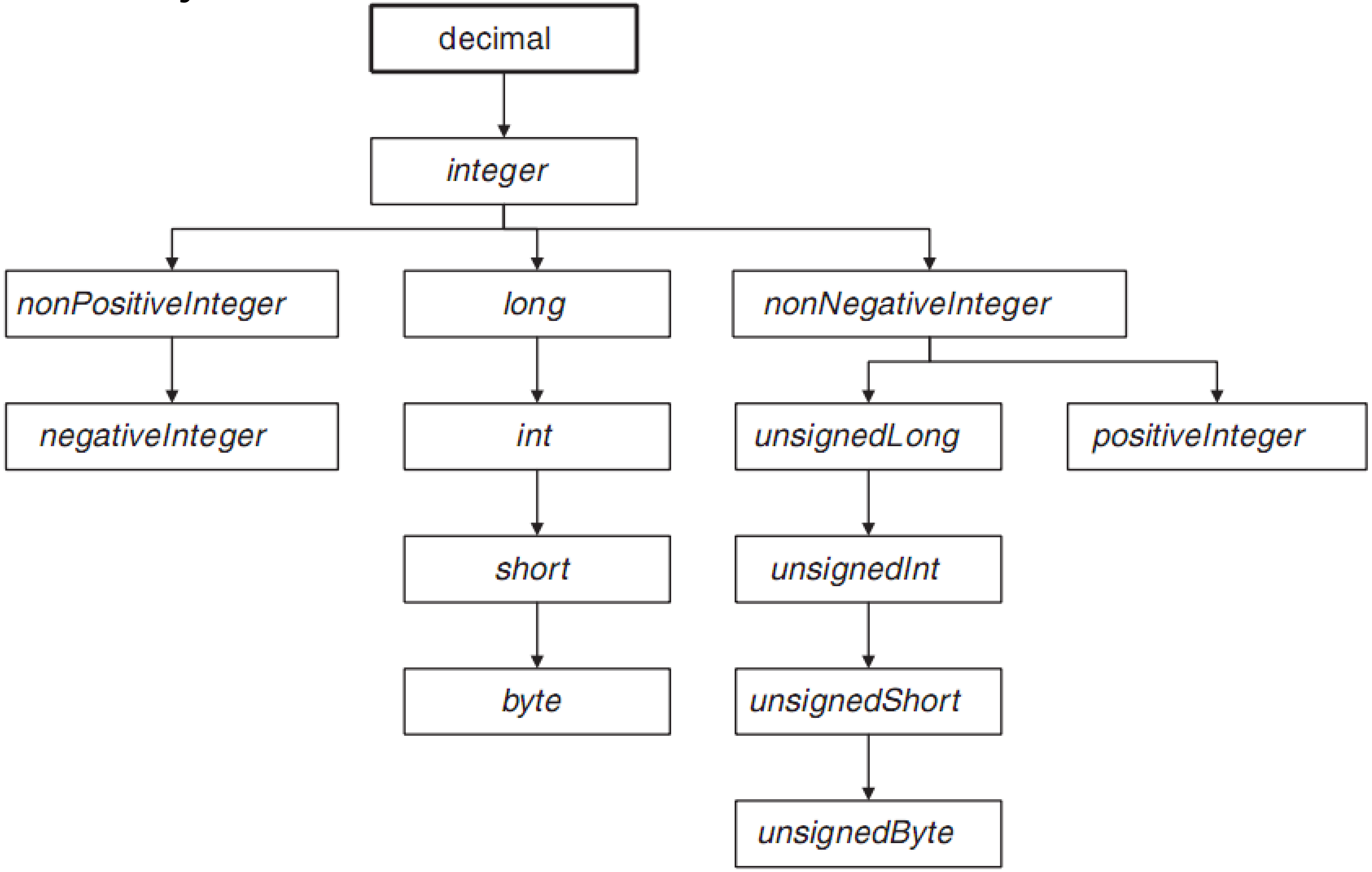
- **Boolean** data types
  - Is used to **specify** a **true** or **false** value
  - **True** can be replaced by the numeric **value of 1** and **false** can be replaced by the **value 0**
  - Is often **used** in **attribute**
  - **Syntax**: **xsd:boolean**
- **Binary** data types
  - Are used to **express binary-formatted data** including **graphic** file, **executable program** or any other strings
  - There are two binary data types
    - **base64Binary** (Base64-encoded binary data)
    - **hexBinary** (hexadecimal-encoded binary data)
  - **Syntax**: **xsd:hexBinary** or **xsd:base64Binary**

Schemas
Structuring–XSD Data Types–Miscellaneous

- **anyURI** data types
  - Is used to **specify** a **URI** that represents the file name or location of the file
  - If a URI has **spaces**, **replace** them with **%20**
  - Is often **used in attribute**
  - **Syntax**: **xsd:anyURI**
- **NOTATION** data types
  - Represents a **NOTATION attribute** type. A set of QNames
  - **Syntax**: **xsd:NOTATION**

Schemas
Structuring–XSD Data Types–Numeric

- Represents a **numeric** value such as whole numbers & real numbers
- Type hierarchy of numeric as



Fpt University

Schemas

Structuring – XSD Data Types – Numeric

Type	Description
decimal	Is used to specify an arbitrary-precision decimal number that presents exact fraction part. **Syntax: xsd:decimal**
integer	Is used to specify a numeric value without a fractional component that includes the positive and negative numbers. **Syntax: xsd:integer**
nonPositiveInteger	An integer containing only non-positive values (..,-2,-1,0). **Syntax: xsd:nonPositiveInteger**
negativeInteger	An integer containing only negative values (..,-2,-1). **Syntax: xsd:negativeInteger**
nonNegativeInteger	An integer containing only non-negative values (0,1,2,..). **Syntax**: **xsd:nonNegativeInteger**
unsignedLong	An unsigned 64-bit integer. **Syntax**: **xsd:unsignedLong**
positiveInteger	An integer containing only positive values (1,2,..). **Syntax**: **xsd:positiveInteger**

Schemas

Structuring – XSD Data Types – Numeric

Type	Description
unsignedInt	An unsigned 32-bit integer. **Syntax: xsd:unsignedInt**
unsignedShort	An unsigned 16-bit integer. **Syntax: xsd:unsignedShort**
unsignedByte	An unsigned 8-bit integer. **Syntax: xsd:unsignedByte**
long	A signed 64-bit integer. **Syntax: xsd:long**
int	A signed 32-bit integer. **Syntax**: **xsd:int**
short	A signed 16-bit integer. **Syntax**: **xsd:short**
byte	A signed 8-bit integer. **Syntax**: **xsd:byte**

Schemas

Structuring – XSD Data Types – Date & Time

Type	Description
dateTime	Represents a specific instance of time. The **pattern** for dateTime is **CCYY-MM-DDThh:mm:ss.sss** with T as a seperator **Syntax: xsd:dateTime**
date	Represents a calendar date. The **pattern** for date is **CCYY-MM-DD.** **Syntax: xsd:date**
time	Represents an instance of time that recurs every day. The pattern for time is **hh:mm:ss.sss.** **Syntax: xsd:time**
duration	Represents a duration of time. The pattern for duration is **PnYnMnDTnHnMnS**. **Syntax: xsd:duration**

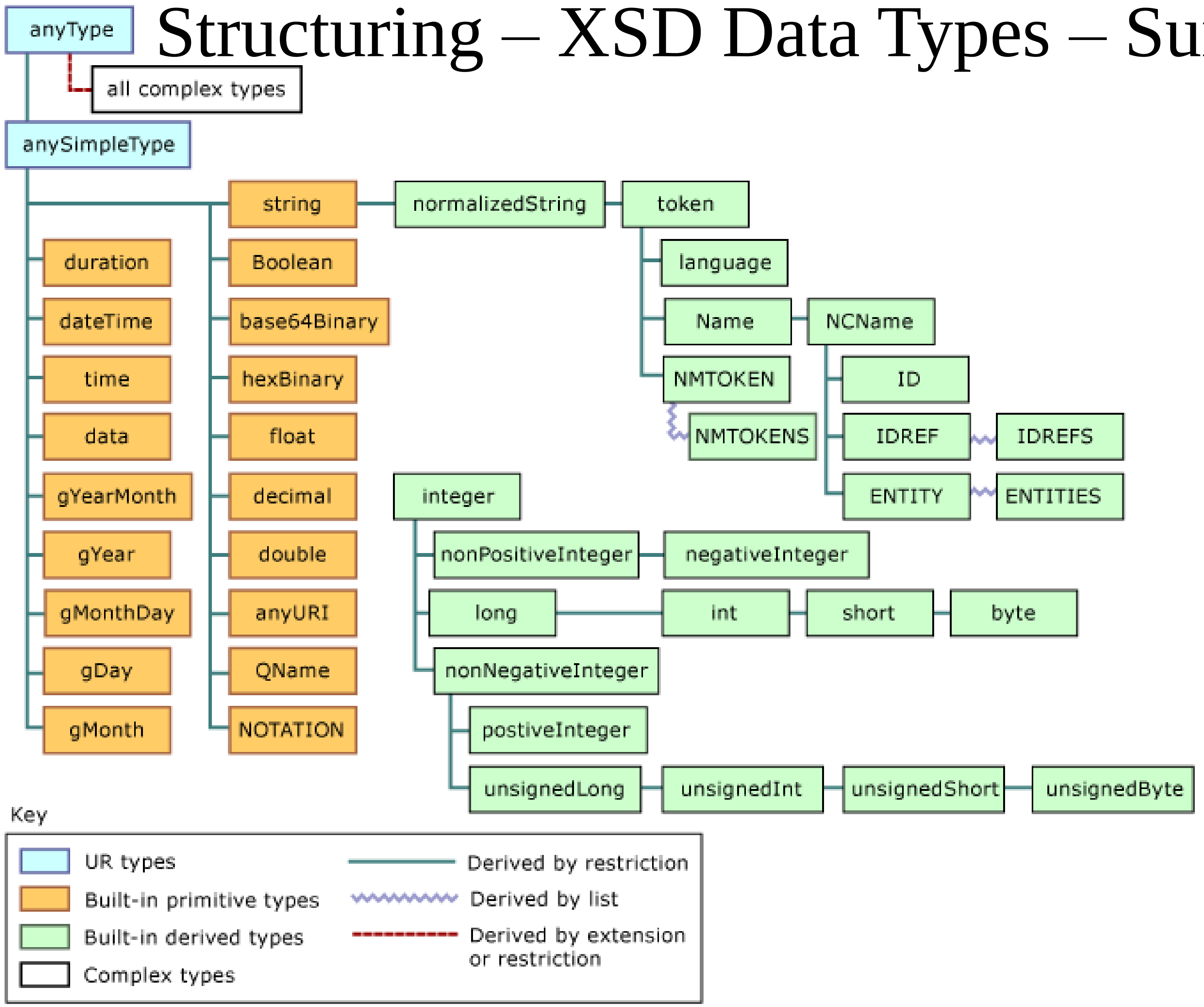
Schemas

Structuring – XSD Data Types – Date & Time

Type	Description
gYearMonth	Represents a specific Gregorian month in a specific Gregorian year. The **pattern** for gYearMonth is **CCYY-MM**. Syntax: **xsd:gYearMonth**
gYear	Represents a specific **Gregorian** year. The **pattern** for gYear is **CCYY**. **Syntax: xsd:gYear**
gMonthDay	Represents a specific Gregorian date. The **pattern** for gMonthDay **is --MM-DD.** **Syntax: xsd:gMonthDay**
gDay	Represents a Gregorian day. The **pattern** for gDay is **---DD**. **Syntax: xsd:gDay**
gMonth	Represents a Gregorian month. The **pattern** for gMonth **is --MM--. Syntax: xsd:gMonth**

Schemas

Structuring – XSD Data Types – Summary



Schemas
Structuring – User Derived Simple Type

- Supports to create custom user-defined data types
- **Syntax**:

  <xsd:simpleType name=“newName”>

  restriction type, or list type, or union type

  </xsd:simpleType>
- Simple types are **derived** by
  - **Restricting built-in** simple types, or, other **user-derived** simple types.
  - **Ex**:

```
<xsd:simpleType name="Password">
     <xsd:restriction base="xsd:string">
          <xsd:length value="8"/>
     </xsd:restriction>
</xsd:simpleType>
```

- Simple types can be derived by applying one or more of the facets

Schemas
Structuring – XSD Data Types – Non-Atomic

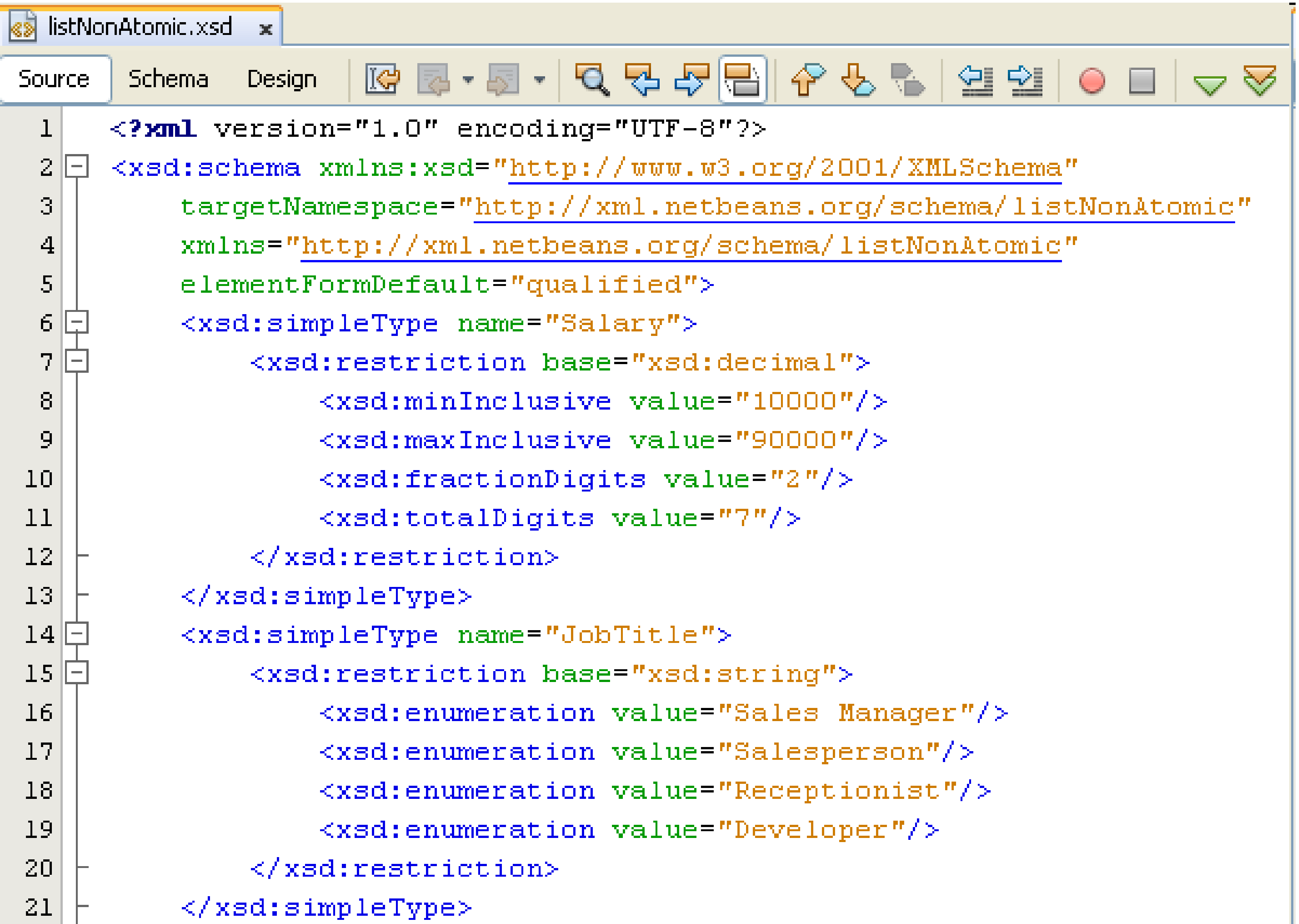
- All of XML Schema's **built-in types are atomic**
  - Meaning that they **cannot** be **broken down** into meaningful bits
- XML Schema provides for **two non-atomic** types
  - **lists**
    - Are **sequences** of **atomic types separated** by **whitespace**
    - Represent a **single value** within an element
    - **Syntax**

```
<xsd:simpleType name=“type_name">
   <xsd:list itemType=“data_type"/>
</xsd:simpleType>
```

  - **unions**
    - Are **groupings of types**, essentially allowing for the value of an element to be of **more than one type**
    - **Syntax**

```
<xsd:simpleType name=“type_name">
   <xsd:union memberTypes=“list of type separating with spaces"/>
</xsd:simpleType>
```

Schemas
Structuring – Non-Atomic – Example



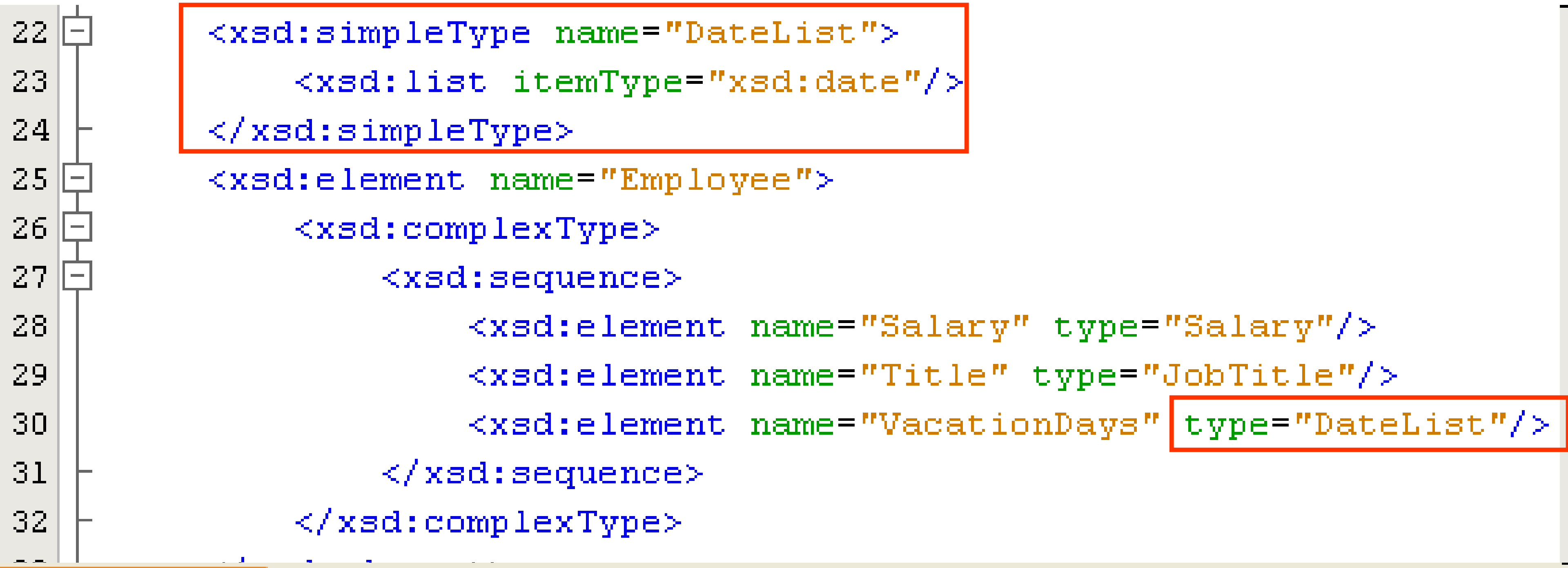
listNonAtomic.xsd

Source Schema Design

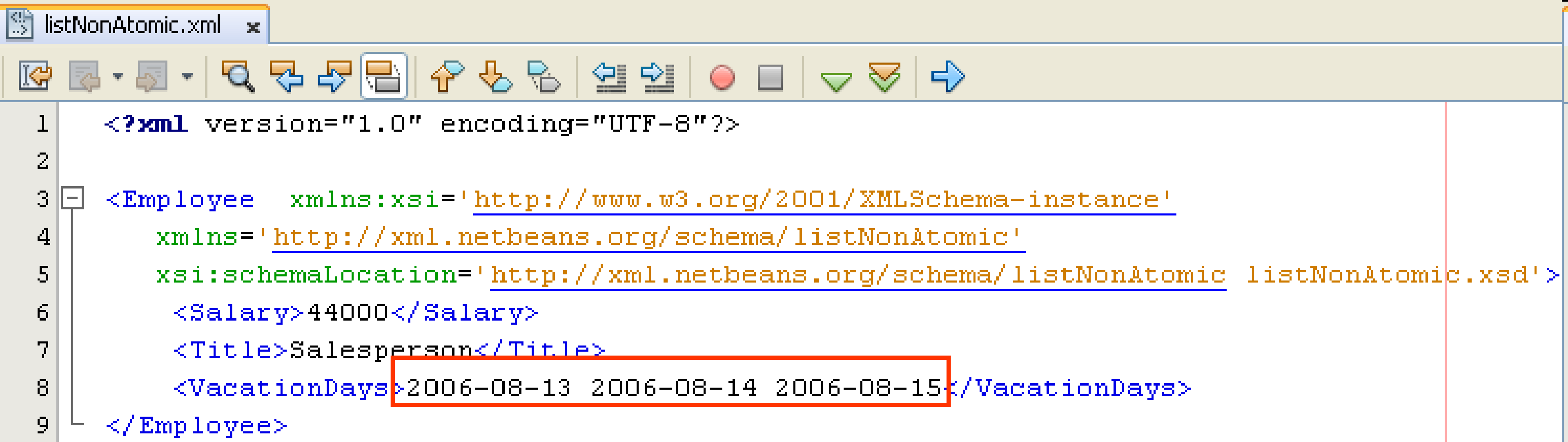
```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/listNonAtomic"
    xmlns="http://xml.netbeans.org/schema/listNonAtomic"
    elementFormDefault="qualified">
    <xsd:simpleType name="Salary">
        <xsd:restriction base="xsd:decimal">
            <xsd:minInclusive value="10000"/>
            <xsd:maxInclusive value="90000"/>
            <xsd:fractionDigits value="2"/>
            <xsd:totalDigits value="7"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="JobTitle">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Sales Manager"/>
            <xsd:enumeration value="Salesperson"/>
            <xsd:enumeration value="Receptionist"/>
            <xsd:enumeration value="Developer"/>
        </xsd:restriction>
    </xsd:simpleType>
```

Schemas

Structuring – Non-Atomic – Example



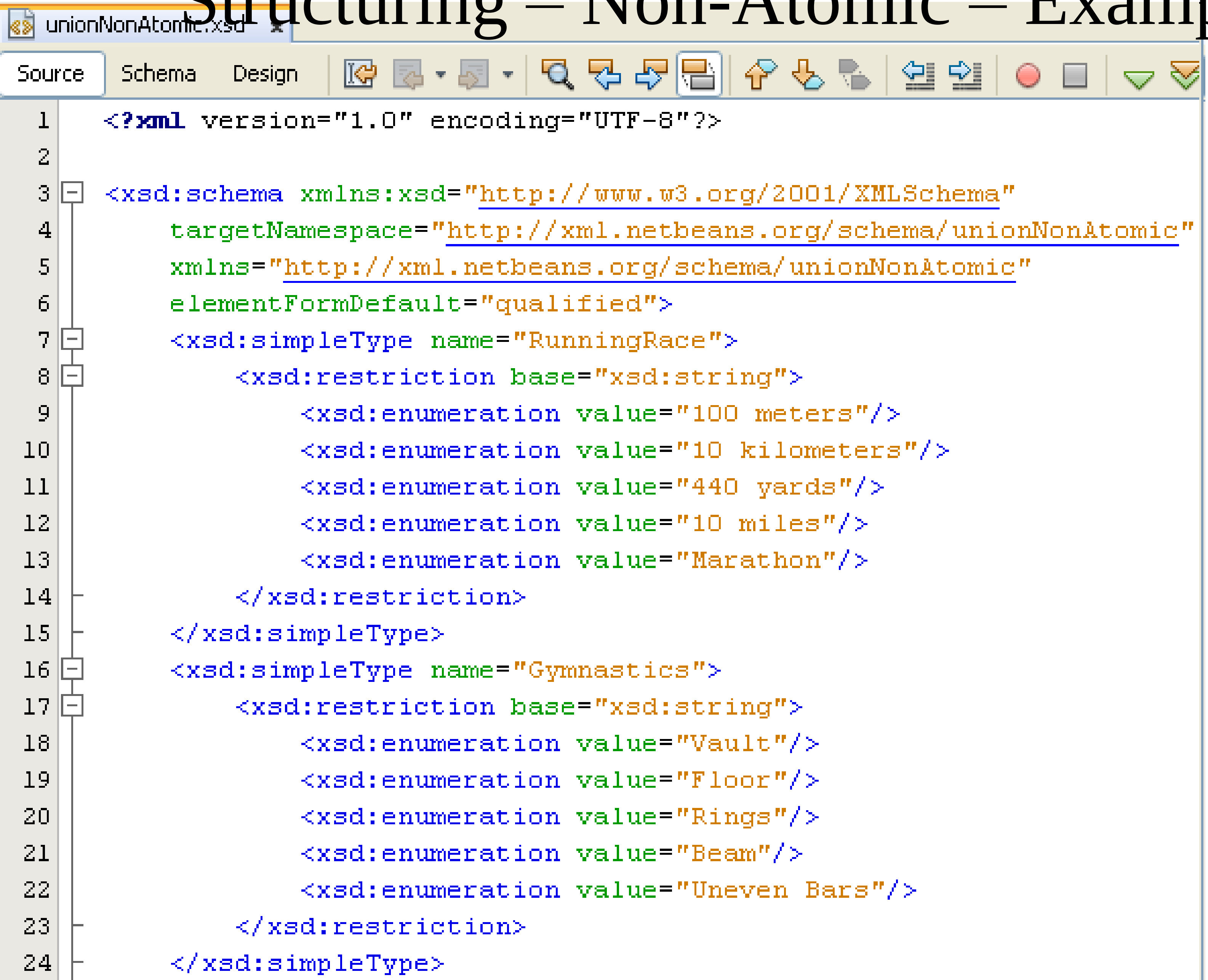
```
<xsd:simpleType name="DateList">
    <xsd:list itemType="xsd:date"/>
</xsd:simpleType>
<xsd:element name="Employee">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="Salary" type="Salary"/>
            <xsd:element name="Title" type="JobTitle"/>
            <xsd:element name="VacationDays" type="DateList"/>
        </xsd:sequence>
    </xsd:complexType>
```



listNonAtomic.xml

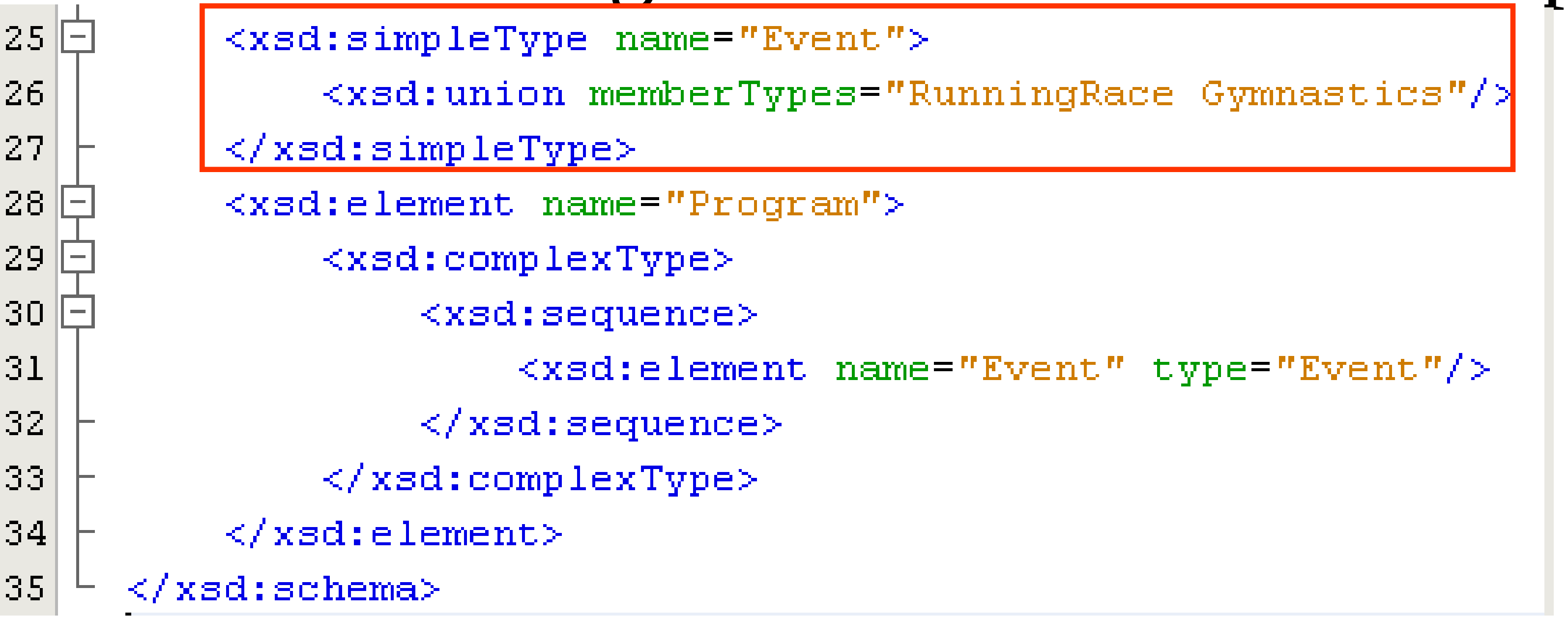
```
<?xml version="1.0" encoding="UTF-8"?>

<Employee  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/listNonAtomic'
   xsi:schemaLocation='http://xml.netbeans.org/schema/listNonAtomic listNonAtomic.xsd'>
    <Salary>44000</Salary>
    <Title>Salesperson</Title>
    <VacationDays>2006-08-13 2006-08-14 2006-08-15</VacationDays>
</Employee>
```

Schemas
Structuring – Non-Atomic – Example

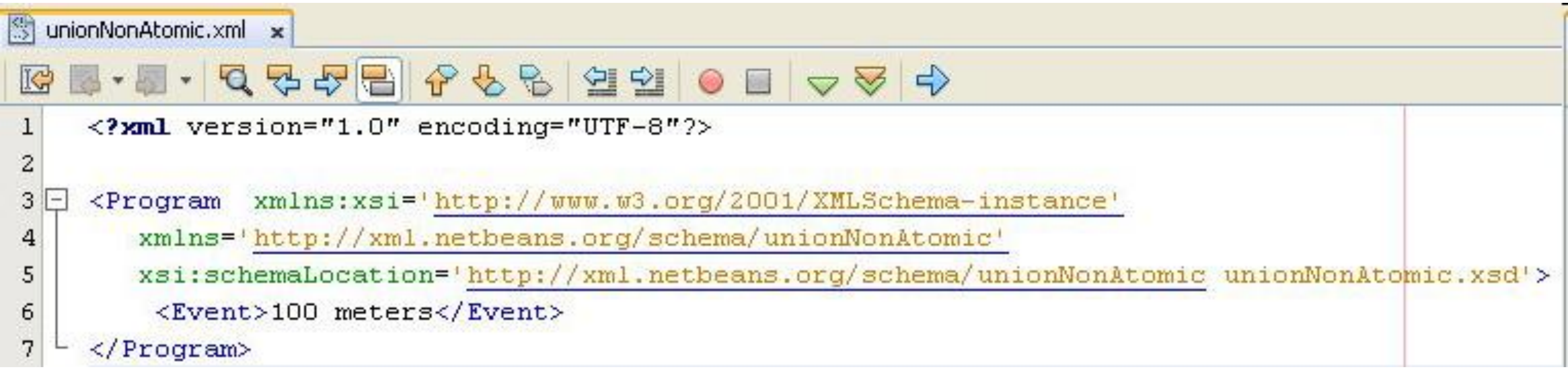


```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/unionNonAtomic"
    xmlns="http://xml.netbeans.org/schema/unionNonAtomic"
    elementFormDefault="qualified">
    <xsd:simpleType name="RunningRace">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="100 meters"/>
            <xsd:enumeration value="10 kilometers"/>
            <xsd:enumeration value="440 yards"/>
            <xsd:enumeration value="10 miles"/>
            <xsd:enumeration value="Marathon"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="Gymnastics">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Vault"/>
            <xsd:enumeration value="Floor"/>
            <xsd:enumeration value="Rings"/>
            <xsd:enumeration value="Beam"/>
            <xsd:enumeration value="Uneven Bars"/>
        </xsd:restriction>
    </xsd:simpleType>
```

Schemas
Structuring – Non-Atomic – Example



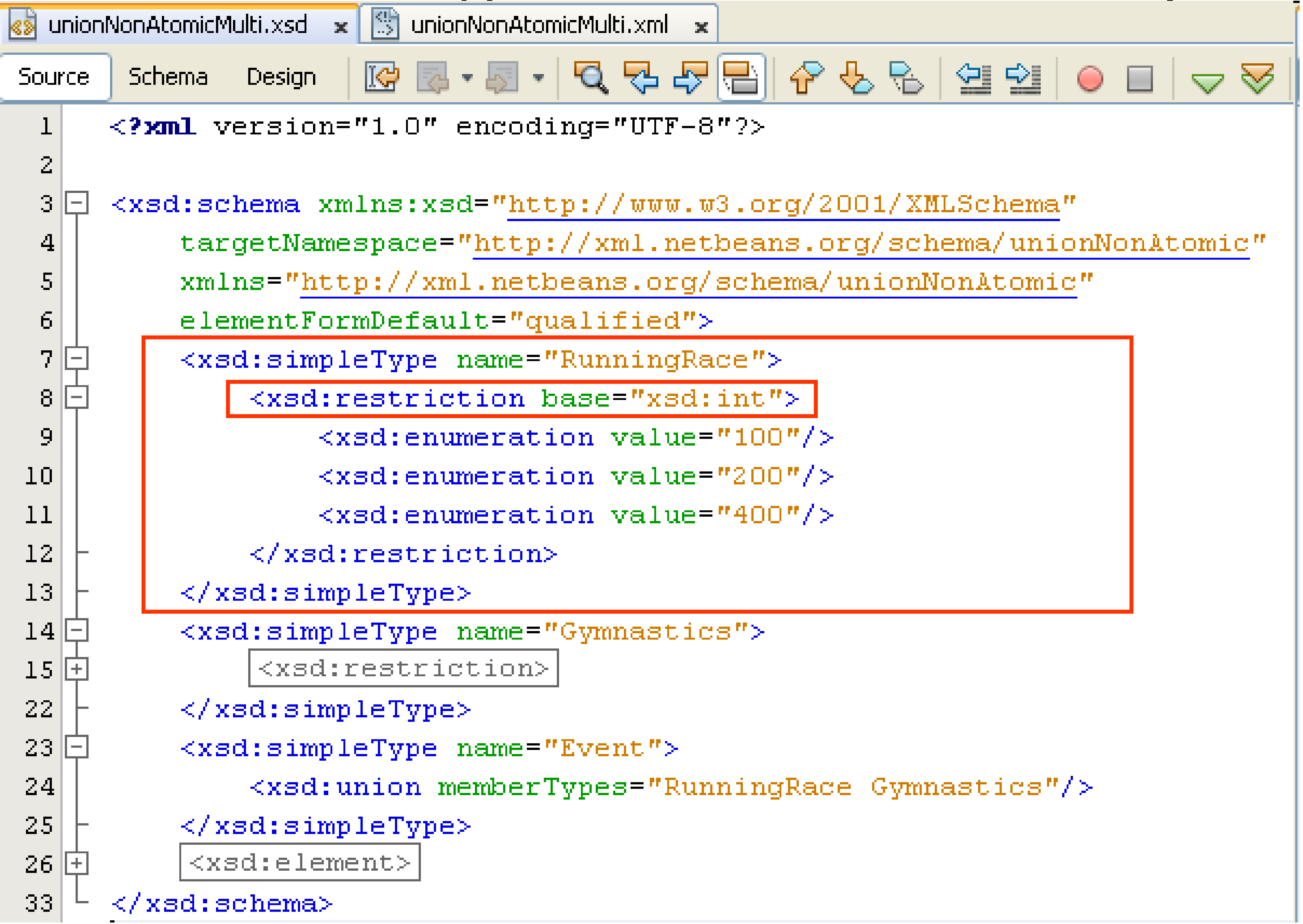
```
    <xsd:simpleType name="Event">
        <xsd:union memberTypes="RunningRace Gymnastics"/>
    </xsd:simpleType>
    <xsd:element name="Program">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Event" type="Event"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```



unionNonAtomic.xml

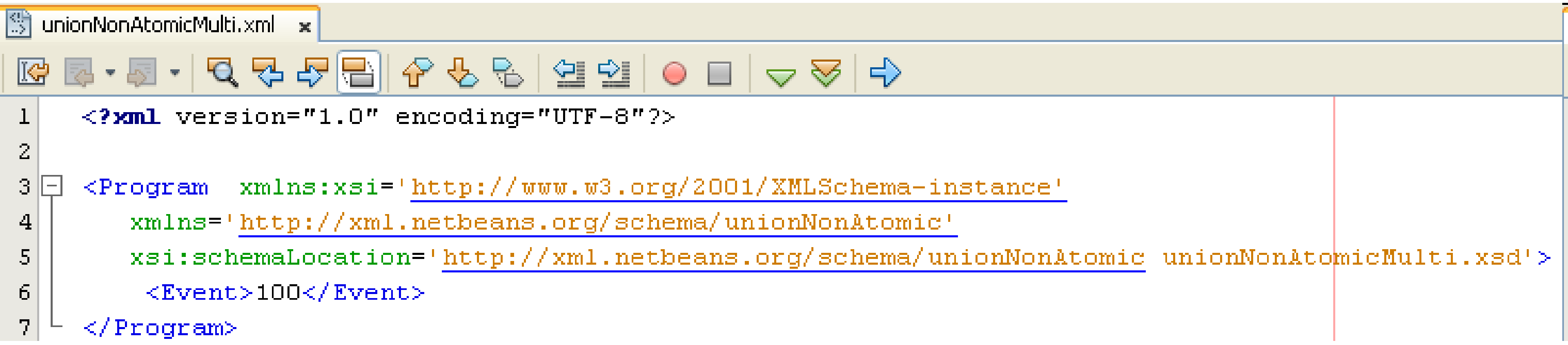
```
<?xml version="1.0" encoding="UTF-8"?>

<Program  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/unionNonAtomic'
   xsi:schemaLocation='http://xml.netbeans.org/schema/unionNonAtomic unionNonAtomic.xsd'>
    <Event>100 meters</Event>
</Program>
```

Schemas
Structuring – Non-Atomic – Example

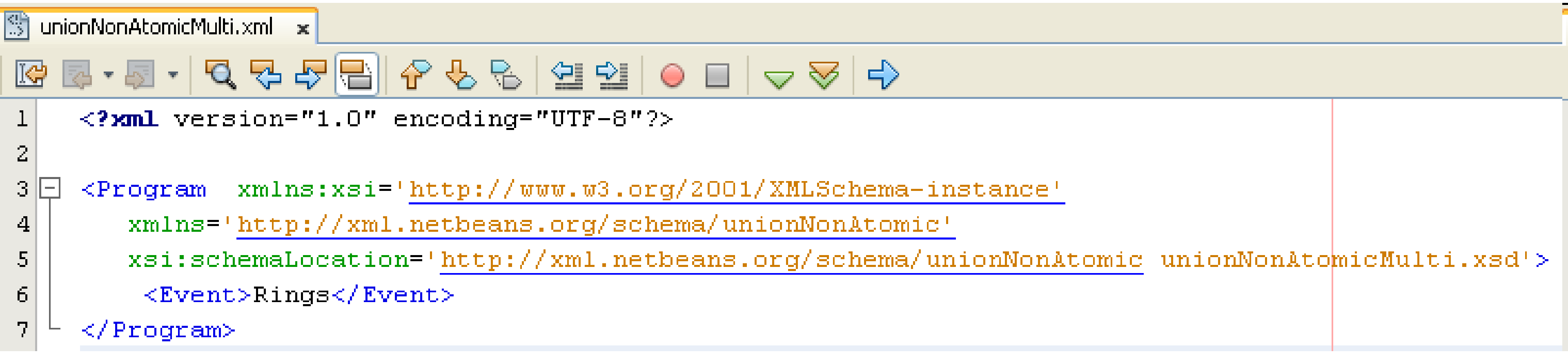


```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/unionNonAtomic"
    xmlns="http://xml.netbeans.org/schema/unionNonAtomic"
    elementFormDefault="qualified">
    <xsd:simpleType name="RunningRace">
        <xsd:restriction base="xsd:int">
            <xsd:enumeration value="100"/>
            <xsd:enumeration value="200"/>
            <xsd:enumeration value="400"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="Gymnastics">
        <xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="Event">
        <xsd:union memberTypes="RunningRace Gymnastics"/>
    </xsd:simpleType>
    <xsd:element>
</xsd:schema>
```

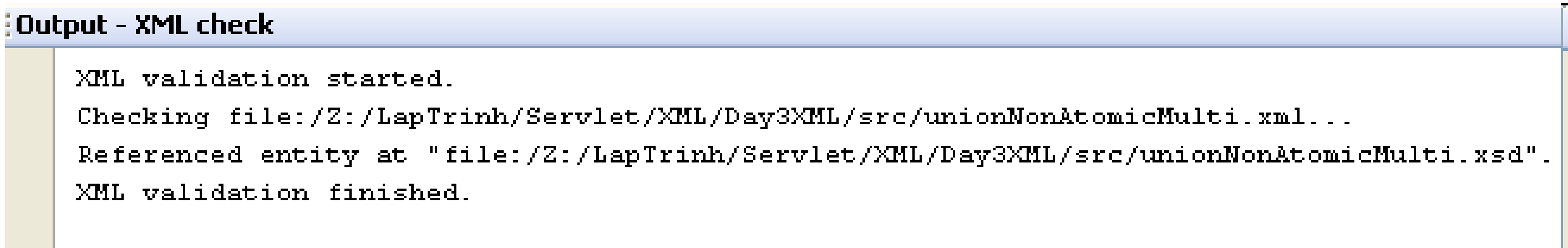
Schemas
Structuring – Non-Atomic – Example



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

<Program  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/unionNonAtomic'
    xsi:schemaLocation='http://xml.netbeans.org/schema/unionNonAtomic unionNonAtomicMulti.xsd'>
     <Event>100</Event>
</Program>
```



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

<Program  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/unionNonAtomic'
    xsi:schemaLocation='http://xml.netbeans.org/schema/unionNonAtomic unionNonAtomicMulti.xsd'>
     <Event>Rings</Event>
</Program>
```



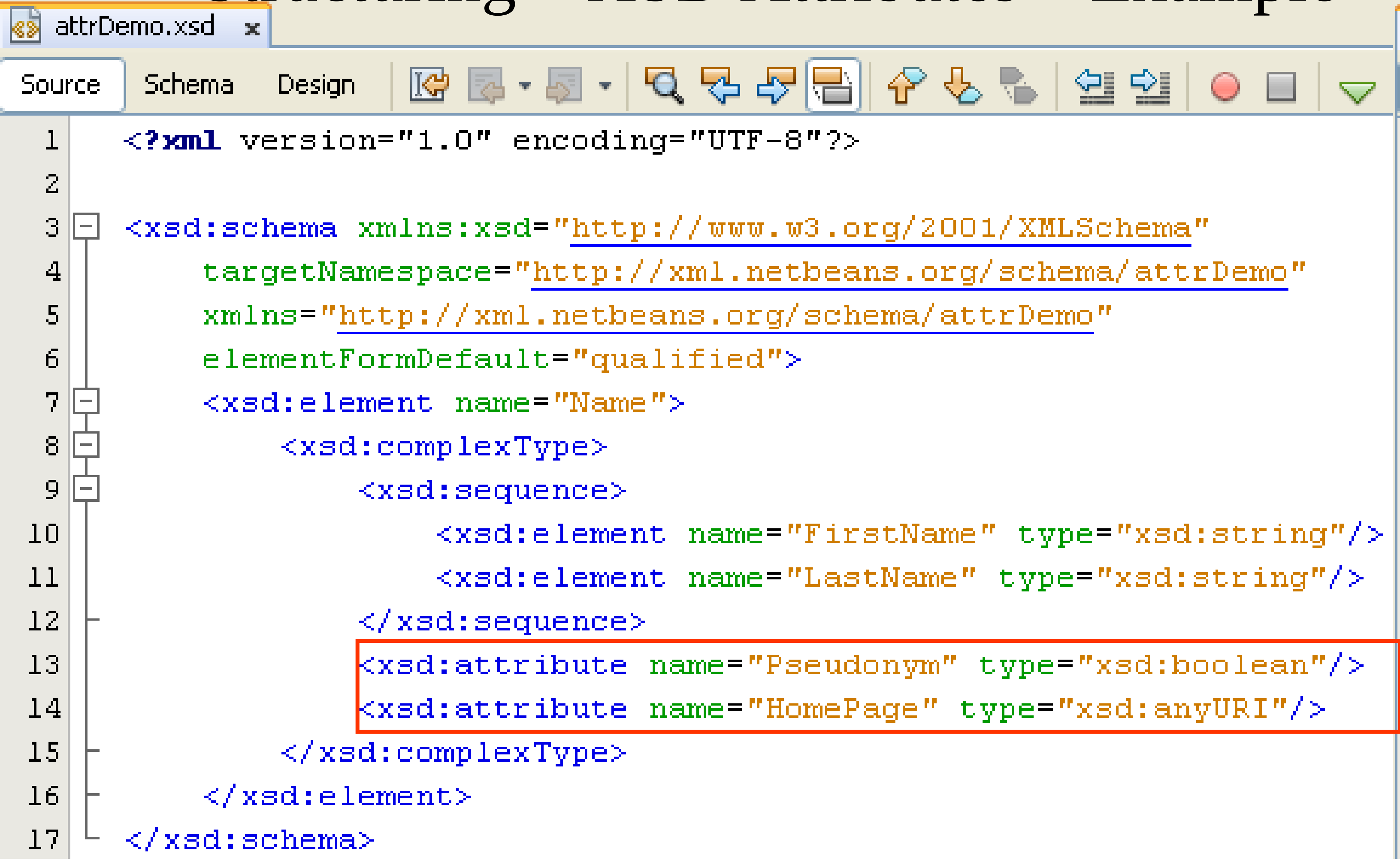
```
XML validation started.
Checking file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/unionNonAtomicMulti.xml...
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/unionNonAtomicMulti.xsd".
XML validation finished.
```

Schemas
Structuring – XSD Attributes

- The **attributes** themselves **must** be of **simple type**
  - **Simple elements cannot** have **attributes**.
  - If an **element has attributes**, it is considered to be of a **complex type**
- **Syntax**
  - <xsd:attribute name="attr_name" type="data_types"/>
- **Default** and **Fixed** values are **same as elements**
  - **Default** syntax
    - <xsd:attribute name="attr_name" type="data_types" default="value"/>
  - **Fixed** syntax
    - <xsd:attribute name="attr_name" type="data_types" fixed="value"/>
- **Optional** and **Required** Attributes
  - Attributes are **optional by default**. To specify that the attribute is required, use the "**use**" **attribute** to **set required** value
  - **Syntax** (default and use are not used in the same time)
    - <xsd:attribute name="attr_name" type="data_types" use="required" [fixed="value"]/>

Schemas

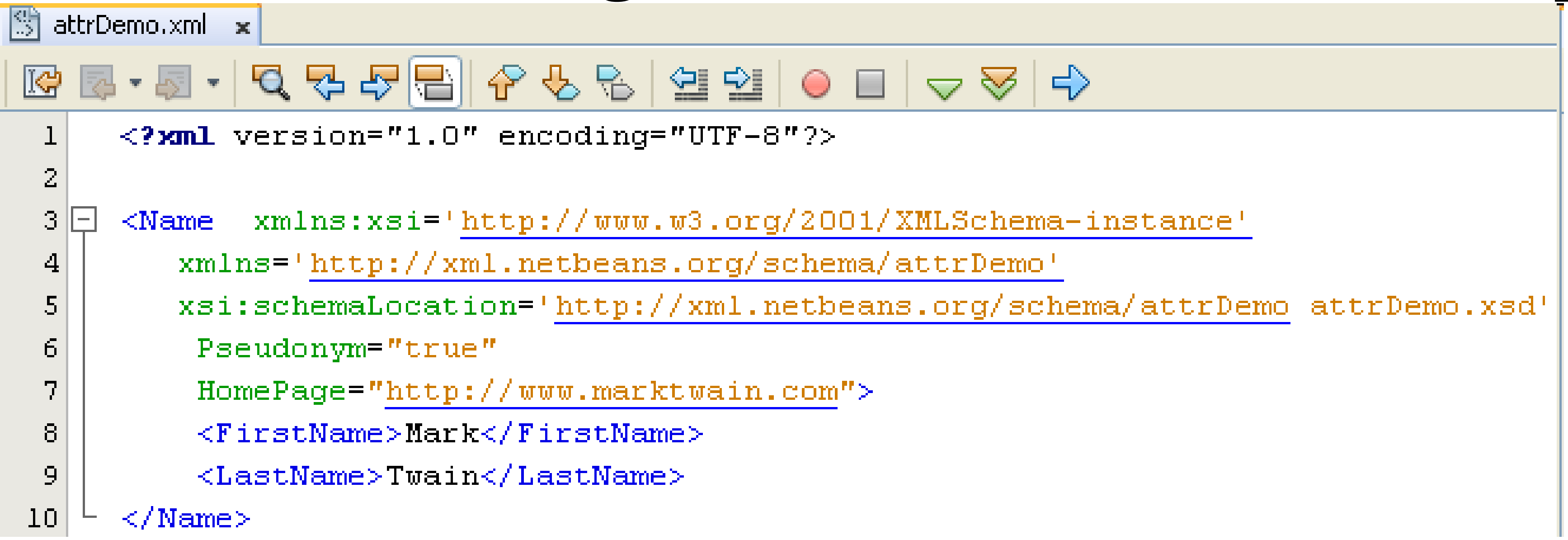
Structuring – XSD Attributes – Example



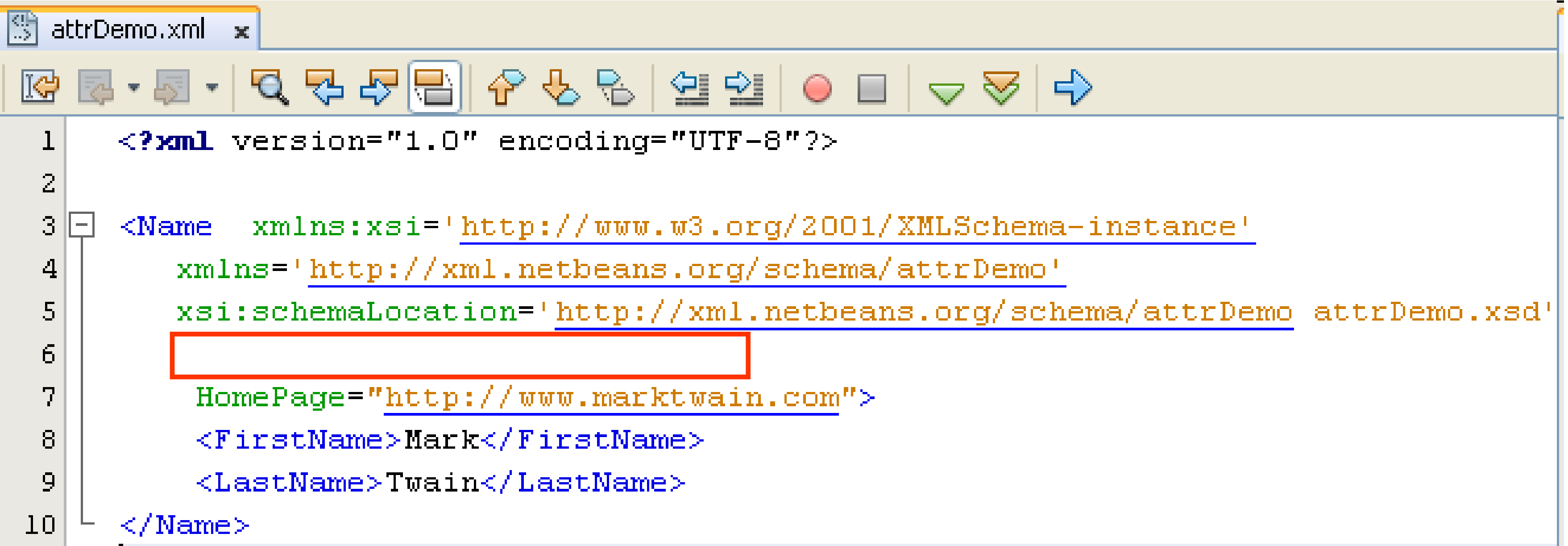
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/attrDemo"
    xmlns="http://xml.netbeans.org/schema/attrDemo"
    elementFormDefault="qualified">
    <xsd:element name="Name">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="FirstName" type="xsd:string"/>
                <xsd:element name="LastName" type="xsd:string"/>
            </xsd:sequence>
            <xsd:attribute name="Pseudonym" type="xsd:boolean"/>
            <xsd:attribute name="HomePage" type="xsd:anyURI"/>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas

Structuring – XSD Attributes – Example



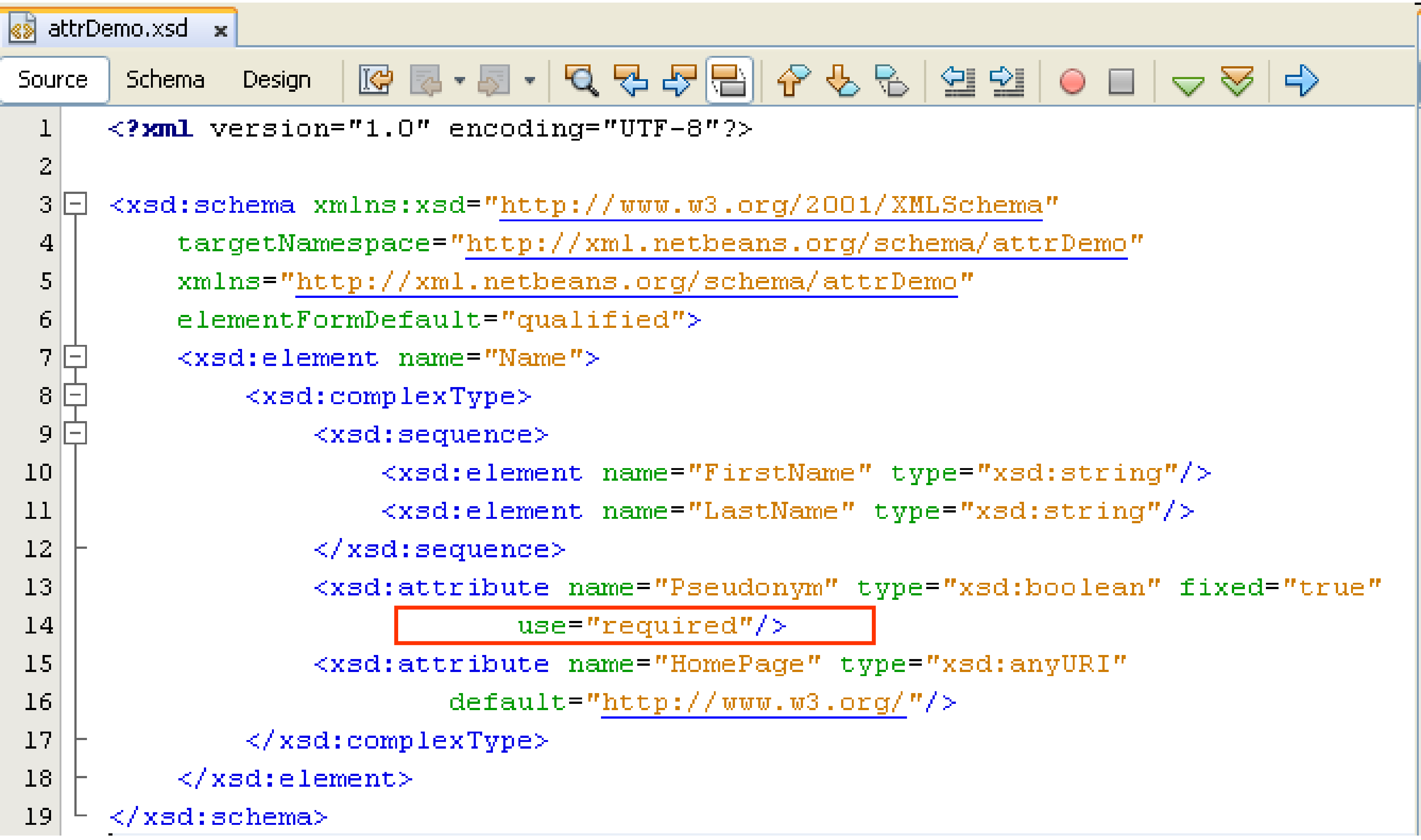
```
<?xml version="1.0" encoding="UTF-8"?>

<Name  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/attrDemo'
    xsi:schemaLocation='http://xml.netbeans.org/schema/attrDemo attrDemo.xsd'
     Pseudonym="true"
     HomePage="http://www.marktwain.com">
     <FirstName>Mark</FirstName>
     <LastName>Twain</LastName>
</Name>
```



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

<Name  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/attrDemo'
    xsi:schemaLocation='http://xml.netbeans.org/schema/attrDemo attrDemo.xsd'

     HomePage="http://www.marktwain.com">
     <FirstName>Mark</FirstName>
     <LastName>Twain</LastName>
</Name>
```

Schemas

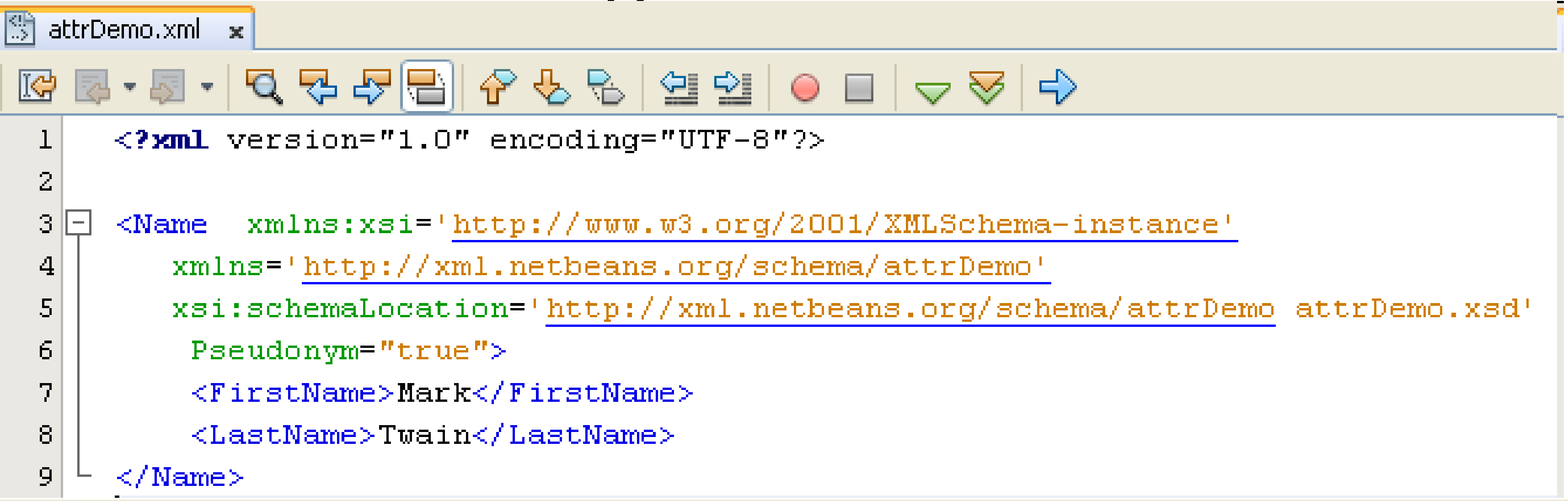
Structuring – XSD Attributes – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/attrDemo"
    xmlns="http://xml.netbeans.org/schema/attrDemo"
    elementFormDefault="qualified">
    <xsd:element name="Name">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="FirstName" type="xsd:string"/>
                <xsd:element name="LastName" type="xsd:string"/>
            </xsd:sequence>
            <xsd:attribute name="Pseudonym" type="xsd:boolean" fixed="true"
                        use="required"/>
            <xsd:attribute name="HomePage" type="xsd:anyURI"
                    default="http://www.w3.org/"/>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

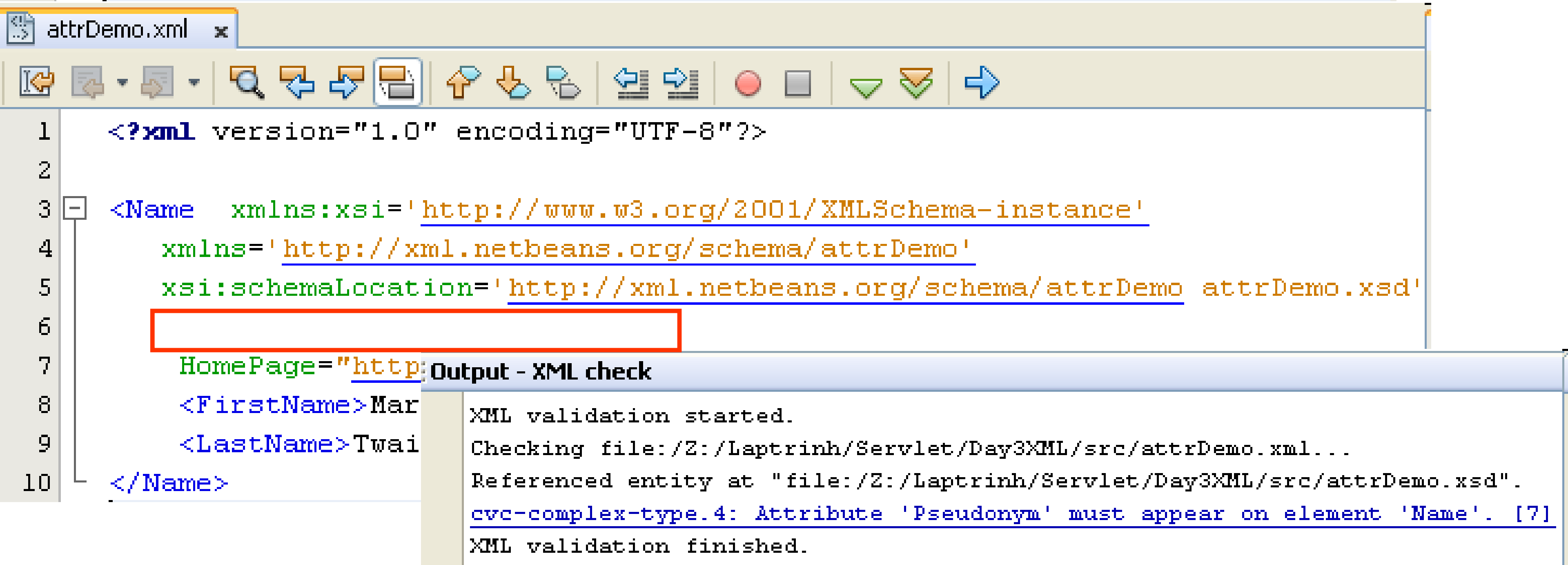
Schemas

Structuring – XSD Attributes – Example



attrDemo.xml

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

<Name  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/attrDemo'
    xsi:schemaLocation='http://xml.netbeans.org/schema/attrDemo attrDemo.xsd'
     Pseudonym="true">
     <FirstName>Mark</FirstName>
     <LastName>Twain</LastName>
</Name>
```



attrDemo.xml

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

<Name  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/attrDemo'
    xsi:schemaLocation='http://xml.netbeans.org/schema/attrDemo attrDemo.xsd'

     HomePage="http
     <FirstName>Mar
     <LastName>Twai
</Name>
```

Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/attrDemo.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/attrDemo.xsd".
cvc-complex-type.4: Attribute 'Pseudonym' must appear on element 'Name'. [7]
XML validation finished.
```

Schemas
Structuring – XSD Facets

- Facets are called **restrictions**
- Are used to **define acceptable values** for XML elements or attributes
  - Are used to **restrict** the **set of values** a data type contain
  - Are **ensured** that the **new value range** must be **equal** to or **narrower than range** of the **base** type
- Syntax

```
<xsd:restriction base="base_type">
    <xsd:facet_element value="value"/>
    ...
</xsd:restriction>
```

- There are **12 facet** elements

Schemas

Structuring – XSD Facets

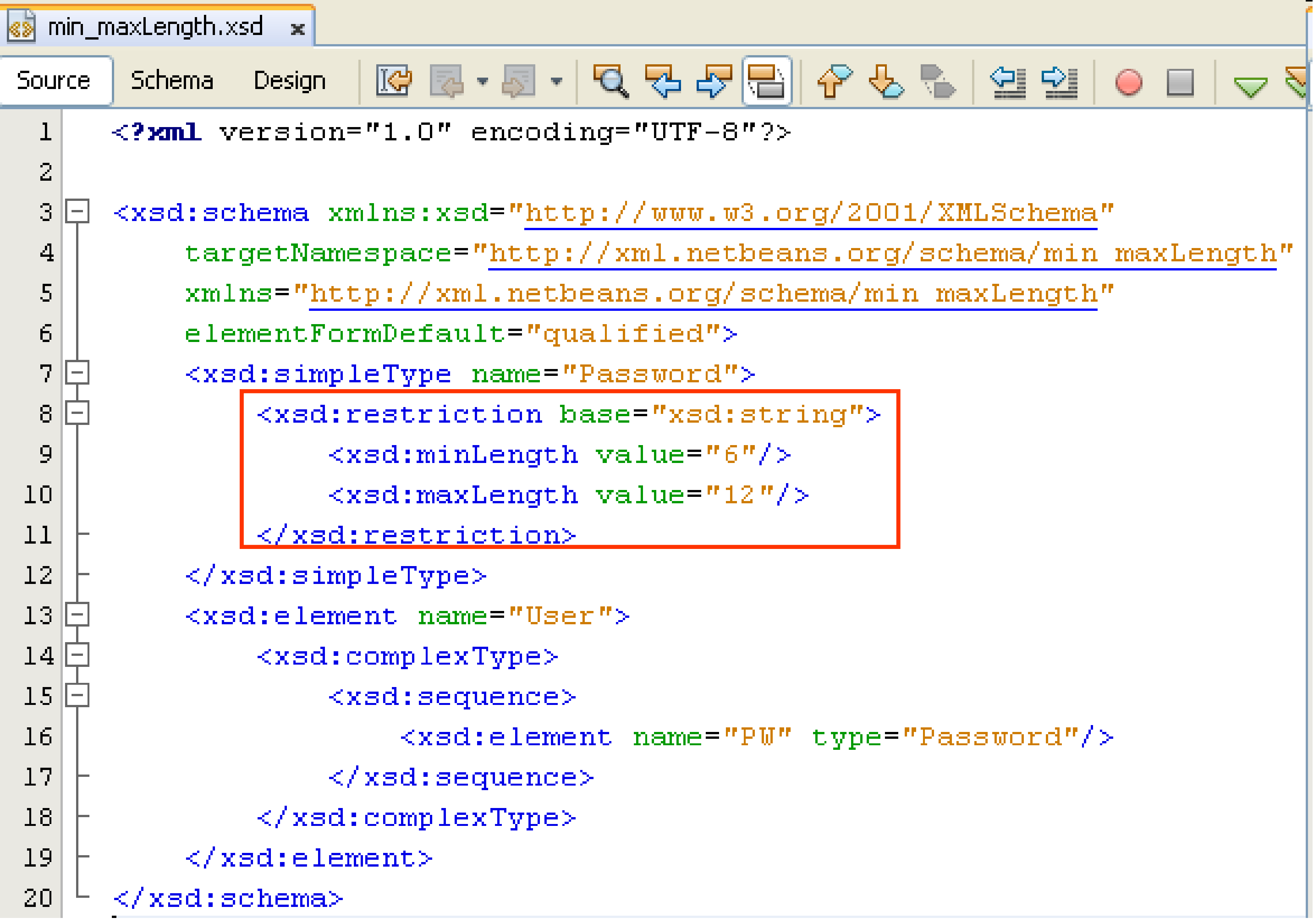
Constraint	Description
enumeration	Defines a **list** of acceptable values. **Syntax: xsd:enumeration**
fractionDigits	Specifies the **maximum** number of **decimal places** allowed. Must be equal to or greater than zero. **Syntax: xsd:fractionDigits**
totalDigits	Specifies the **exact** number of digits allowed. Must be greater than zero. **Syntax: xsd:totalDigits**
length	Specifies the **exact number** of **characters** or **list** items allowed. Must be equal to or greater than zero. **Syntax: xsd:length**
maxExclusive	Specifies the **upper bounds** for numeric values (the value must be less than this value). **Syntax: xsd:maxExclusive**
minExclusive	Specifies the **lower bounds** for numeric values (the value must be greater than this value). **Syntax: xsd:minExclusive**
maxInclusive	Specifies the **upper bounds** for numeric values (the value must be less than or **equal** to this value). **Syntax: xsd:maxInclusive**
minInclusive	Specifies the **lower bounds** for numeric values (the value must be greater than or **equal** to this value). **Syntax: xsd:minInclusive**

Schemas
Structuring – XSD Facets

Constraint	Description
minLength	Specifies the minimum number of characters or list items allowed. Must be equal to or greater than zero. **Syntax: xsd:minLength**
maxLength	Specifies the maximum number of characters or list items allowed. Must be equal to or greater than zero. **Syntax: xsd:maxLength**
pattern	Specific pattern that the data type's values must match. This constrains the data type to literals that match the specified pattern. The pattern value must be a regular expression. **Syntax: xsd:pattern**
whitespace	Value must be one of **preserve** (No normalization is performed), **replace** (replace tab, line feed, carriage return with space), or **collapse** (After the processing implied by replace, multiple space is replace with space and the leading or trailing space is removed). The whiteSpace facet **cannot** be changed for most **numeric data types**. **Syntax: xsd:whitespace**

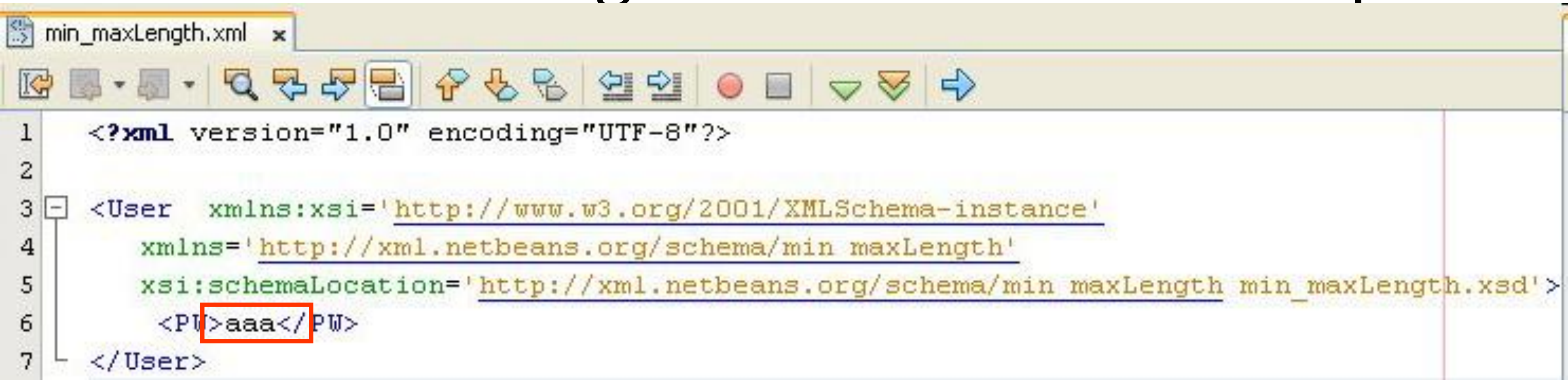
Schemas

Structuring – XSD Facets – Example

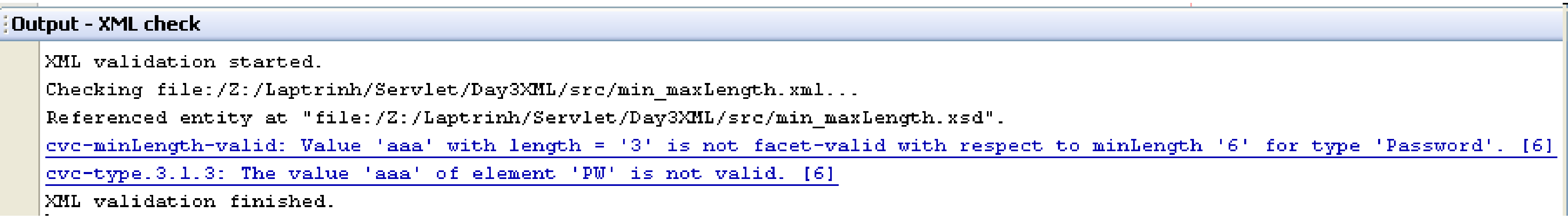


```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://xml.netbeans.org/schema/min_maxLength"
      xmlns="http://xml.netbeans.org/schema/min_maxLength"
      elementFormDefault="qualified">
      <xsd:simpleType name="Password">
          <xsd:restriction base="xsd:string">
              <xsd:minLength value="6"/>
              <xsd:maxLength value="12"/>
          </xsd:restriction>
      </xsd:simpleType>
      <xsd:element name="User">
          <xsd:complexType>
              <xsd:sequence>
                  <xsd:element name="PW" type="Password"/>
              </xsd:sequence>
          </xsd:complexType>
      </xsd:element>
</xsd:schema>
```

Schemas
Structuring – XSD Facets – Example



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

<User  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/min_maxLength'
    xsi:schemaLocation='http://xml.netbeans.org/schema/min_maxLength min_maxLength.xsd'>
     <PW>aaa</PW>
</User>
```

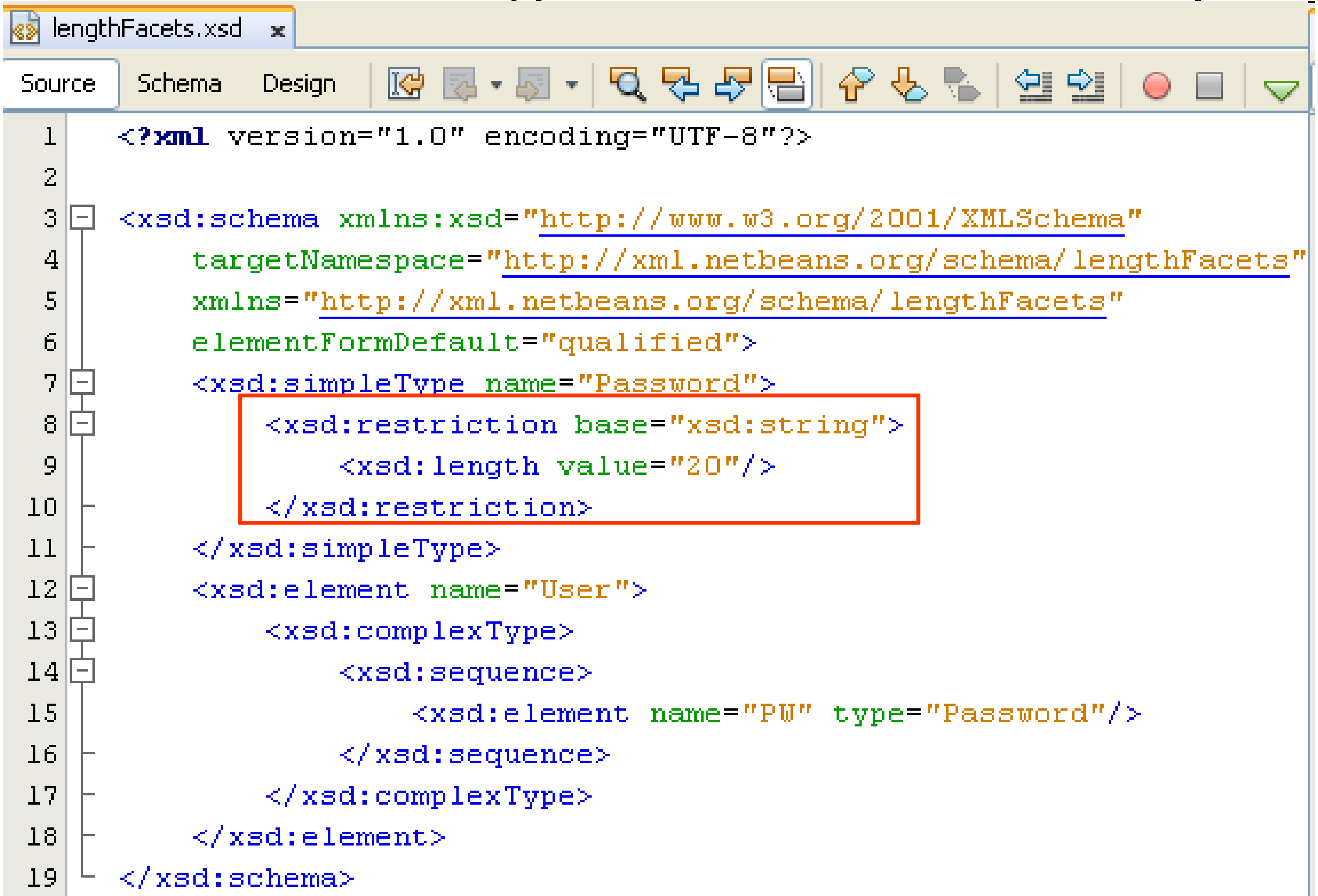


Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/min_maxLength.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/min_maxLength.xsd".
cvc-minLength-valid: Value 'aaa' with length = '3' is not facet-valid with respect to minLength '6' for type 'Password'. [6]
cvc-type.3.1.3: The value 'aaa' of element 'PW' is not valid. [6]
XML validation finished.
```

Schemas

Structuring – XSD Facets – Example



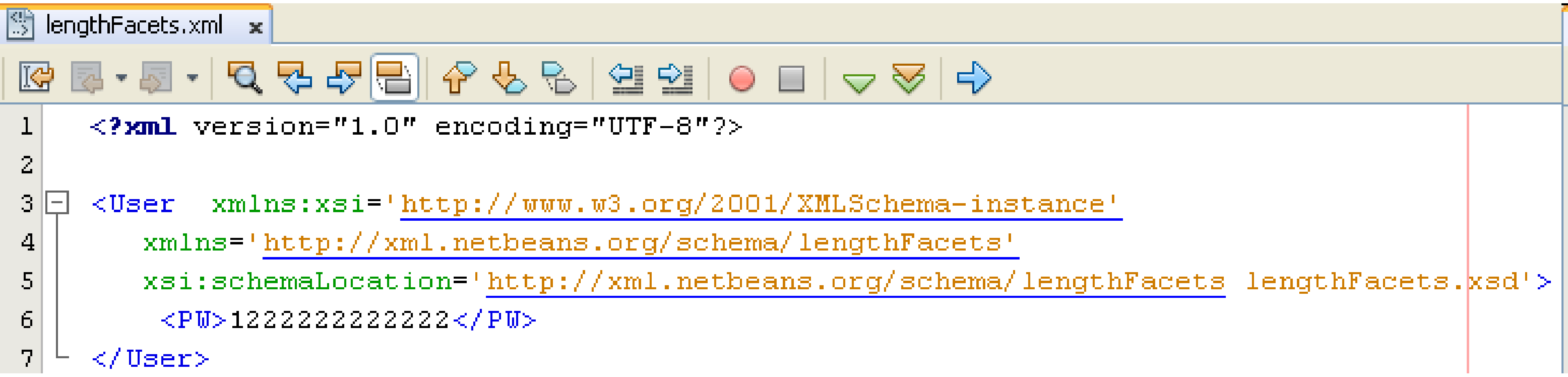
lengthFacets.xsd

Source Schema Design

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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/lengthFacets"
    xmlns="http://xml.netbeans.org/schema/lengthFacets"
    elementFormDefault="qualified">
    <xsd:simpleType name="Password">
        <xsd:restriction base="xsd:string">
            <xsd:length value="20"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="User">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="PW" type="Password"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

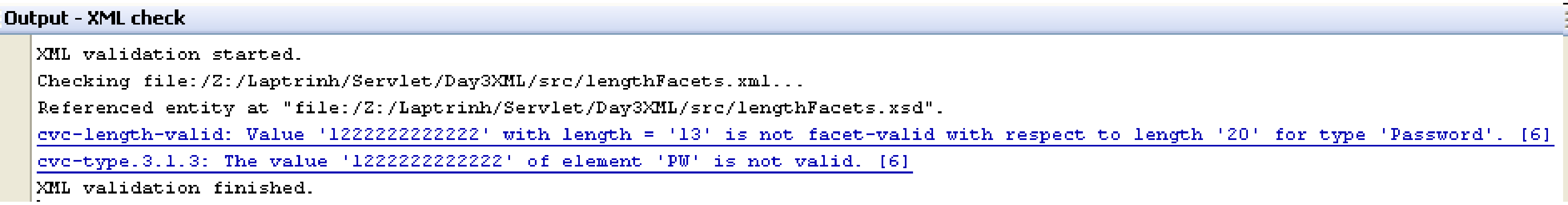
Schemas

Structuring – XSD Facets – Example



lengthFacets.xml

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

<User  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/lengthFacets'
    xsi:schemaLocation='http://xml.netbeans.org/schema/lengthFacets lengthFacets.xsd'>
     <PW>1222222222222</PW>
</User>
```

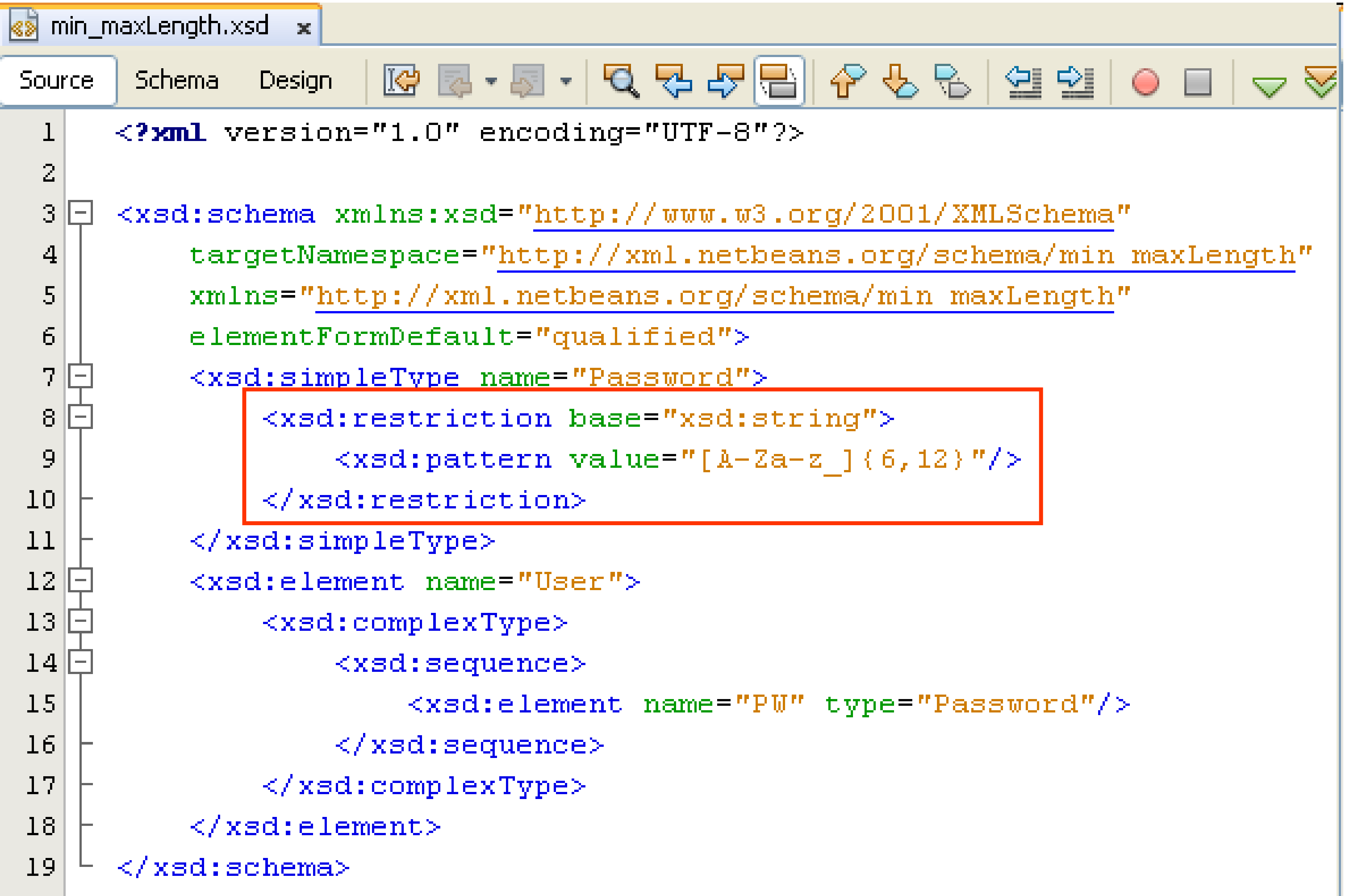


Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/lengthFacets.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/lengthFacets.xsd".
cvc-length-valid: Value '1222222222222' with length = '13' is not facet-valid with respect to length '20' for type 'Password'. [6]
cvc-type.3.1.3: The value '1222222222222' of element 'PW' is not valid. [6]
XML validation finished.
```

Schemas

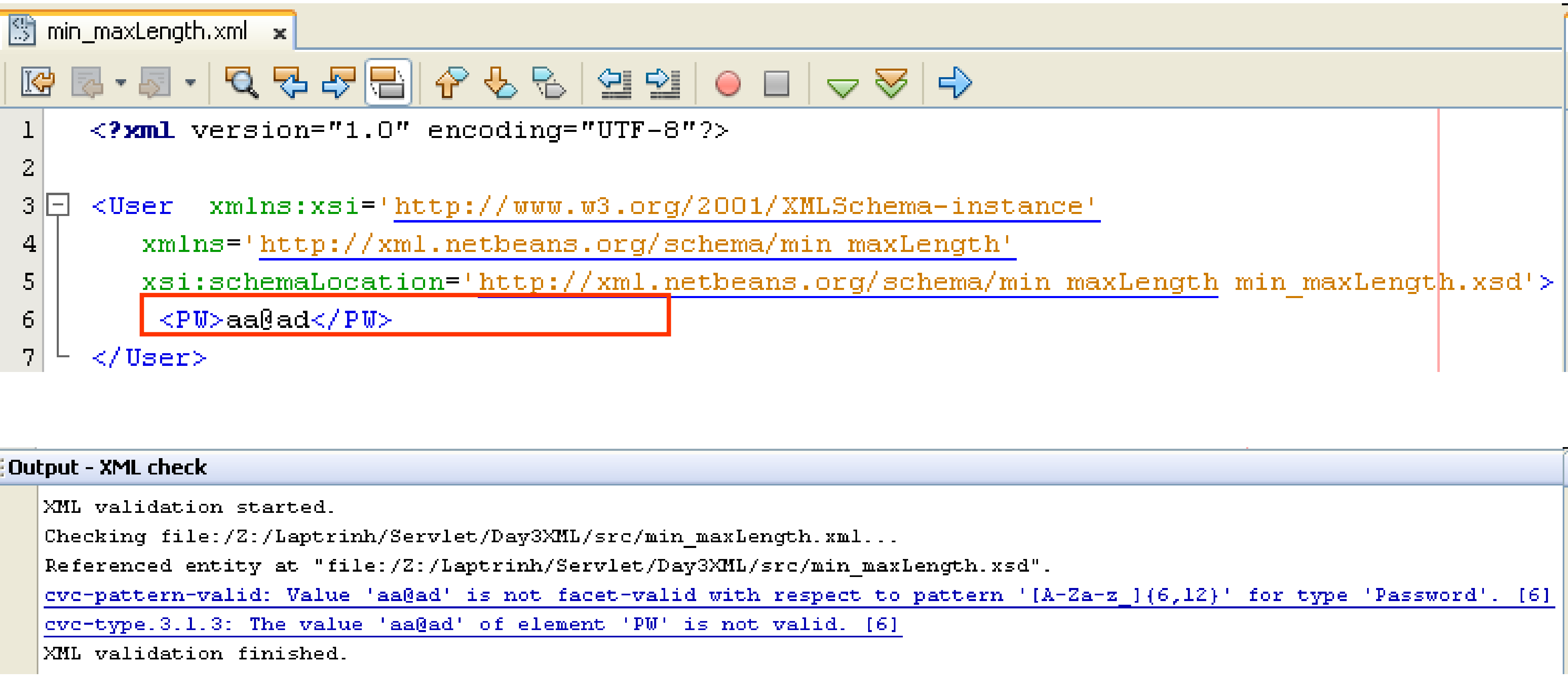
Structuring – XSD Facets – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/min_maxLength"
    xmlns="http://xml.netbeans.org/schema/min_maxLength"
    elementFormDefault="qualified">
    <xsd:simpleType name="Password">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="[A-Za-z_]{6,12}"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="User">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="PW" type="Password"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas

Structuring – XSD Facets – Example



min_maxLength.xml

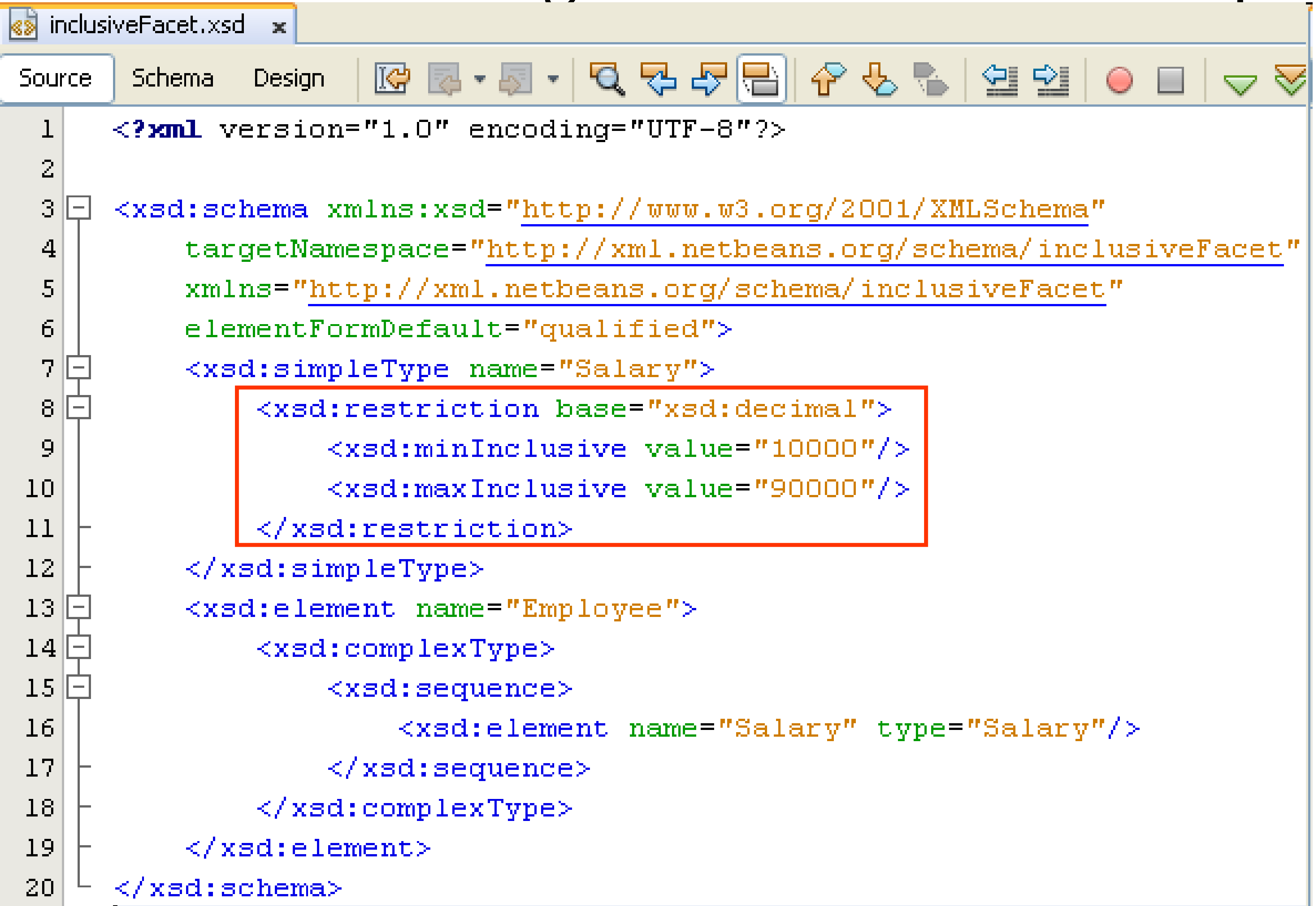
```
<?xml version="1.0" encoding="UTF-8"?>

<User  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/min_maxLength'
   xsi:schemaLocation='http://xml.netbeans.org/schema/min_maxLength min_maxLength.xsd'>
    <PW>aa@ad</PW>
</User>
```

Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/min_maxLength.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/min_maxLength.xsd".
cvc-pattern-valid: Value 'aa@ad' is not facet-valid with respect to pattern '[A-Za-z_]{6,12}' for type 'Password'. [6]
cvc-type.3.1.3: The value 'aa@ad' of element 'PW' is not valid. [6]
XML validation finished.
```

Schemas

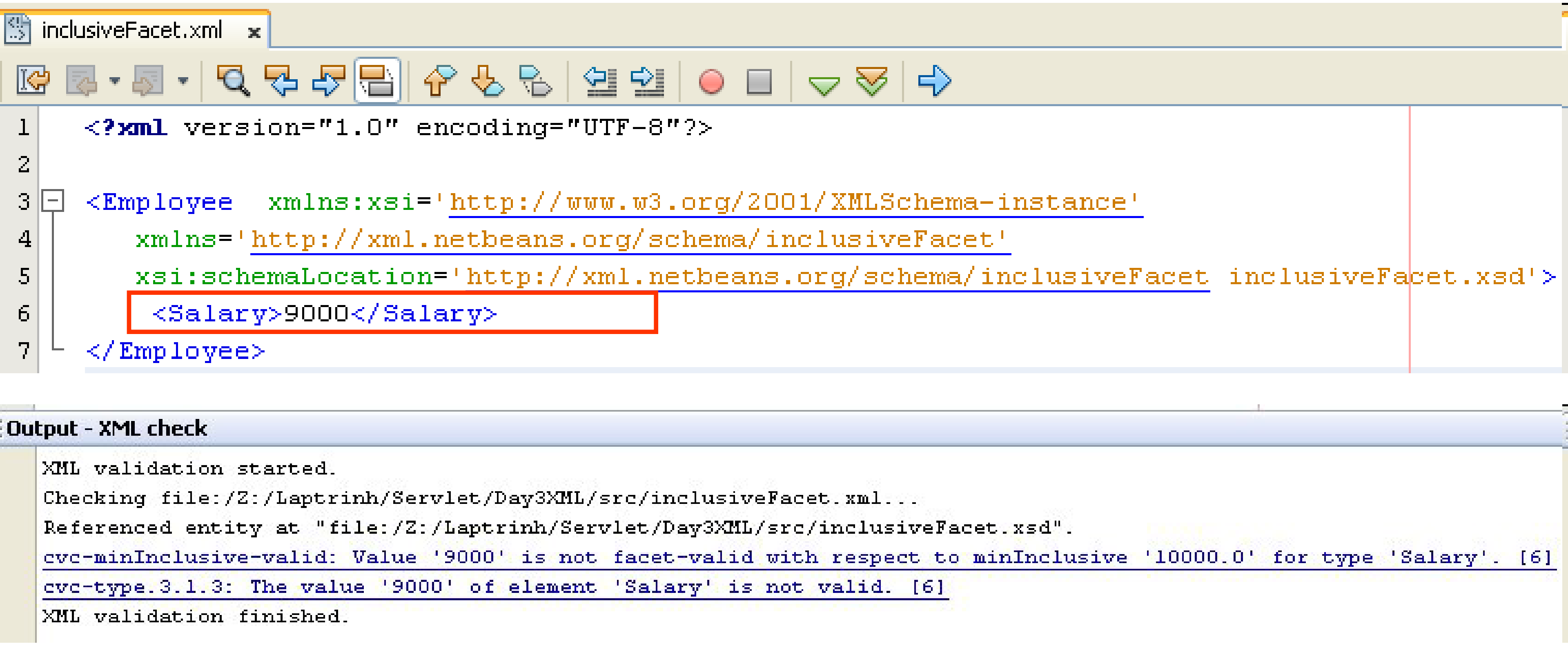
Structuring – XSD Facets – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/inclusiveFacet"
    xmlns="http://xml.netbeans.org/schema/inclusiveFacet"
    elementFormDefault="qualified">
    <xsd:simpleType name="Salary">
        <xsd:restriction base="xsd:decimal">
            <xsd:minInclusive value="10000"/>
            <xsd:maxInclusive value="90000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="Employee">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Salary" type="Salary"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas

Structuring – XSD Facets – Example



inclusiveFacet.xml

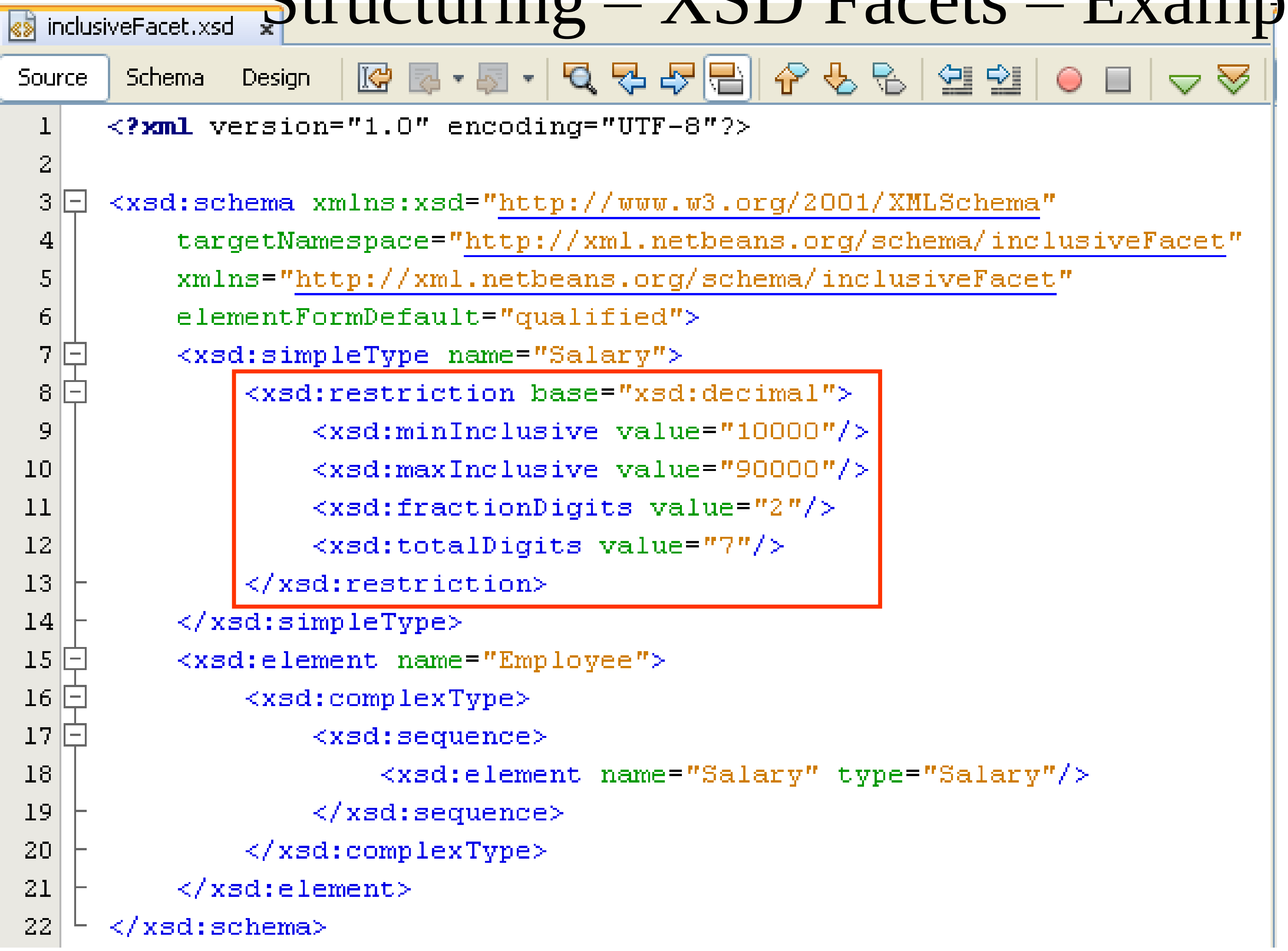
```
<?xml version="1.0" encoding="UTF-8"?>

<Employee  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/inclusiveFacet'
   xsi:schemaLocation='http://xml.netbeans.org/schema/inclusiveFacet inclusiveFacet.xsd'>
    <Salary>9000</Salary>
</Employee>
```

Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/inclusiveFacet.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/inclusiveFacet.xsd".
cvc-minInclusive-valid: Value '9000' is not facet-valid with respect to minInclusive '10000.0' for type 'Salary'. [6]
cvc-type.3.1.3: The value '9000' of element 'Salary' is not valid. [6]
XML validation finished.
```

Schemas

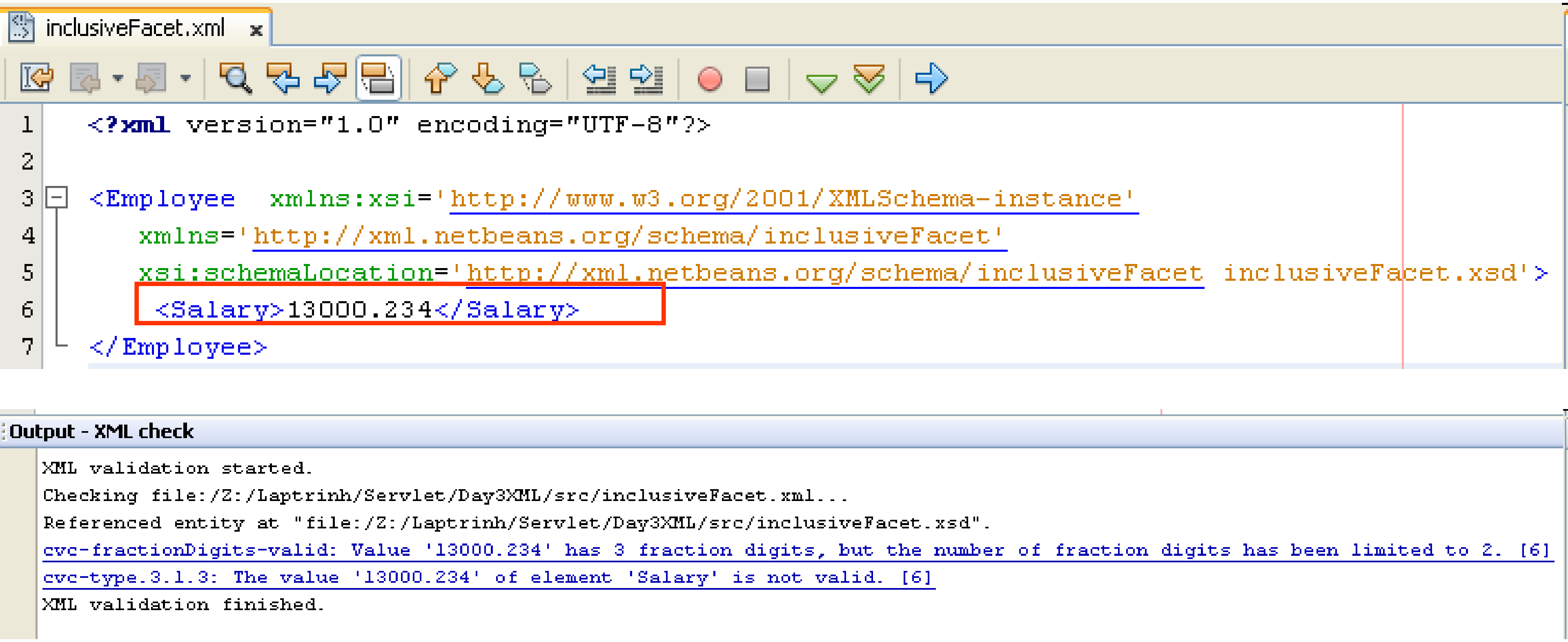
Structuring – XSD Facets – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/inclusiveFacet"
    xmlns="http://xml.netbeans.org/schema/inclusiveFacet"
    elementFormDefault="qualified">
    <xsd:simpleType name="Salary">
        <xsd:restriction base="xsd:decimal">
            <xsd:minInclusive value="10000"/>
            <xsd:maxInclusive value="90000"/>
            <xsd:fractionDigits value="2"/>
            <xsd:totalDigits value="7"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="Employee">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Salary" type="Salary"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas

Structuring – XSD Facets – Example



inclusiveFacet.xml

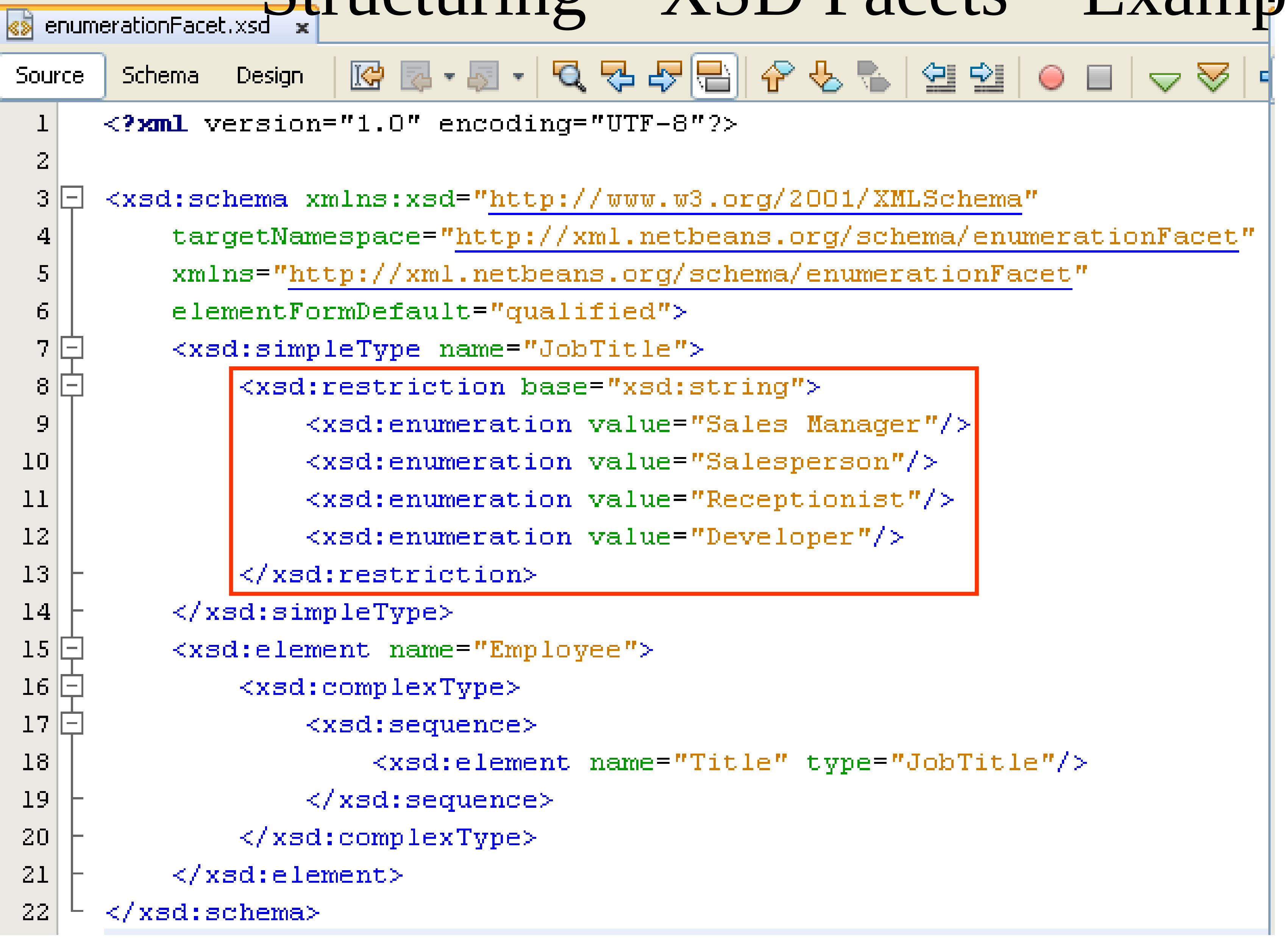
```xml
<?xml version="1.0" encoding="UTF-8"?>

<Employee  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/inclusiveFacet'
   xsi:schemaLocation='http://xml.netbeans.org/schema/inclusiveFacet inclusiveFacet.xsd'>
    <Salary>13000.234</Salary>
</Employee>
```

Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/inclusiveFacet.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/inclusiveFacet.xsd".
cvc-fractionDigits-valid: Value '13000.234' has 3 fraction digits, but the number of fraction digits has been limited to 2. [6]
cvc-type.3.1.3: The value '13000.234' of element 'Salary' is not valid. [6]
XML validation finished.
```

Schemas

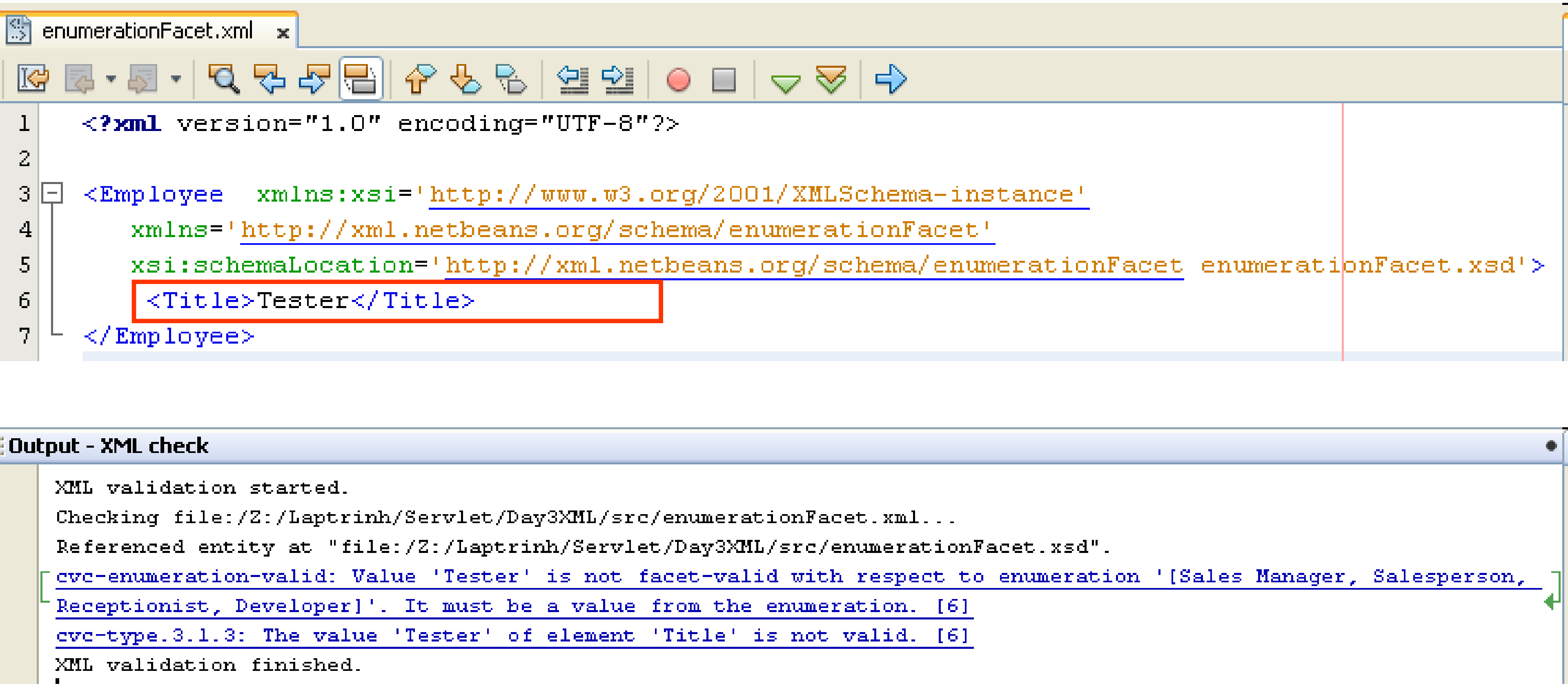
Structuring – XSD Facets – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/enumerationFacet"
    xmlns="http://xml.netbeans.org/schema/enumerationFacet"
    elementFormDefault="qualified">
    <xsd:simpleType name="JobTitle">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Sales Manager"/>
            <xsd:enumeration value="Salesperson"/>
            <xsd:enumeration value="Receptionist"/>
            <xsd:enumeration value="Developer"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="Employee">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Title" type="JobTitle"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas

Structuring – XSD Facets – Example


enumerationFacet.xml

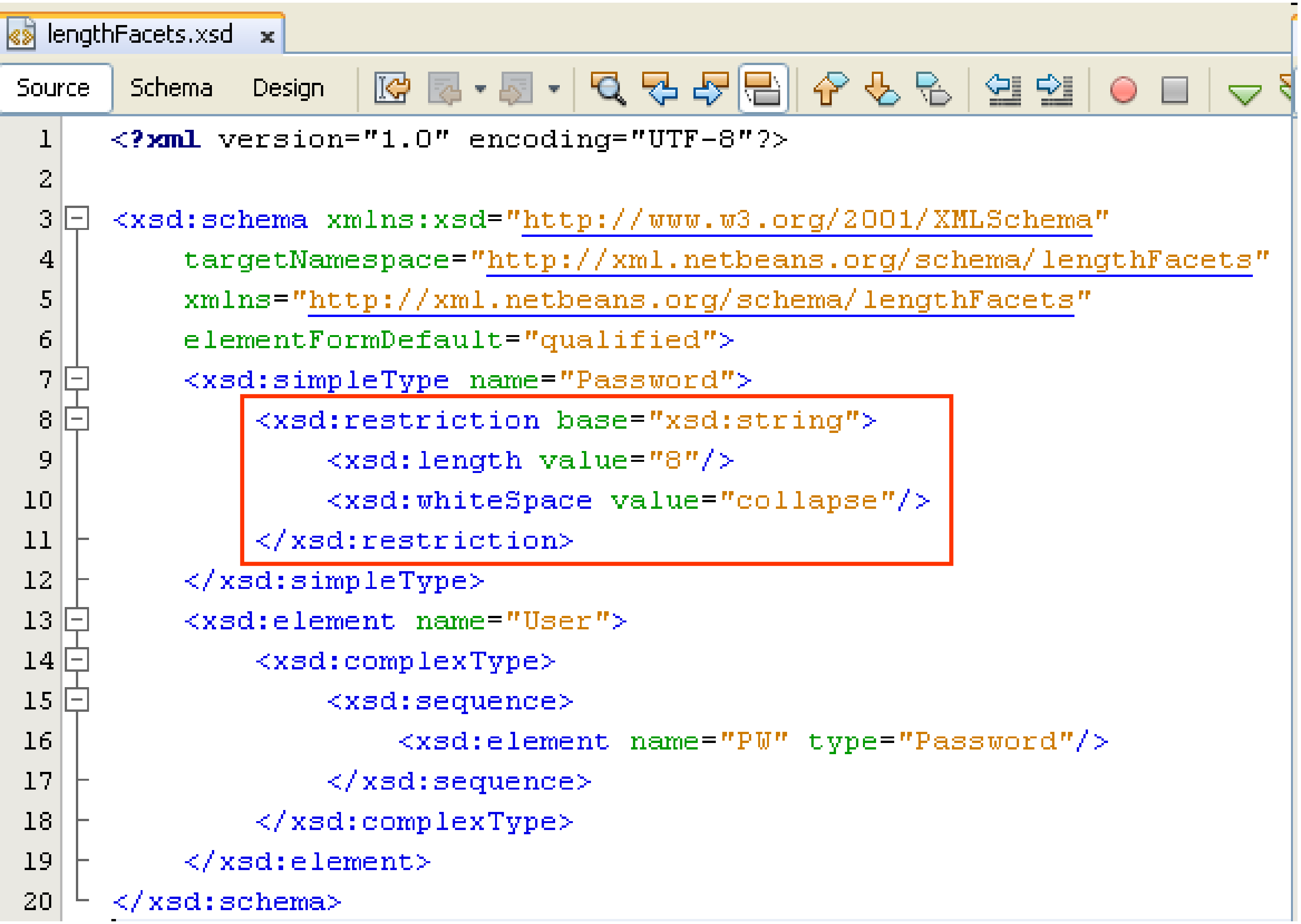
```
<?xml version="1.0" encoding="UTF-8"?>

<Employee  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/enumerationFacet'
   xsi:schemaLocation='http://xml.netbeans.org/schema/enumerationFacet enumerationFacet.xsd'>
    <Title>Tester</Title>
</Employee>
```

Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/enumerationFacet.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/enumerationFacet.xsd".
cvc-enumeration-valid: Value 'Tester' is not facet-valid with respect to enumeration '[Sales Manager, Salesperson,
Receptionist, Developer]'. It must be a value from the enumeration. [6]
cvc-type.3.1.3: The value 'Tester' of element 'Title' is not valid. [6]
XML validation finished.
```

Schemas

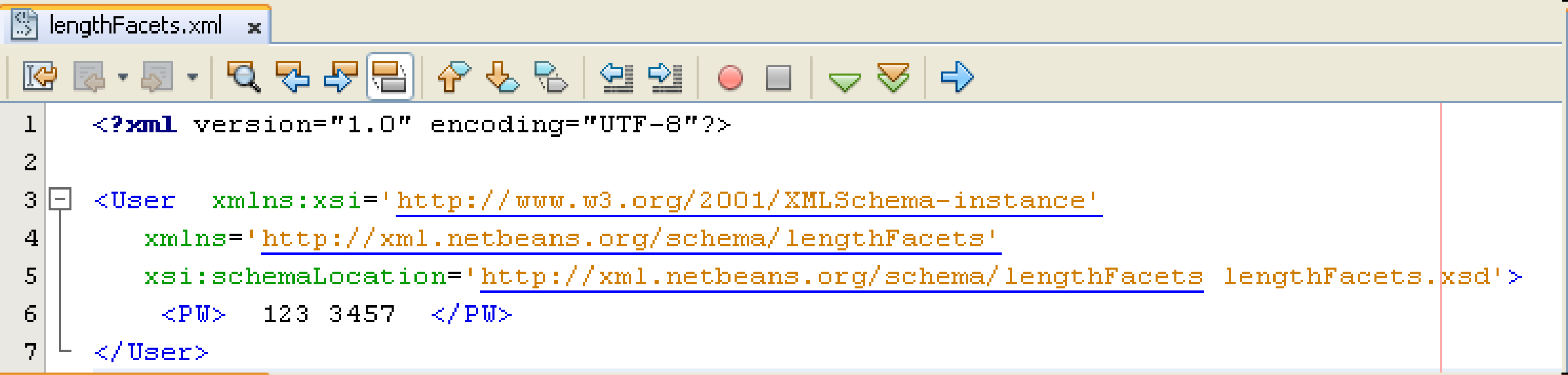
Structuring – XSD Facets – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/lengthFacets"
    xmlns="http://xml.netbeans.org/schema/lengthFacets"
    elementFormDefault="qualified">
    <xsd:simpleType name="Password">
        <xsd:restriction base="xsd:string">
            <xsd:length value="8"/>
            <xsd:whiteSpace value="collapse"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="User">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="PW" type="Password"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

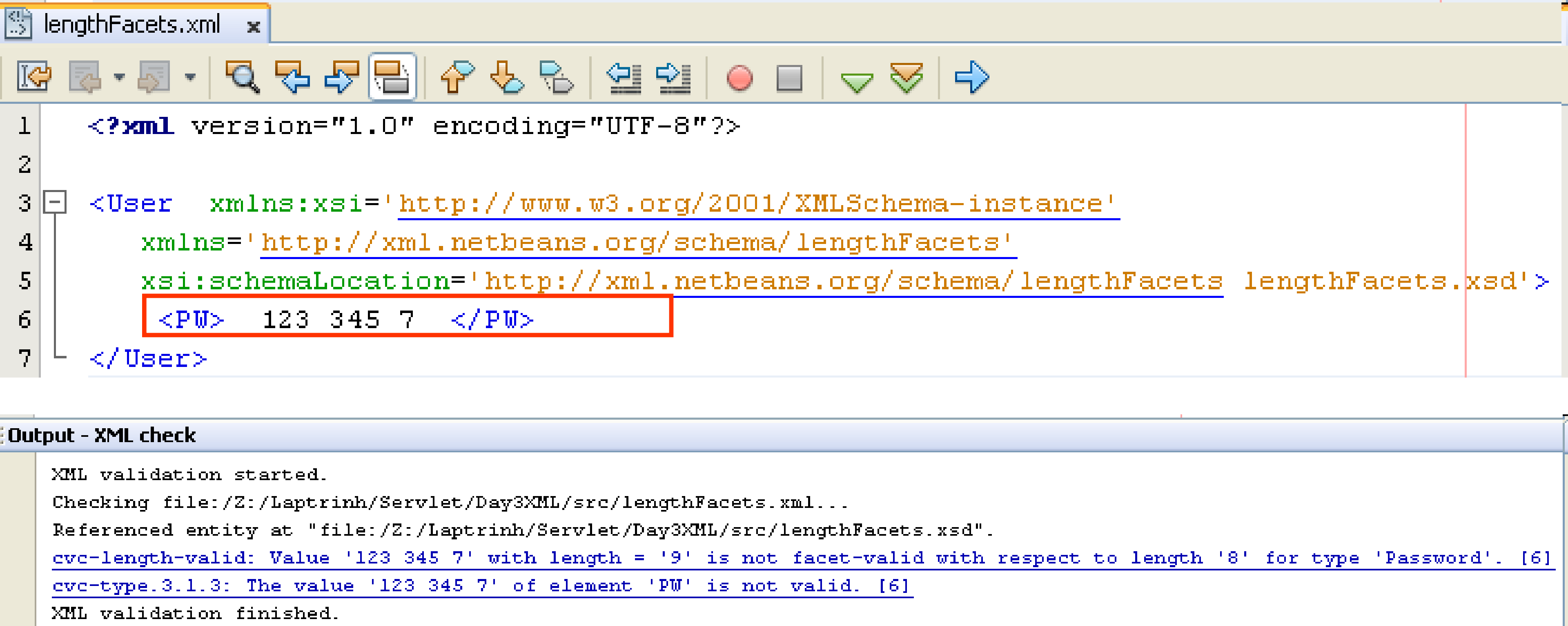
Schemas

Structuring – XSD Facets – Example



lengthFacets.xml

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

<User  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/lengthFacets'
    xsi:schemaLocation='http://xml.netbeans.org/schema/lengthFacets lengthFacets.xsd'>
     <PW>  123 3457  </PW>
</User>
```



lengthFacets.xml

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

<User  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/lengthFacets'
    xsi:schemaLocation='http://xml.netbeans.org/schema/lengthFacets lengthFacets.xsd'>
    <PW>  123 345 7  </PW>
</User>
```

Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/lengthFacets.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/lengthFacets.xsd".
cvc-length-valid: Value '123 345 7' with length = '9' is not facet-valid with respect to length '8' for type 'Password'. [6]
cvc-type.3.1.3: The value '123 345 7' of element 'PW' is not valid. [6]
XML validation finished.
```

Schemas
Structuring – XSD Complex Type Elements

- **Have** attributes, child elements, or some **combination** of the two
  - May contain **nested elements and have attributes**
- Have **04 variations**
  - **Empty** elements
    - Optionally specify attribute types, but **do not permit content**
  - **Only** elements (**simple** element)
    - Can only contain elements and **do not contain** any attributes
      - **However**, either they can be contained attribute, or **their sub elements** are
  - **Text-Only** (simple content)
    - Contains only **simple content** (text and attributes)
    - A **simpleContent** element must be around the content
    - A **extension** or **restriction** tag must be defined within a simpleContent
  - **Mixed**
    - Can contain **text** content as well **as sub-elements within element**
    - They may or may not attributes

Schemas
Structuring – XSD Content Elements

- Is used to **define** an **extension or** a **restriction** of simple or complex type
- **There are 2 types**
  - **Simple Content**
    - Are created by **adding** a **list of attributes to** a **simple type**
    - Appear right **after** we **declare complexType element**
    - Only contains character data (**text**) and/or **attributes** (**if any**).
    - Can be derived by,
      - extensions (**extending simple type by adding attribute**)
      - Or, restrictions **(allows not only restriction of the scope of the text node, but also the restriction of the scope of the attribute)**

Schemas
Structuring – XSD Content Elements

- **There are 2 types (cont)**
  - **Complex Content**
    - Elements that **have child elements**.
    - Appear right **after** we **declare complexType element**
    - **Attributes** for such elements are declared after the element's model group.
    - Can be **derived, by extension or by restriction**, from complex types
      - All the instance structures that match the restricted complex type must also match the base complex type
    - Can be contained simple content in complex content

Fpt University

Schemas

Structuring – XSD Complex Type Elements

- There are **02 ways** to define a complex type
  - The element can be **declared directly** by naming the element

```
<xsd:element name=“element_name">

     <xsd:complexType>
        <xsd:sequence>
            <xsd:element name=“element_name" type=“data types"/>
            . . .
        </xsd:sequence>
     </xsd:complexType>

< /xsd:element>
```

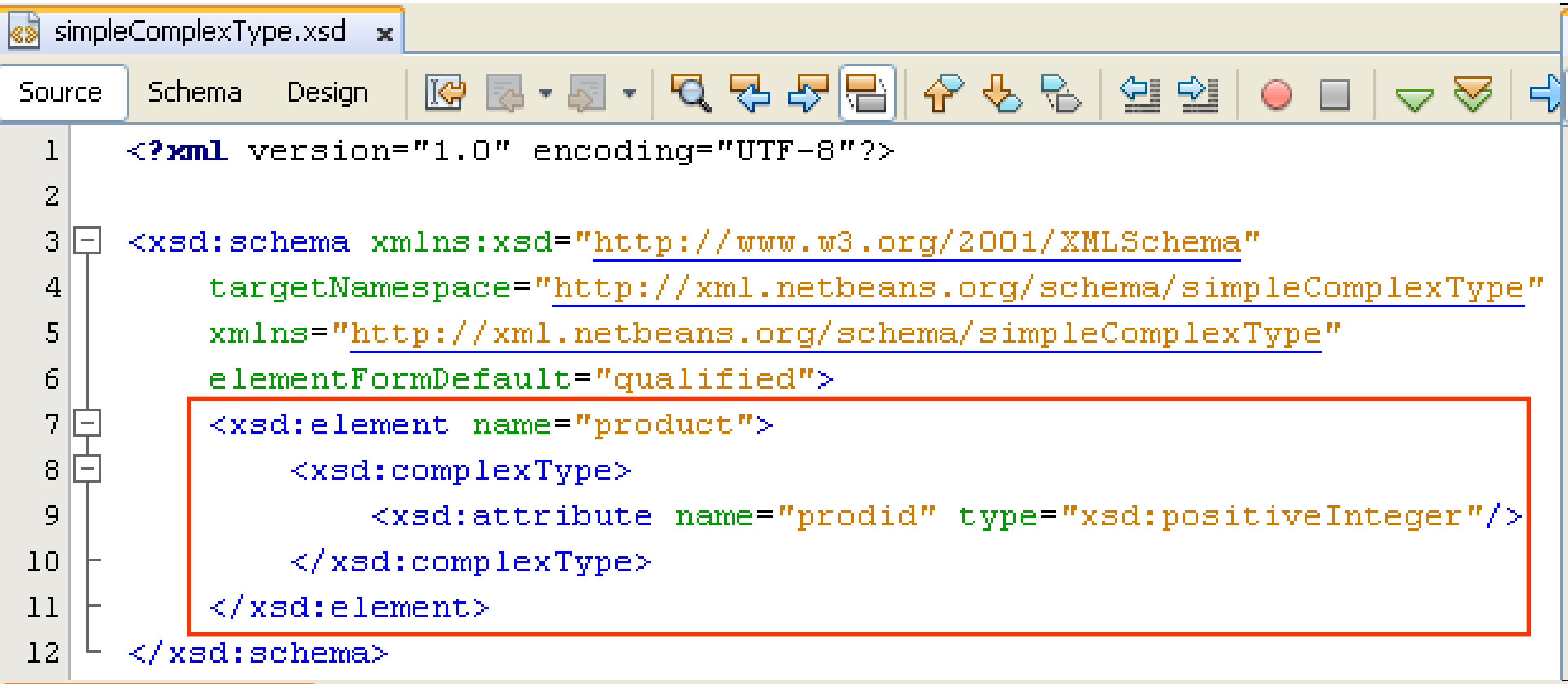
- The **disadvantage** of this is **only the declared element** can use the specified complex type

  - The element can have a type attribute that **refers** to the name of the complex type to use

```
<xsd:element name=“element_name" type=“new_data_type"/>
<xsd:complexType name=“new_data_type">
        <xsd:sequence>
           <xsd:element name=“element_name" type=“data types"/>
           . . .
        </xsd:sequence>

< /xsd:complexType>
```

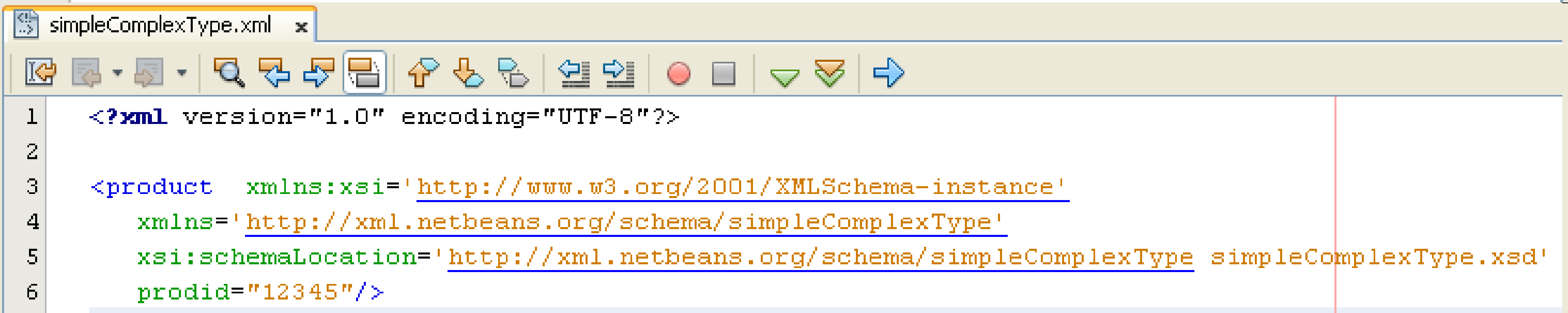
- The **advantage** is **reusable** and **extensible**

Schemas
Structuring – Empty Complex Type – Example


simpleComplexType.xsd

Source Schema Design

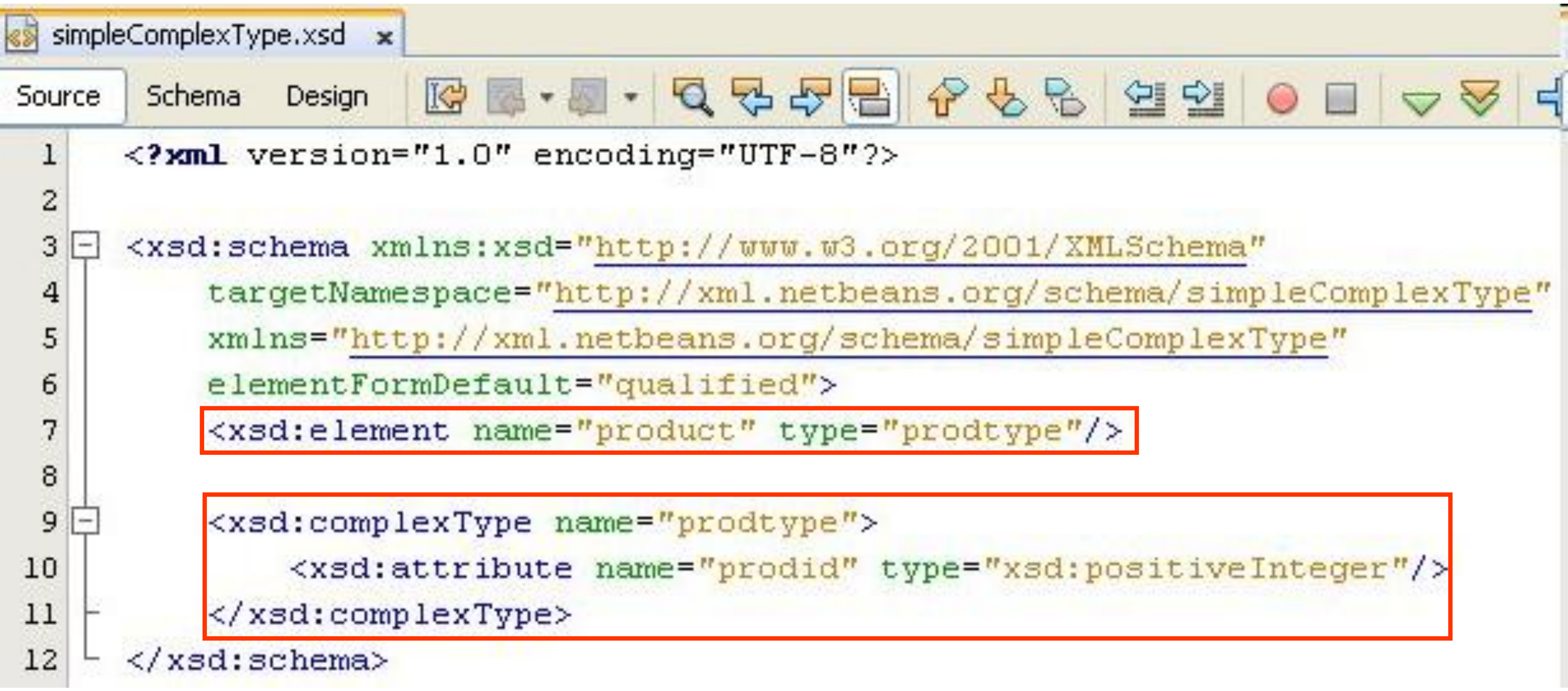
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/simpleComplexType"
    xmlns="http://xml.netbeans.org/schema/simpleComplexType"
    elementFormDefault="qualified">
    <xsd:element name="product">
        <xsd:complexType>
            <xsd:attribute name="prodid" type="xsd:positiveInteger"/>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```


simpleComplexType.xml

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

<product  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/simpleComplexType'
   xsi:schemaLocation='http://xml.netbeans.org/schema/simpleComplexType simpleComplexType.xsd'
   prodid="12345"/>
```

Schemas

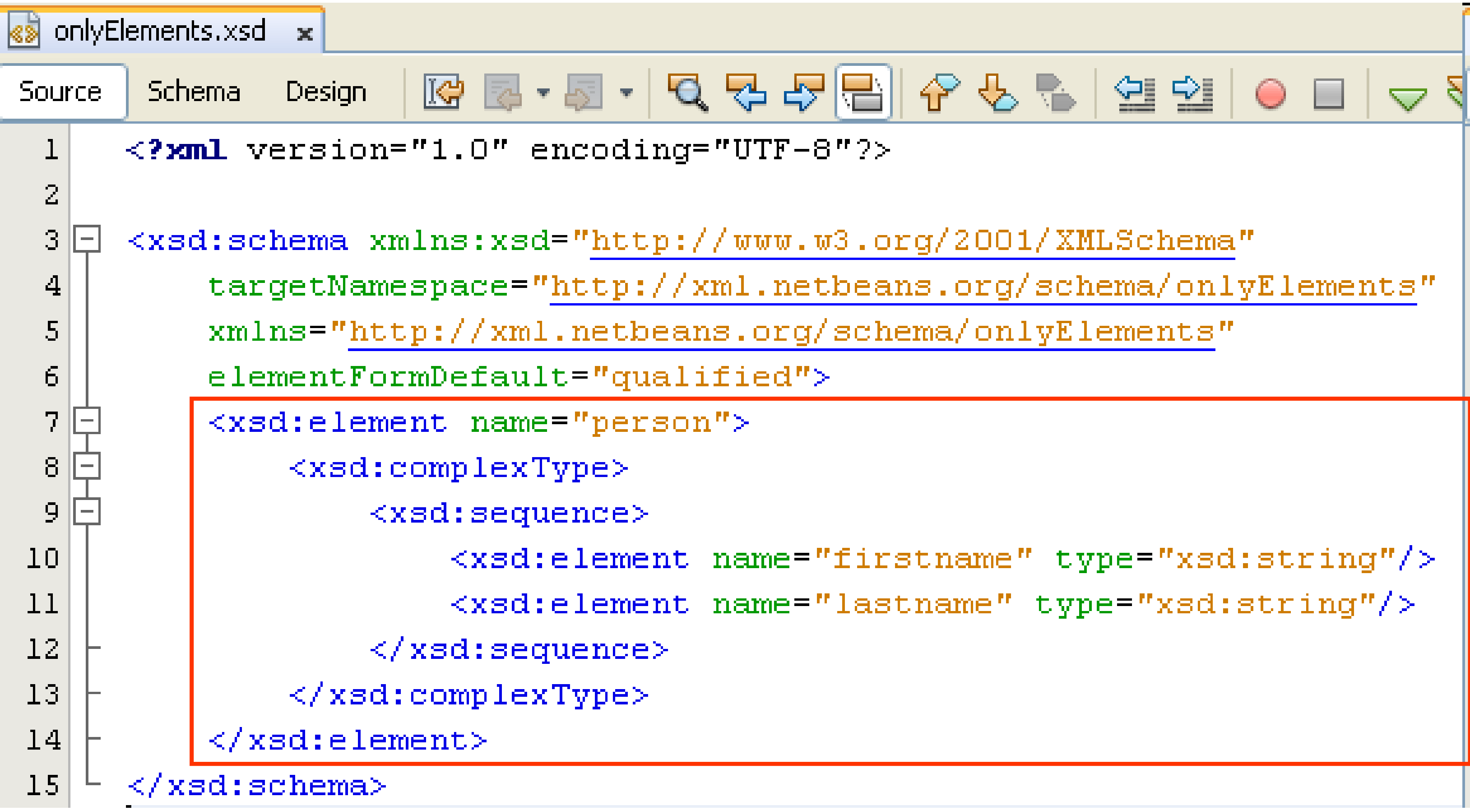
Structuring – Empty Complex Type – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/simpleComplexType"
    xmlns="http://xml.netbeans.org/schema/simpleComplexType"
    elementFormDefault="qualified">
    <xsd:element name="product" type="prodtype"/>

    <xsd:complexType name="prodtype">
        <xsd:attribute name="prodid" type="xsd:positiveInteger"/>
    </xsd:complexType>
</xsd:schema>
```

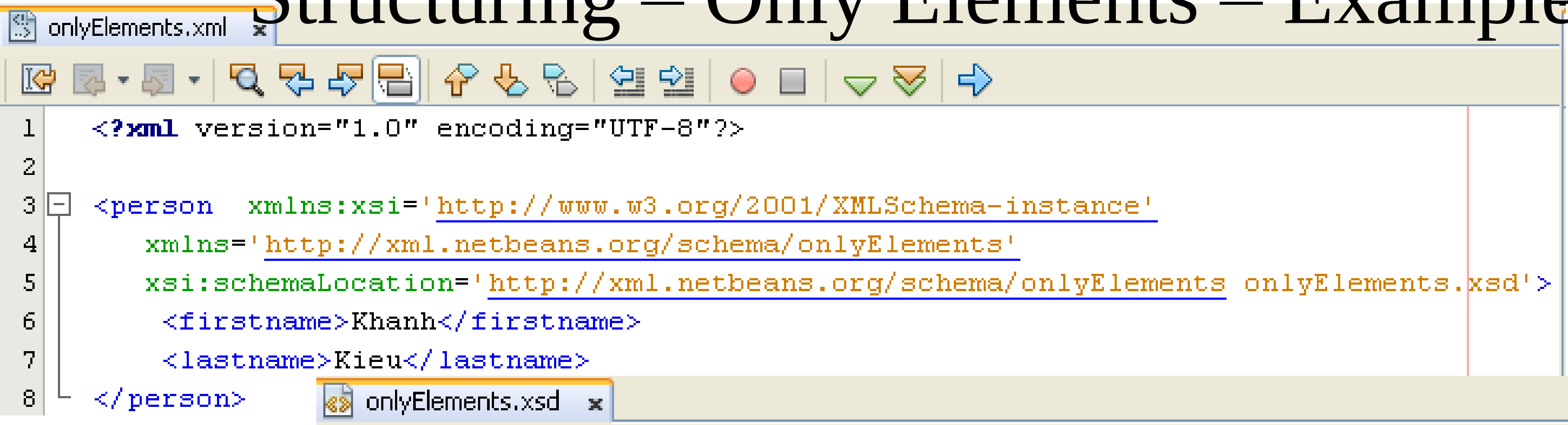
Schemas

Structuring – Only Elements – Example



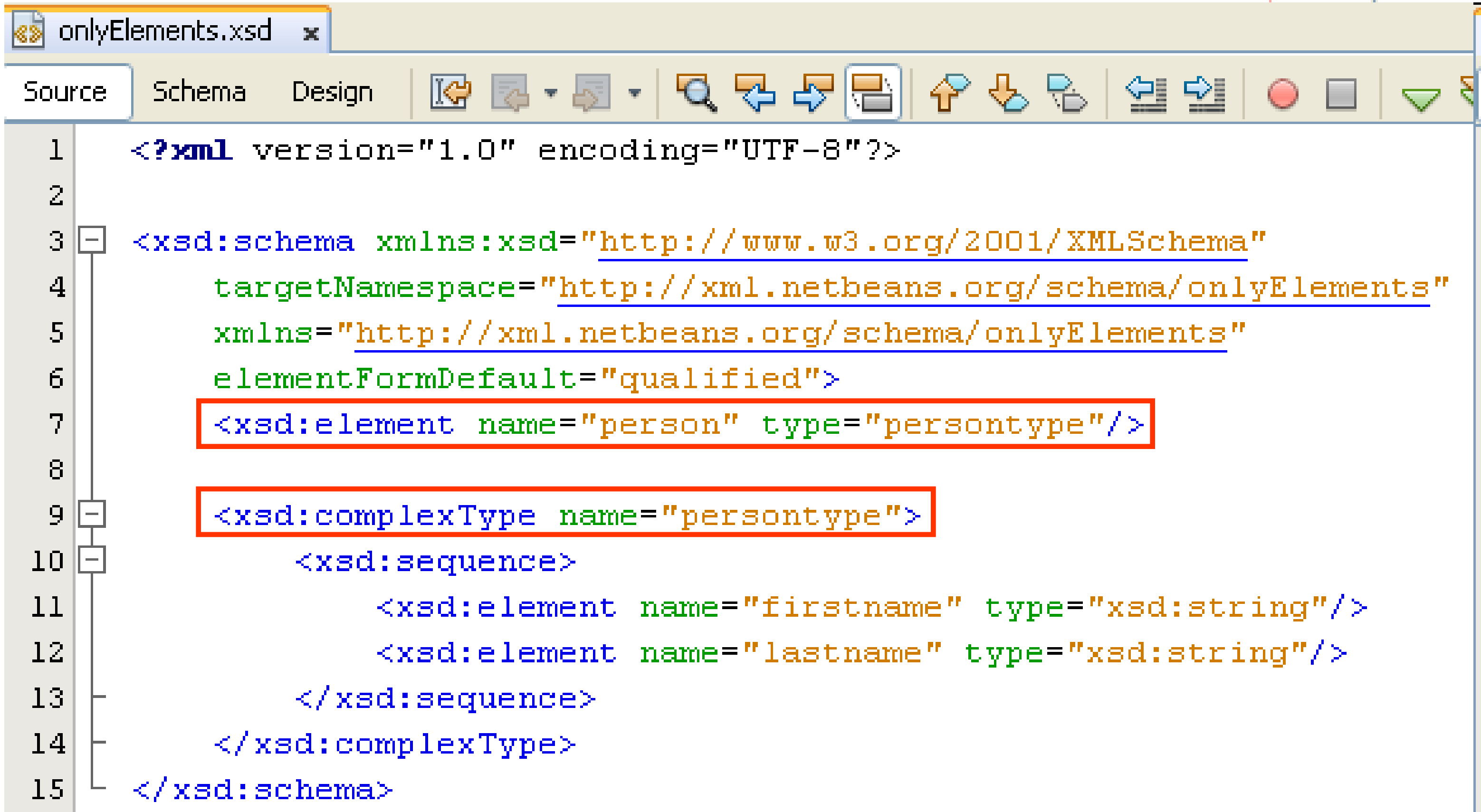
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/onlyElements"
    xmlns="http://xml.netbeans.org/schema/onlyElements"
    elementFormDefault="qualified">
    <xsd:element name="person">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="firstname" type="xsd:string"/>
                <xsd:element name="lastname" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas
Structuring – Only Elements – Example



onlyElements.xml

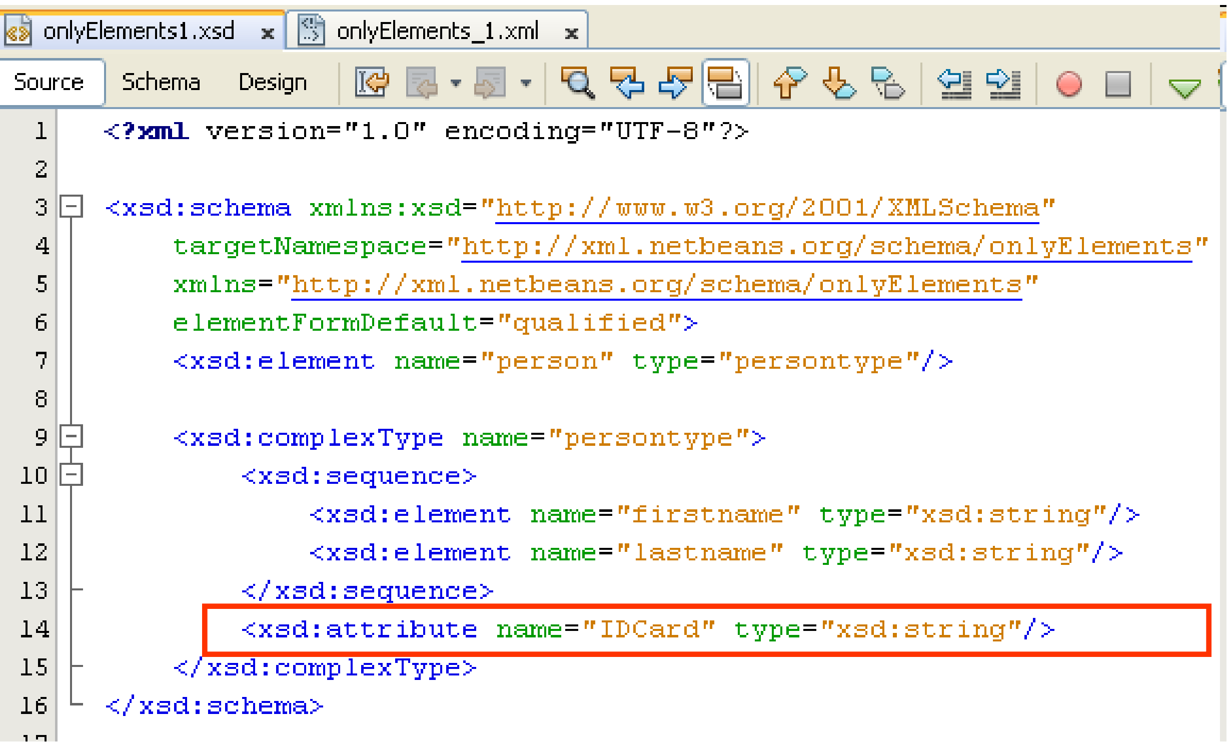
```
<?xml version="1.0" encoding="UTF-8"?>

<person  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/onlyElements'
   xsi:schemaLocation='http://xml.netbeans.org/schema/onlyElements onlyElements.xsd'>
    <firstname>Khanh</firstname>
    <lastname>Kieu</lastname>
</person>
```



onlyElements.xsd

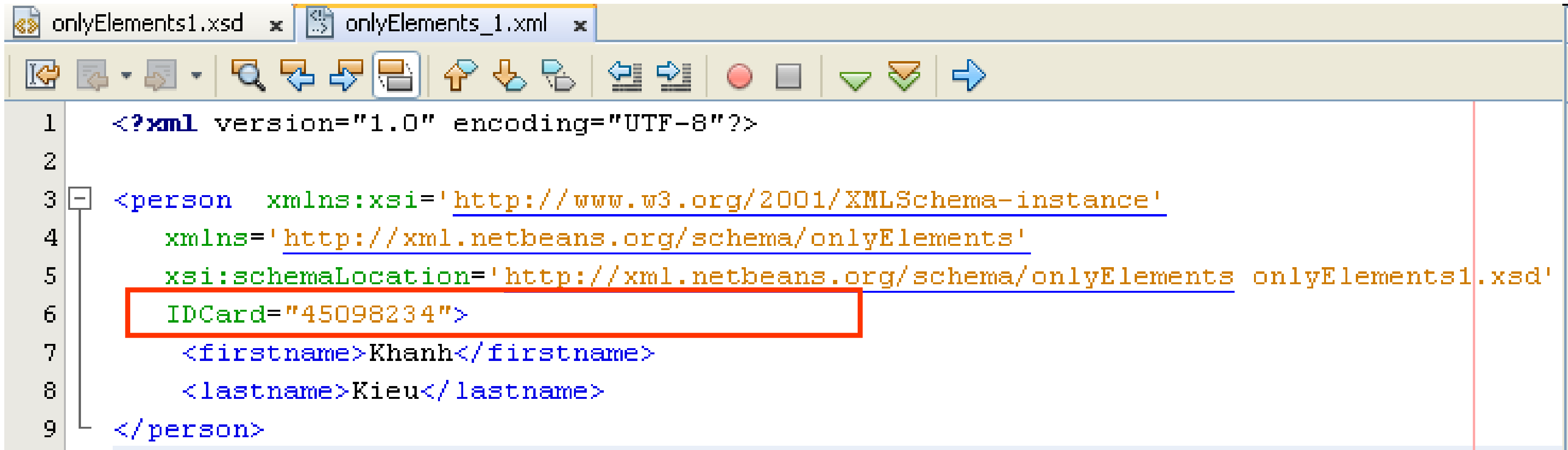
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/onlyElements"
    xmlns="http://xml.netbeans.org/schema/onlyElements"
    elementFormDefault="qualified">
    <xsd:element name="person" type="persontype"/>

    <xsd:complexType name="persontype">
        <xsd:sequence>
            <xsd:element name="firstname" type="xsd:string"/>
            <xsd:element name="lastname" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

Schemas
Structuring – Only Elements – Example


onlyElements1.xsd | onlyElements_1.xml

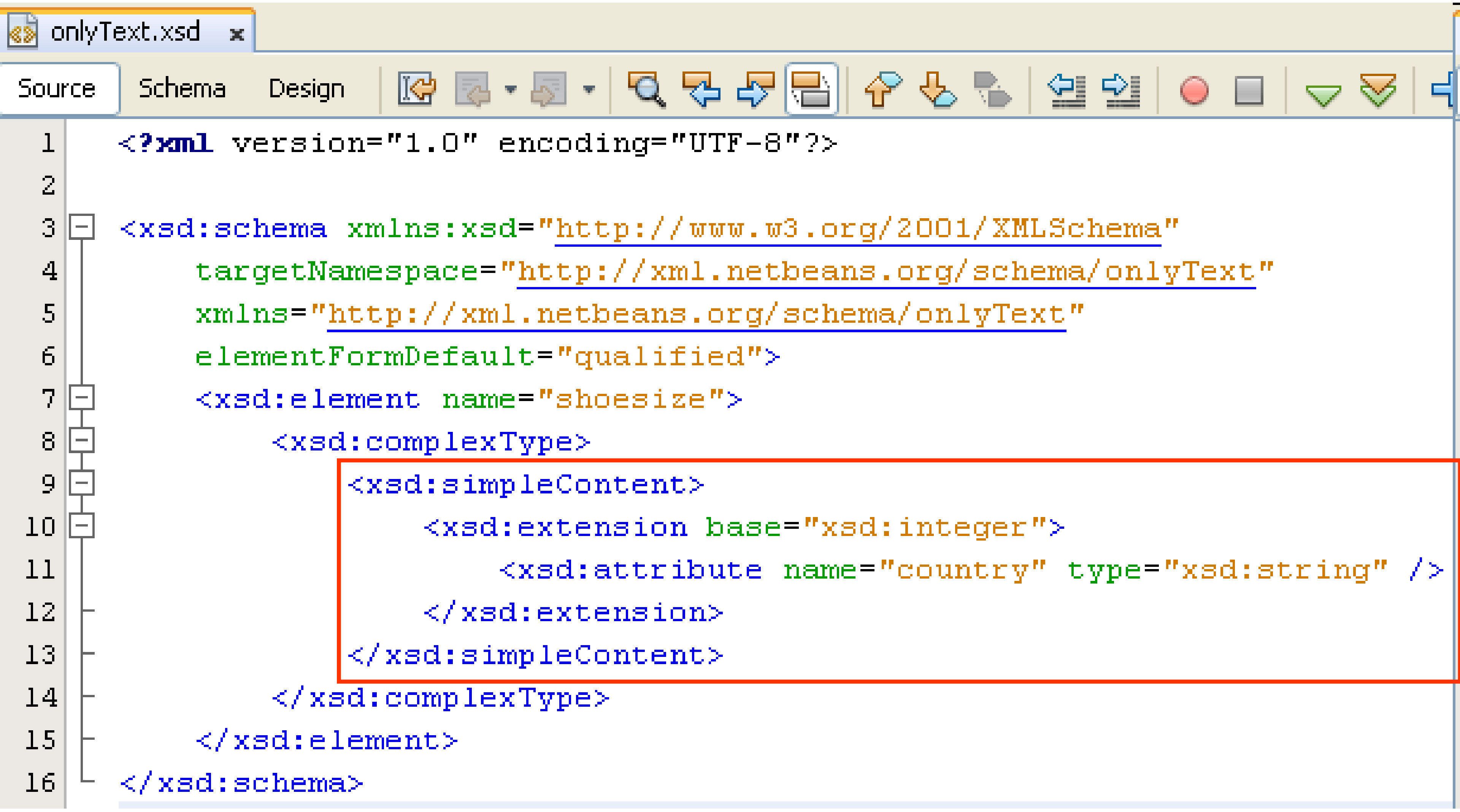
Source | Schema | Design

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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/onlyElements"
    xmlns="http://xml.netbeans.org/schema/onlyElements"
    elementFormDefault="qualified">
    <xsd:element name="person" type="persontype"/>

    <xsd:complexType name="persontype">
        <xsd:sequence>
            <xsd:element name="firstname" type="xsd:string"/>
            <xsd:element name="lastname" type="xsd:string"/>
        </xsd:sequence>
        <xsd:attribute name="IDCard" type="xsd:string"/>
    </xsd:complexType>
</xsd:schema>
```


onlyElements1.xsd | onlyElements_1.xml

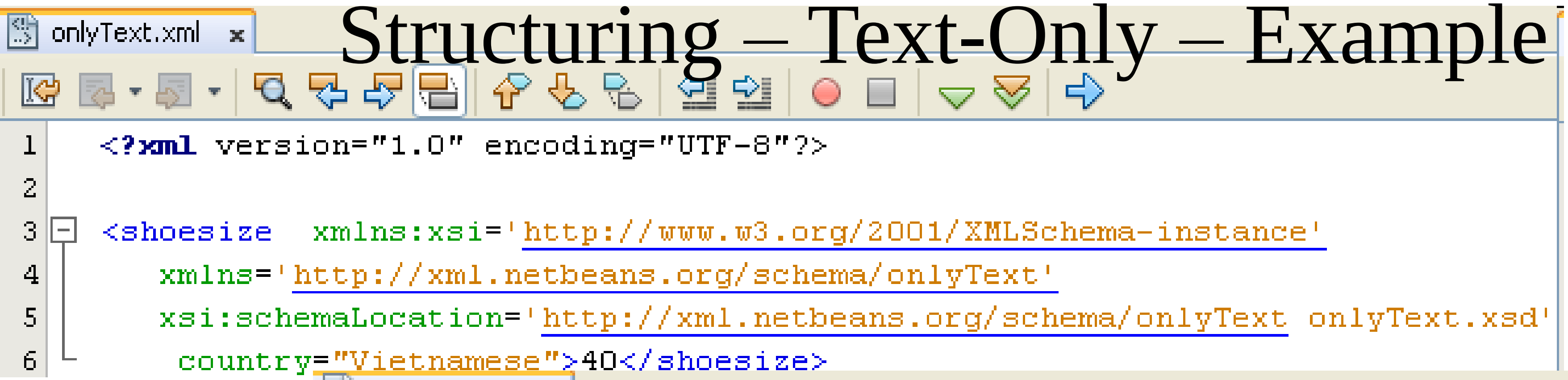
```
<?xml version="1.0" encoding="UTF-8"?>

<person  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/onlyElements'
   xsi:schemaLocation='http://xml.netbeans.org/schema/onlyElements onlyElements1.xsd'
   IDCard="45098234">
    <firstname>Khanh</firstname>
    <lastname>Kieu</ lastname>
</person>
```

Schemas
Structuring – Text-Only – Example



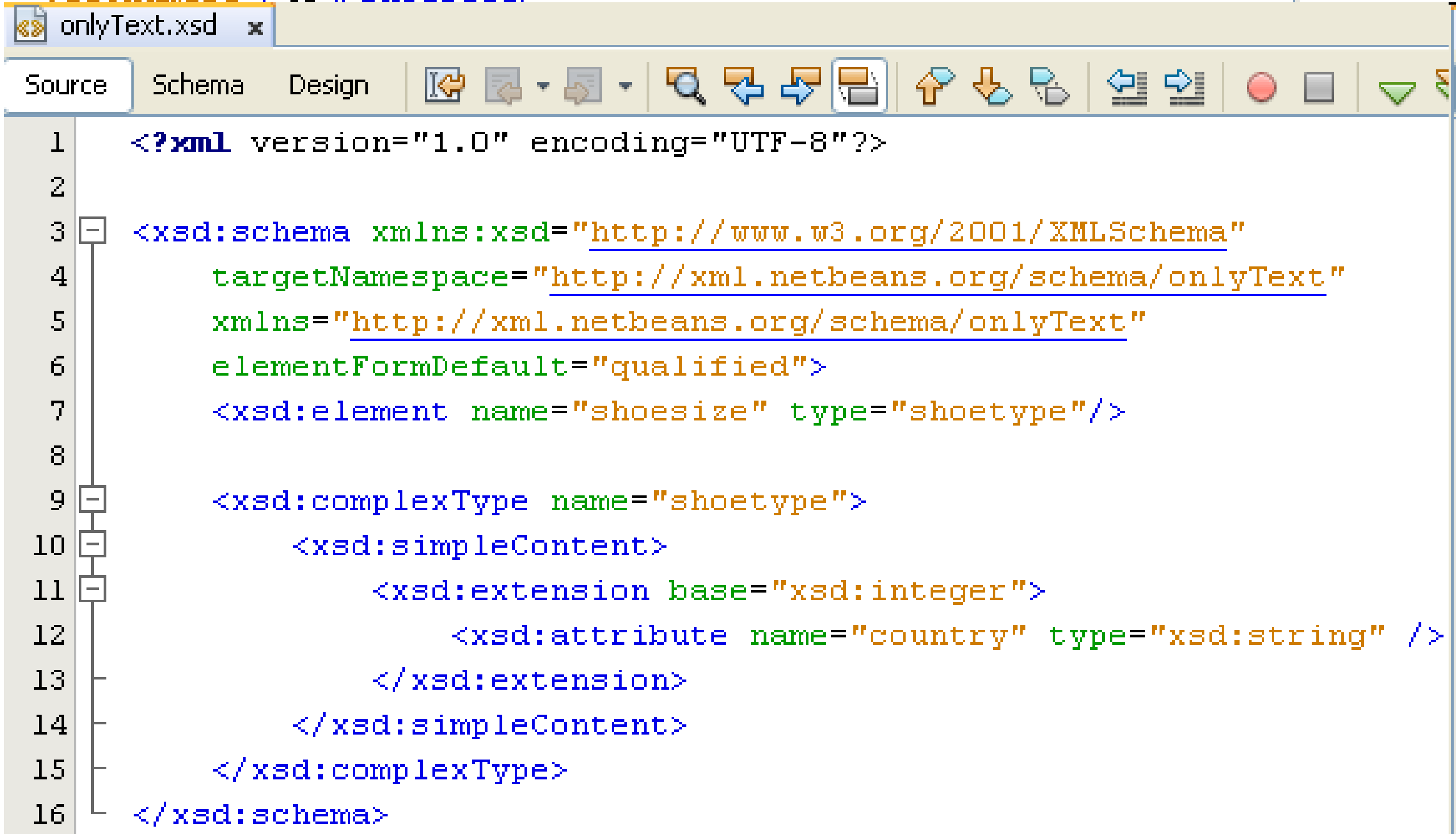
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/onlyText"
    xmlns="http://xml.netbeans.org/schema/onlyText"
    elementFormDefault="qualified">
    <xsd:element name="shoesize">
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base="xsd:integer">
                    <xsd:attribute name="country" type="xsd:string" />
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas
Structuring – Text-Only – Example



onlyText.xml

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

<shoesize  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/onlyText'
   xsi:schemaLocation='http://xml.netbeans.org/schema/onlyText onlyText.xsd'
    country="Vietnamese">40</shoesize>
```



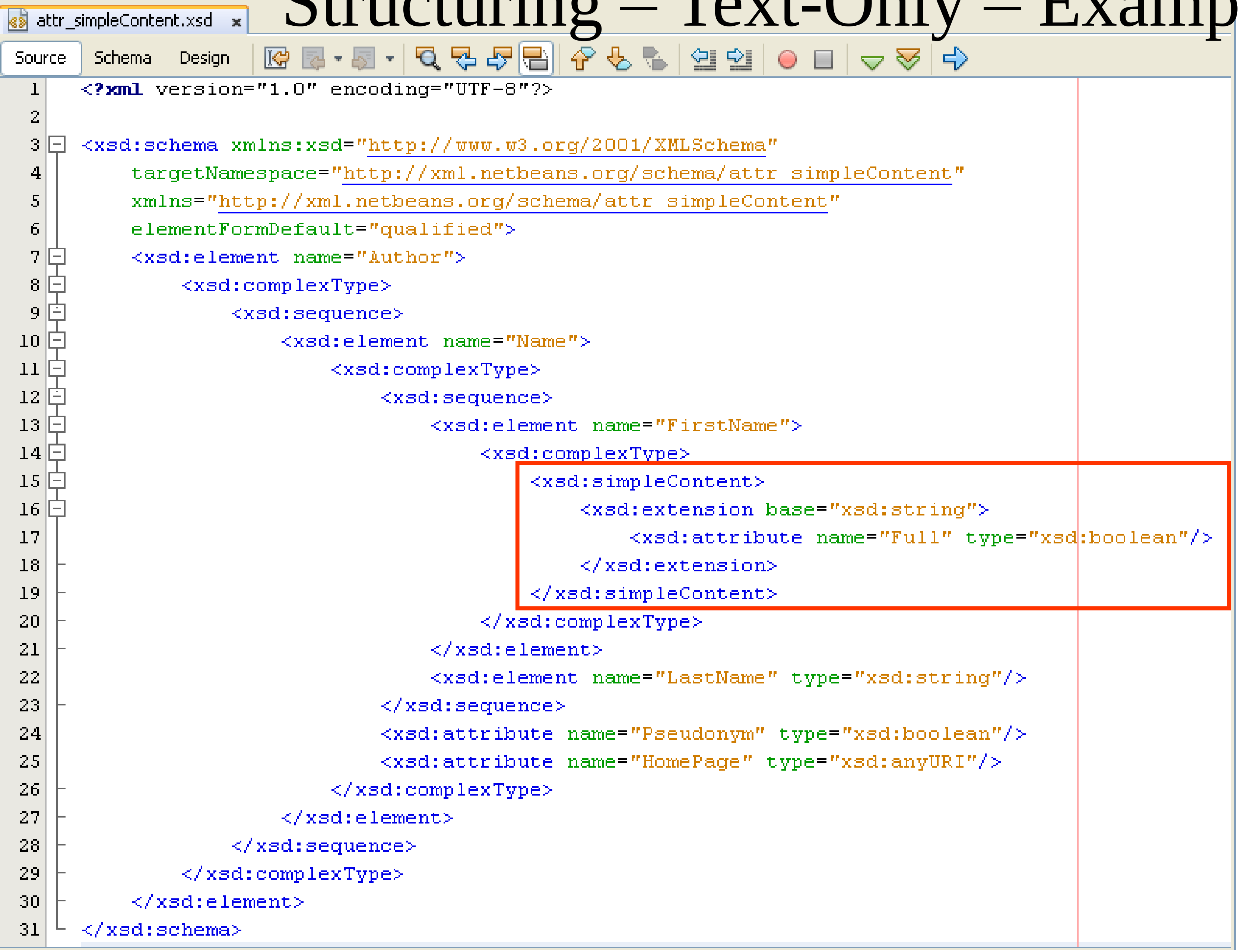
onlyText.xsd

Source Schema Design

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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/onlyText"
    xmlns="http://xml.netbeans.org/schema/onlyText"
    elementFormDefault="qualified">
    <xsd:element name="shoesize" type="shoetype"/>

    <xsd:complexType name="shoetype">
        <xsd:simpleContent>
            <xsd:extension base="xsd:integer">
                <xsd:attribute name="country" type="xsd:string" />
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
</xsd:schema>
```

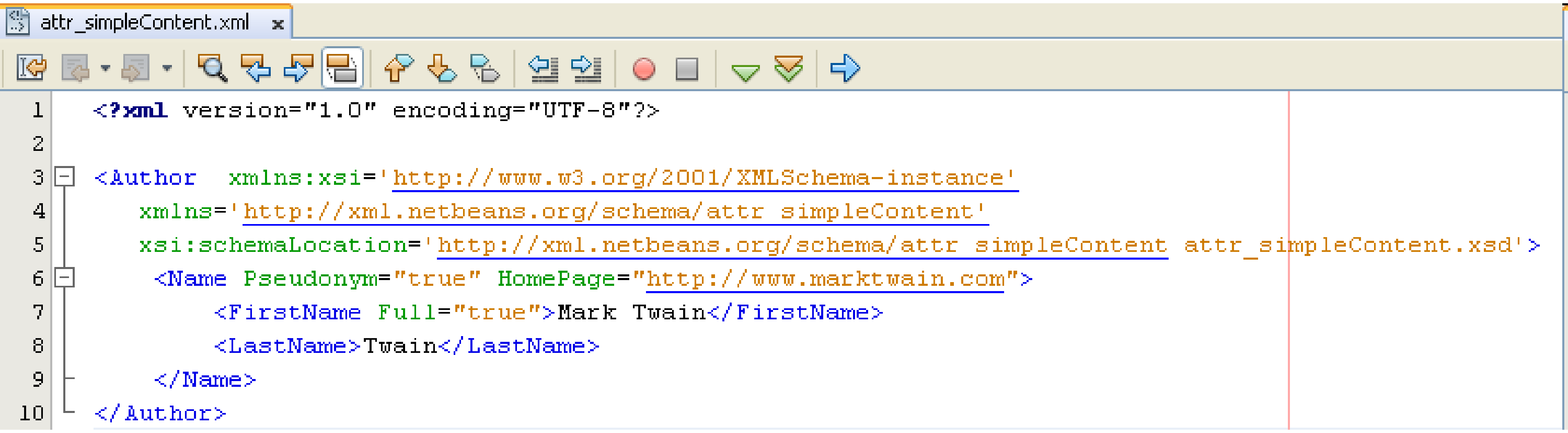
Schemas

Structuring – Text-Only – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/attr_simpleContent"
    xmlns="http://xml.netbeans.org/schema/attr_simpleContent"
    elementFormDefault="qualified">
    <xsd:element name="Author">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Name">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="FirstName">
                                <xsd:complexType>
                                    <xsd:simpleContent>
                                        <xsd:extension base="xsd:string">
                                            <xsd:attribute name="Full" type="xsd:boolean"/>
                                        </xsd:extension>
                                    </xsd:simpleContent>
                                </xsd:complexType>
                            </xsd:element>
                            <xsd:element name="LastName" type="xsd:string"/>
                        </xsd:sequence>
                        <xsd:attribute name="Pseudonym" type="xsd:boolean"/>
                        <xsd:attribute name="HomePage" type="xsd:anyURI"/>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

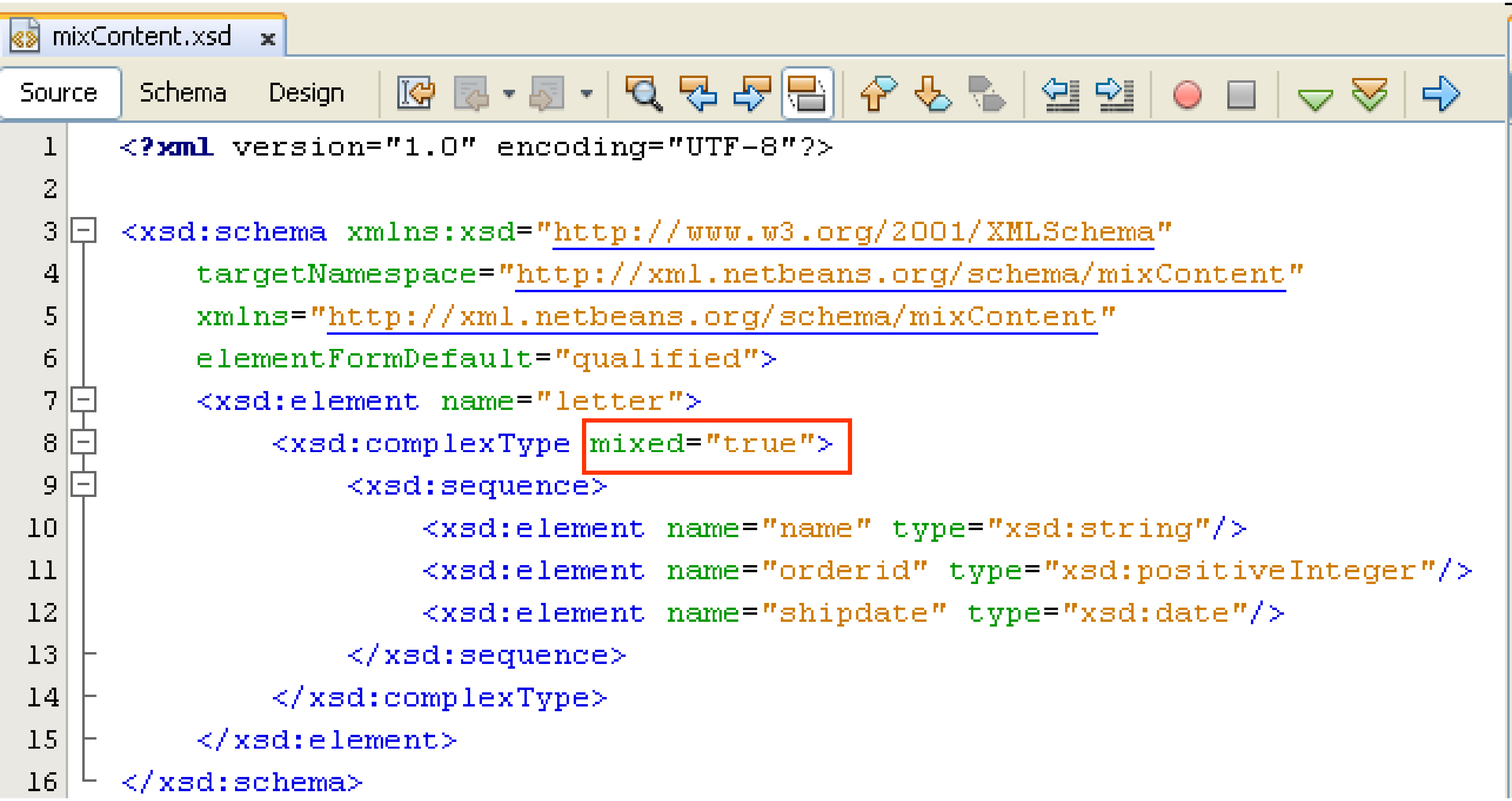
Schemas
Structuring – Text-Only – Example


attr_simpleContent.xml

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

<Author  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/attr_simpleContent'
   xsi:schemaLocation='http://xml.netbeans.org/schema/attr_simpleContent attr_simpleContent.xsd'>
    <Name Pseudonym="true" HomePage="http://www.marktwain.com">
        <FirstName Full="true">Mark Twain</FirstName>
        <LastName>Twain</LastName>
    </Name>
</Author>
```

Schemas

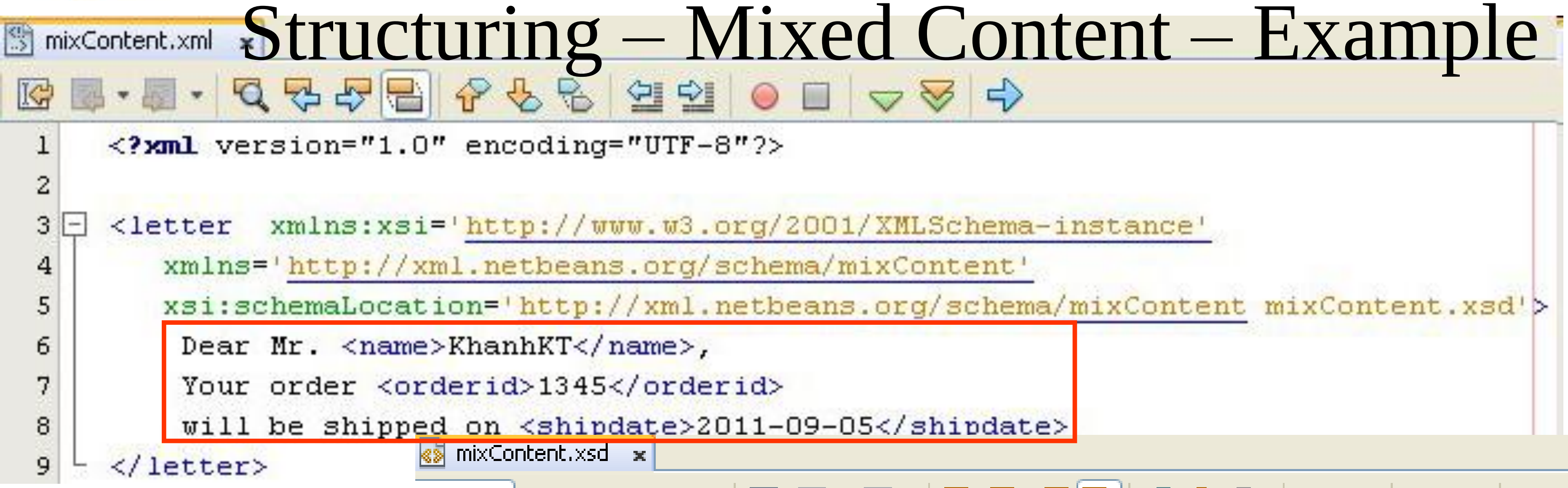
Structuring – Mixed Content – Example



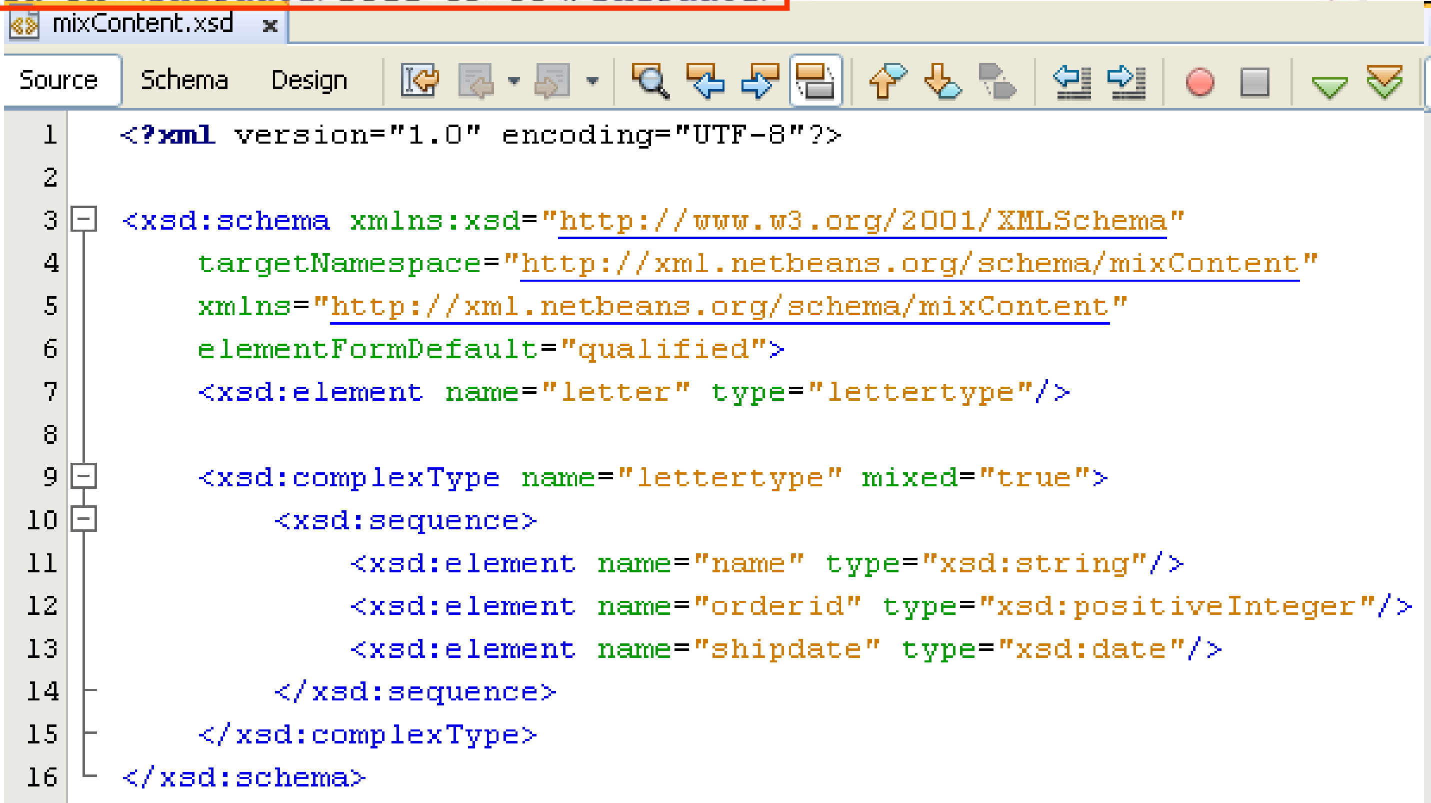
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/mixContent"
    xmlns="http://xml.netbeans.org/schema/mixContent"
    elementFormDefault="qualified">
    <xsd:element name="letter">
        <xsd:complexType mixed="true">
            <xsd:sequence>
                <xsd:element name="name" type="xsd:string"/>
                <xsd:element name="orderid" type="xsd:positiveInteger"/>
                <xsd:element name="shipdate" type="xsd:date"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas

Structuring – Mixed Content – Example



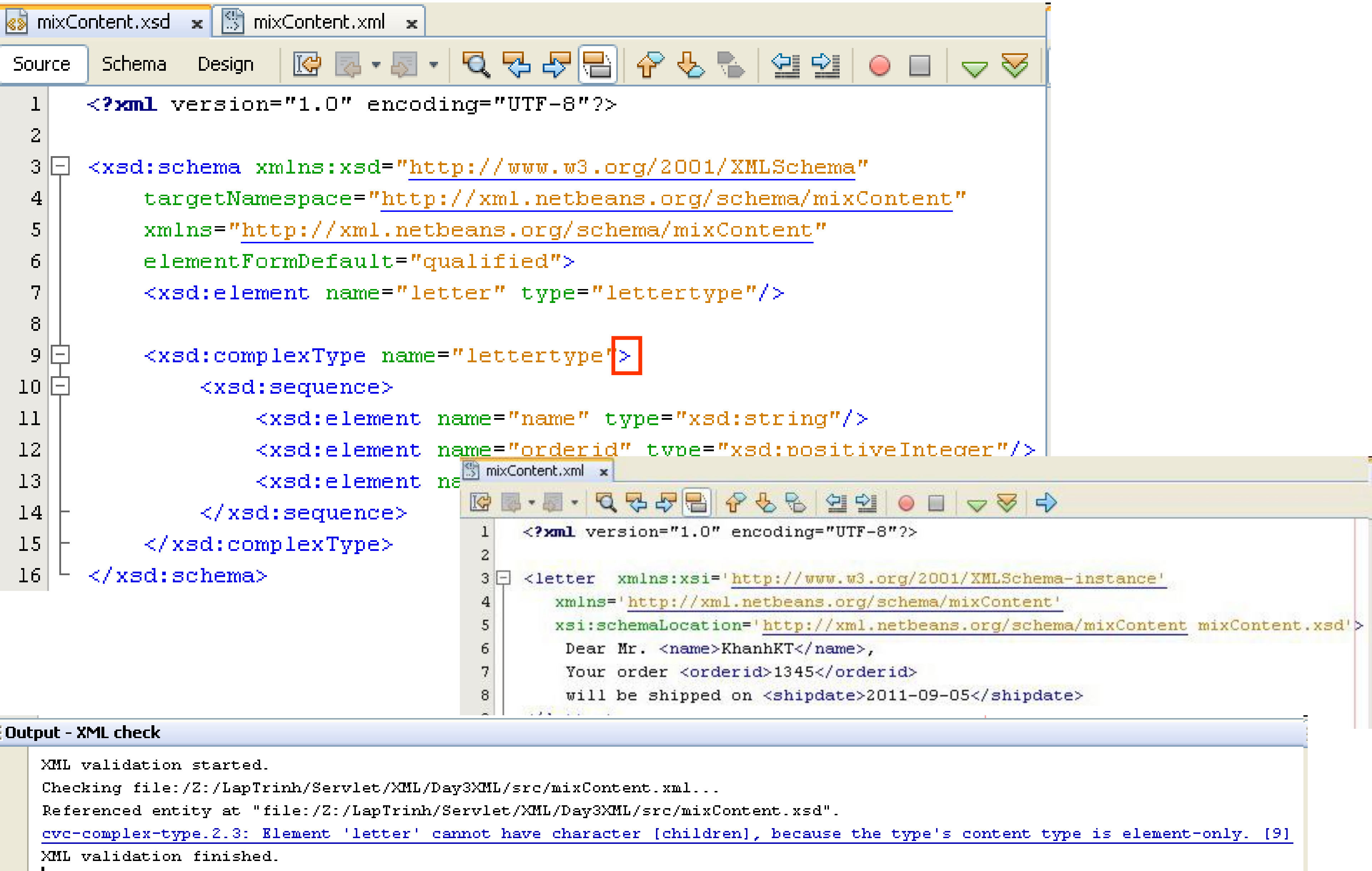
```
<?xml version="1.0" encoding="UTF-8"?>

<letter  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/mixContent'
   xsi:schemaLocation='http://xml.netbeans.org/schema/mixContent mixContent.xsd'>
    Dear Mr. <name>KhanhKT</name>,
    Your order <orderid>1345</orderid>
    will be shipped on <shipdate>2011-09-05</shipdate>
</letter>
```



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/mixContent"
    xmlns="http://xml.netbeans.org/schema/mixContent"
    elementFormDefault="qualified">
    <xsd:element name="letter" type="lettertype"/>

    <xsd:complexType name="lettertype" mixed="true">
        <xsd:sequence>
            <xsd:element name="name" type="xsd:string"/>
            <xsd:element name="orderid" type="xsd:positiveInteger"/>
            <xsd:element name="shipdate" type="xsd:date"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

Schemas

Structuring – Mixed Content – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/mixContent"
    xmlns="http://xml.netbeans.org/schema/mixContent"
    elementFormDefault="qualified">
    <xsd:element name="letter" type="lettertype"/>

    <xsd:complexType name="lettertype">
        <xsd:sequence>
            <xsd:element name="name" type="xsd:string"/>
            <xsd:element name="orderid" type="xsd:positiveInteger"/>
            <xsd:element na
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

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

<letter  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/mixContent'
   xsi:schemaLocation='http://xml.netbeans.org/schema/mixContent mixContent.xsd'>
    Dear Mr. <name>KhanhKT</name>,
    Your order <orderid>1345</orderid>
    will be shipped on <shipdate>2011-09-05</shipdate>
```

Output - XML check

```
XML validation started.
Checking file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mixContent.xml...
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/mixContent.xsd".
cvc-complex-type.2.3: Element 'letter' cannot have character [children], because the type's content type is element-only. [9]
XML validation finished.
```

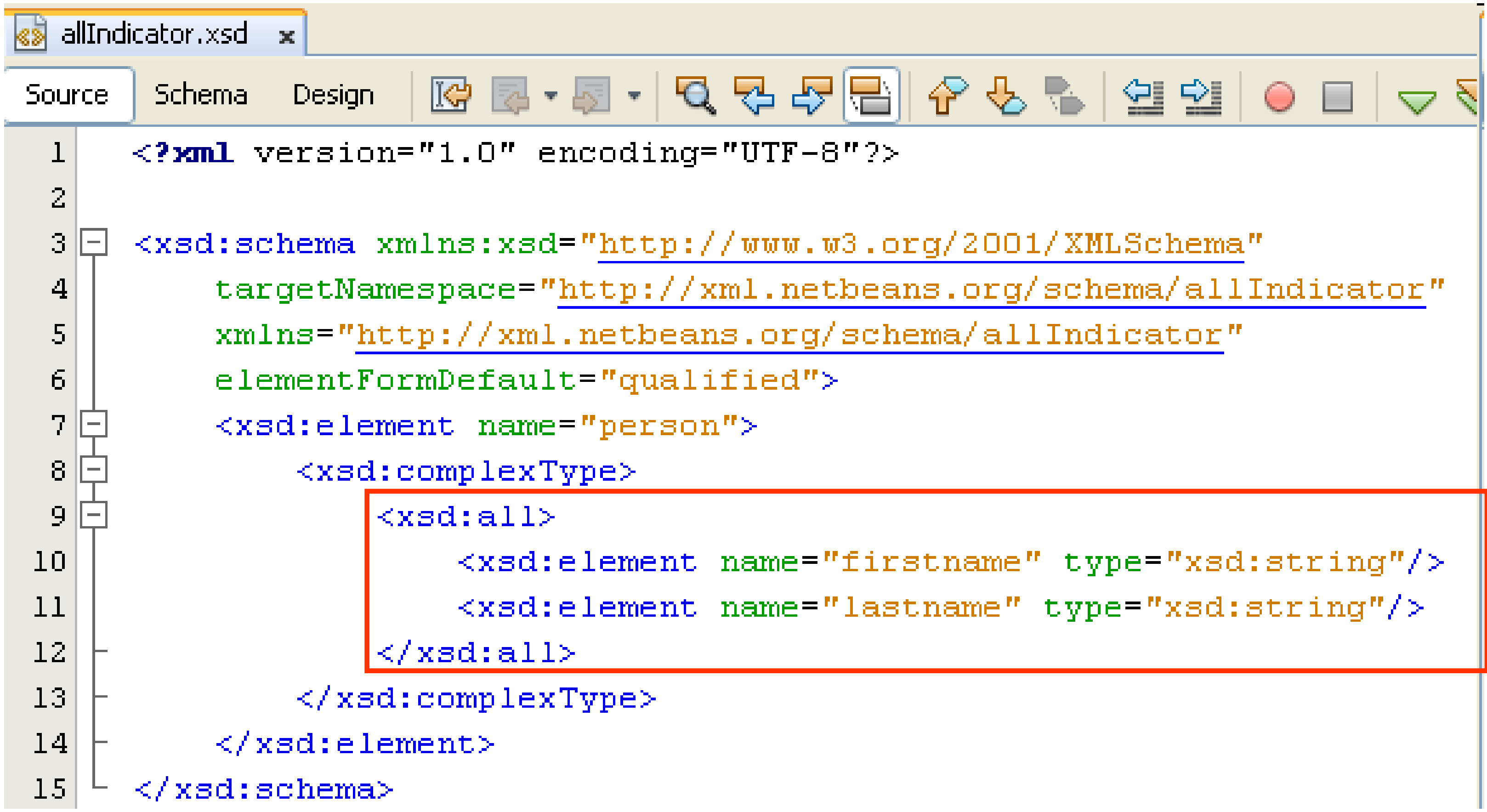
Schemas

Structuring – XSD Indicators

- **Control how elements** are to be **used** in documents
- **Indicate** the **structure** and **order** in which **child elements** can appear within their parent element
- There are seven indicators:
  - **Order** indicators:
    - All
    - Choice
    - Sequence
  - **Occurrence** indicators:
    - maxOccurs
    - minOccurs
  - **Group** indicators:
    - Group name
    - attributeGroup name

Schemas
Structuring – Order Indicators

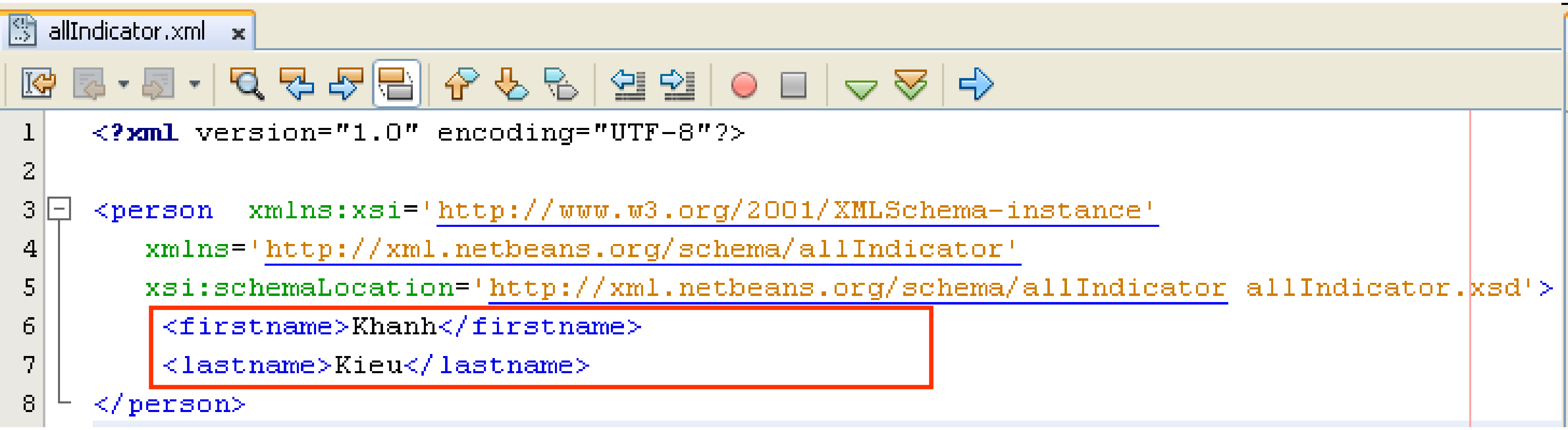
- Are used to define the **order of the elements**
- **All** indicator
  - **Specifies** that the child elements can appear in **any order**, and that **each** child element **must occur only once**
  - **Ex:**



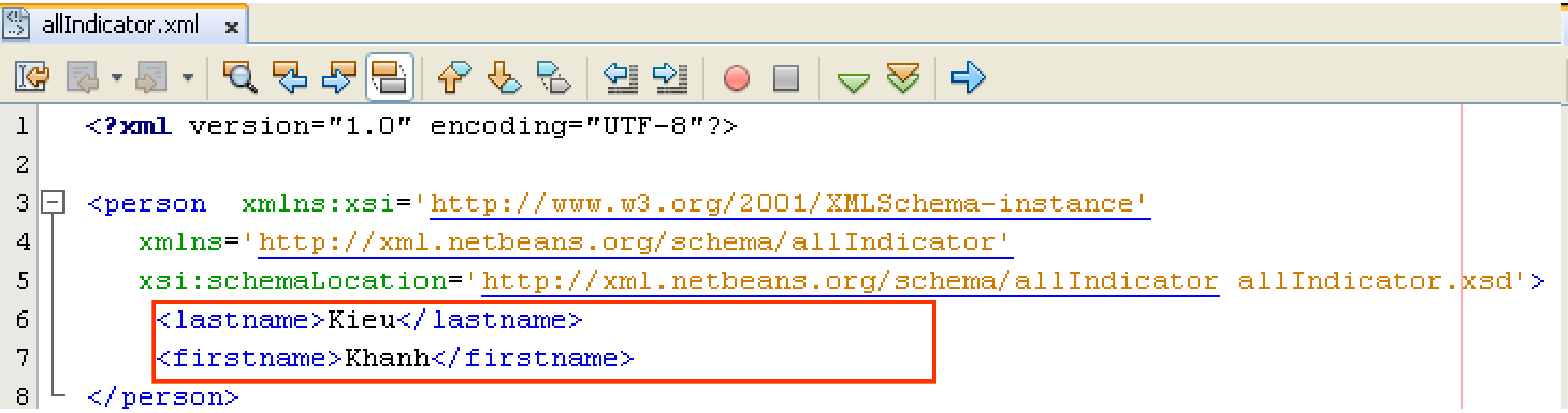
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/allIndicator"
    xmlns="http://xml.netbeans.org/schema/allIndicator"
    elementFormDefault="qualified">
    <xsd:element name="person">
        <xsd:complexType>
            <xsd:all>
                <xsd:element name="firstname" type="xsd:string"/>
                <xsd:element name="lastname" type="xsd:string"/>
            </xsd:all>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas

Structuring – Order Indicators – Example



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

<person  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/allIndicator'
   xsi:schemaLocation='http://xml.netbeans.org/schema/allIndicator allIndicator.xsd'>
    <firstname>Khanh</firstname>
    <lastname>Kieu</lastname>
</person>
```

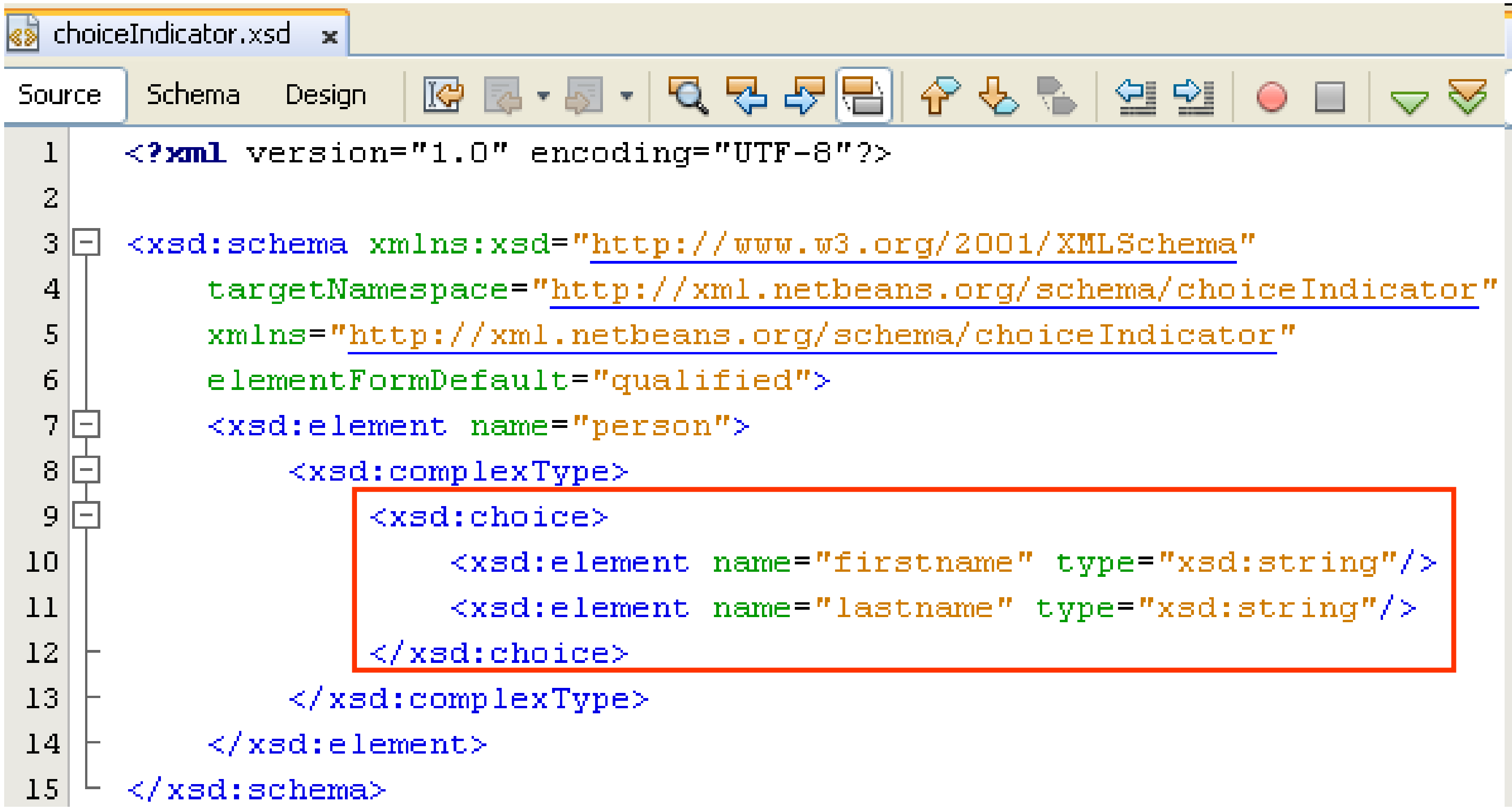


```
<?xml version="1.0" encoding="UTF-8"?>

<person  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/allIndicator'
   xsi:schemaLocation='http://xml.netbeans.org/schema/allIndicator allIndicator.xsd'>
    <lastname>Kieu</lastname>
    <firstname>Khanh</firstname>
</person>
```

Schemas

Structuring – Order Indicators

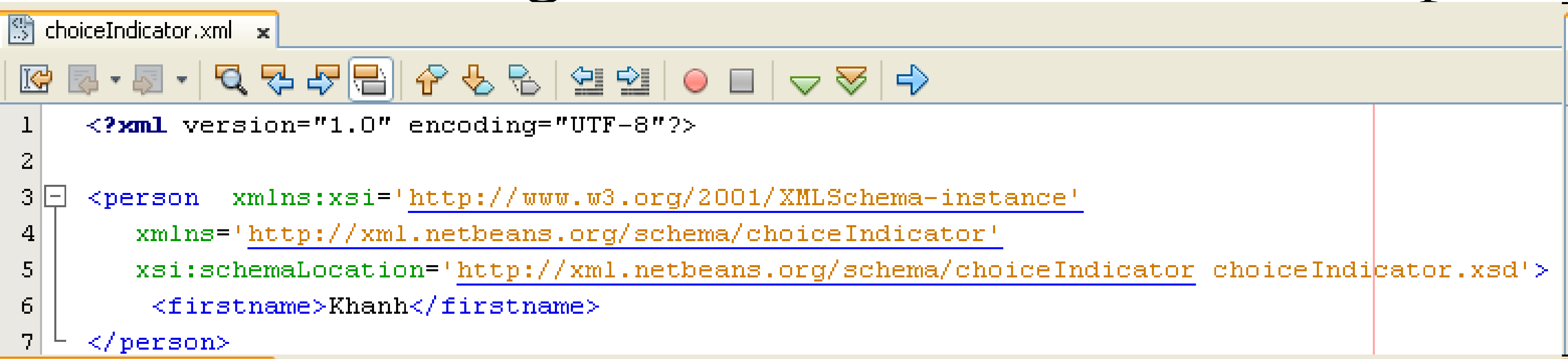
- **Choice** indicator
  - Specifies that **either one** child element **or another** can **occur** (meaning that **only one** of the child elements **may show up**)
  - Ex:



choiceIndicator.xsd

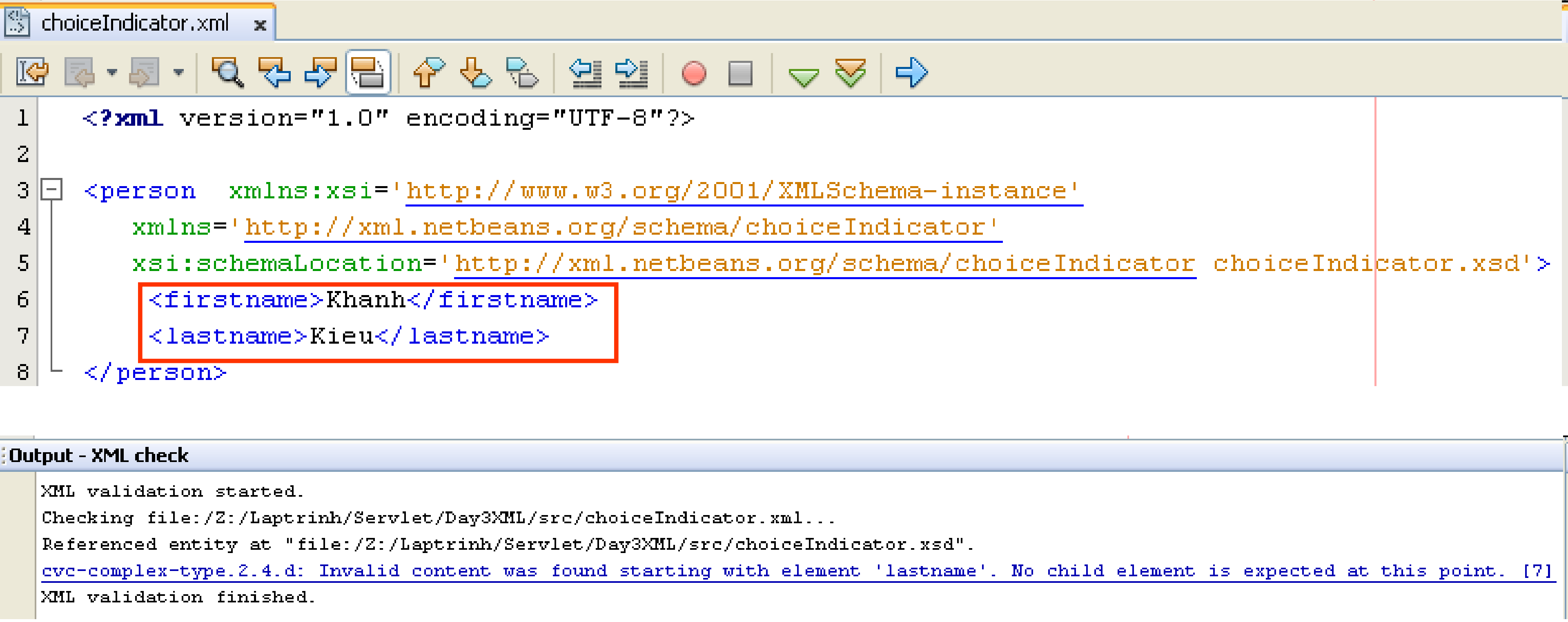
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/choiceIndicator"
    xmlns="http://xml.netbeans.org/schema/choiceIndicator"
    elementFormDefault="qualified">
    <xsd:element name="person">
        <xsd:complexType>
            <xsd:choice>
                <xsd:element name="firstname" type="xsd:string"/>
                <xsd:element name="lastname" type="xsd:string"/>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas
Structuring – Order Indicators – Example



choiceIndicator.xml

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

<person  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/choiceIndicator'
    xsi:schemaLocation='http://xml.netbeans.org/schema/choiceIndicator choiceIndicator.xsd'>
     <firstname>Khanh</firstname>
</person>
```



choiceIndicator.xml

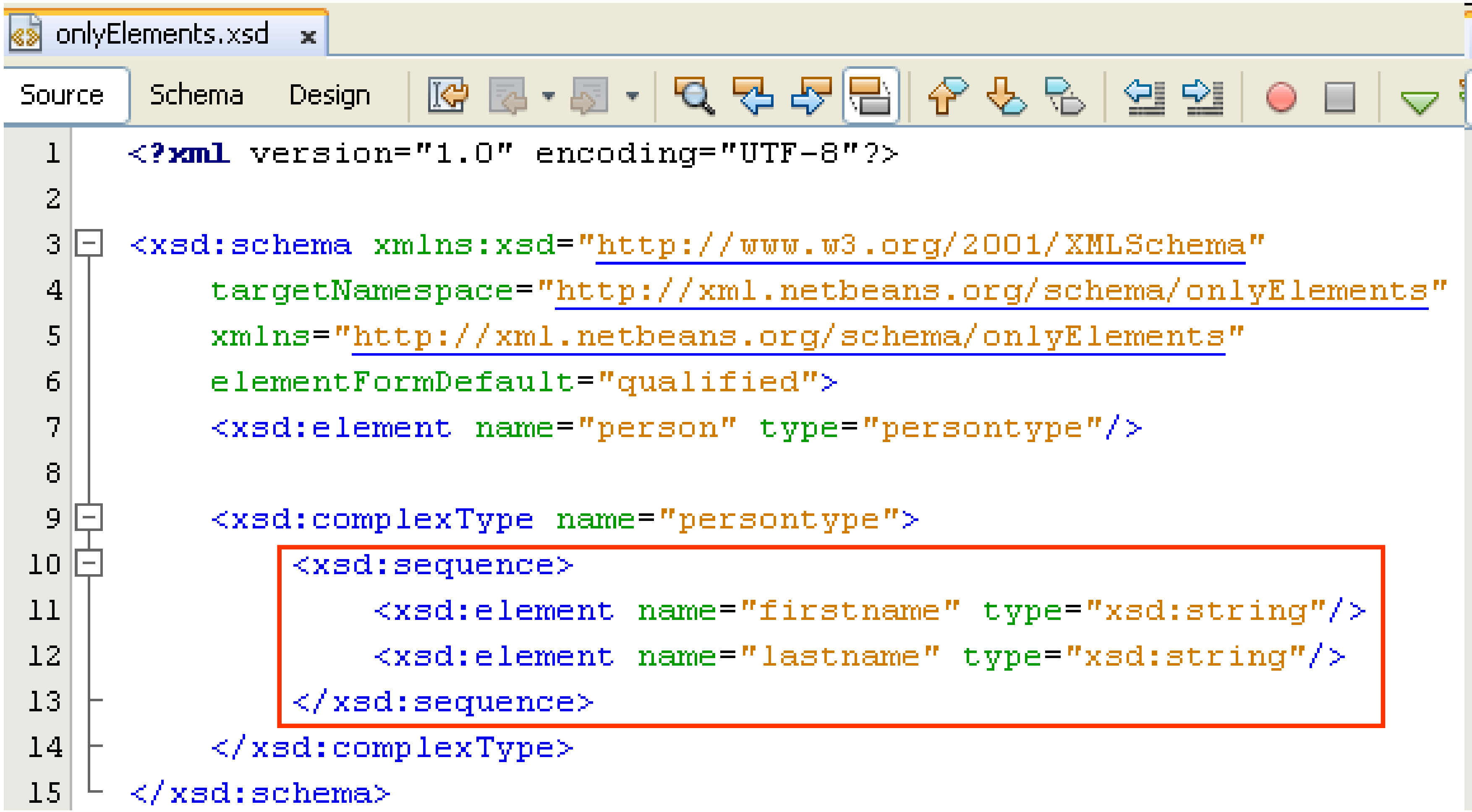
```xml
<?xml version="1.0" encoding="UTF-8"?>

<person  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/choiceIndicator'
    xsi:schemaLocation='http://xml.netbeans.org/schema/choiceIndicator choiceIndicator.xsd'>
     <firstname>Khanh</firstname>
     <lastname>Kieu</lastname>
</person>
```

Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/choiceIndicator.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/choiceIndicator.xsd".
cvc-complex-type.2.4.d: Invalid content was found starting with element 'lastname'. No child element is expected at this point. [7]
XML validation finished.
```

Schemas
Structuring – Order Indicators

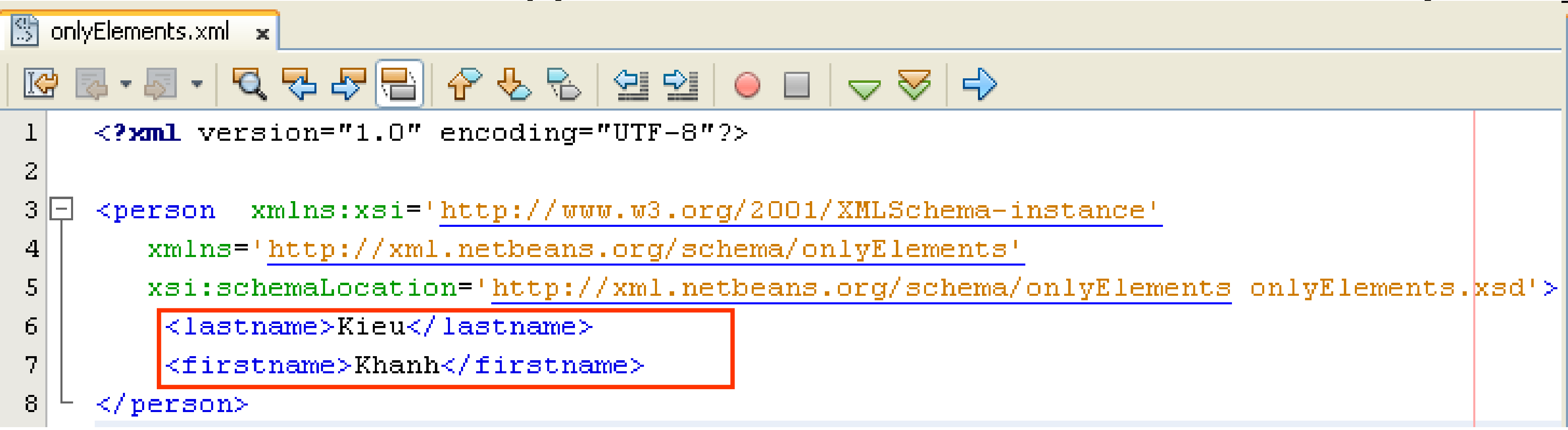
- Sequence indicator
  - **Specifies** that the **child** elements must **appear in** a specific **order**
  - **Ex:**



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/onlyElements"
    xmlns="http://xml.netbeans.org/schema/onlyElements"
    elementFormDefault="qualified">
    <xsd:element name="person" type="persontype"/>

    <xsd:complexType name="persontype">
        <xsd:sequence>
            <xsd:element name="firstname" type="xsd:string"/>
            <xsd:element name="lastname" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

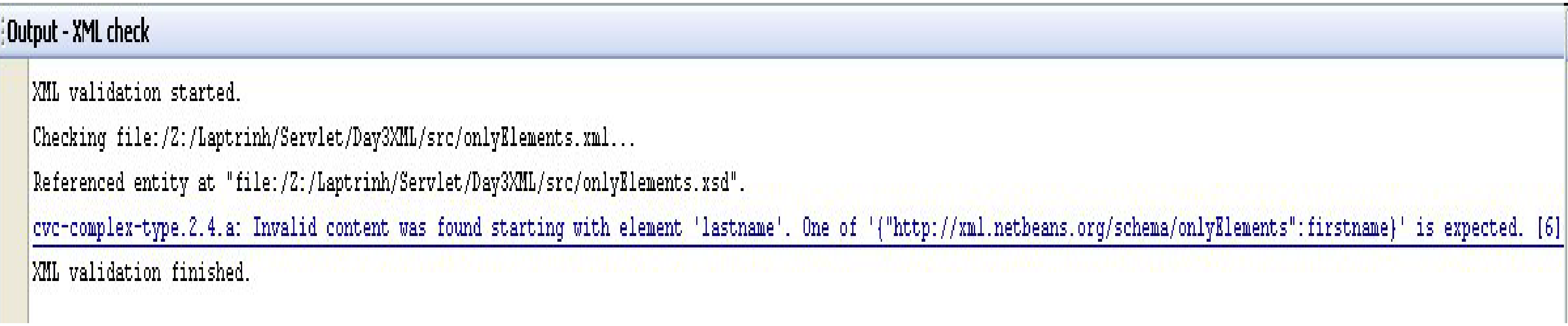
Schemas

Structuring – Order Indicators – Example



onlyElements.xml

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

<person  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/onlyElements'
    xsi:schemaLocation='http://xml.netbeans.org/schema/onlyElements onlyElements.xsd'>
     <lastname>Kieu</lastname>
     <firstname>Khanh</firstname>
</person>
```

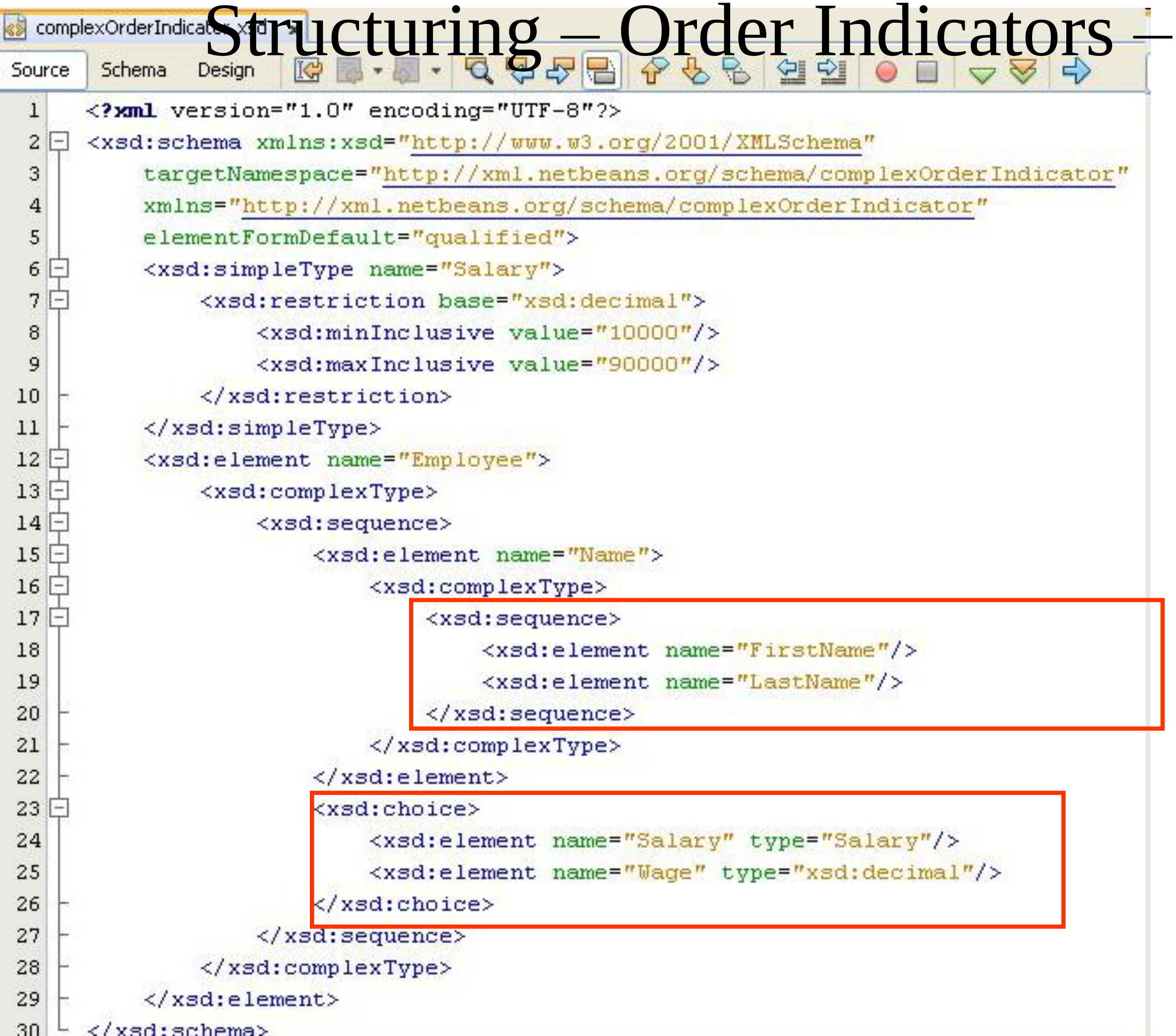


Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/onlyElements.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/onlyElements.xsd".
cvc-complex-type.2.4.a: Invalid content was found starting with element 'lastname'. One of '{"http://xml.netbeans.org/schema/onlyElements":firstname}' is expected. [6]
XML validation finished.
```

Schemas

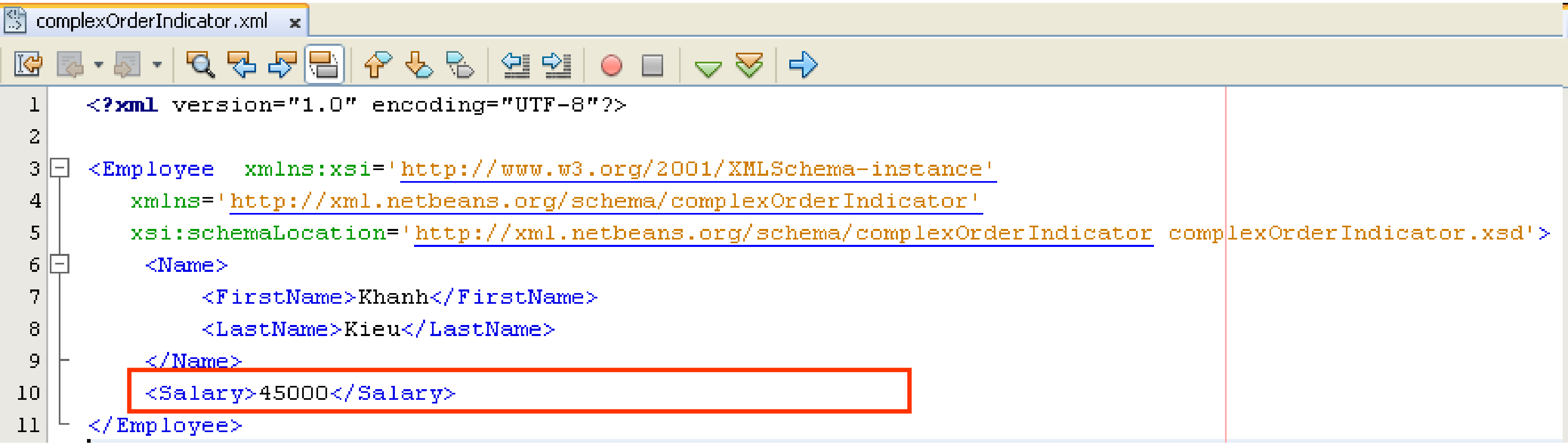
Structuring – Order Indicators – Example



```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/complexOrderIndicator"
    xmlns="http://xml.netbeans.org/schema/complexOrderIndicator"
    elementFormDefault="qualified">
    <xsd:simpleType name="Salary">
        <xsd:restriction base="xsd:decimal">
            <xsd:minInclusive value="10000"/>
            <xsd:maxInclusive value="90000"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="Employee">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Name">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="FirstName"/>
                            <xsd:element name="LastName"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
                <xsd:choice>
                    <xsd:element name="Salary" type="Salary"/>
                    <xsd:element name="Wage" type="xsd:decimal"/>
                </xsd:choice>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

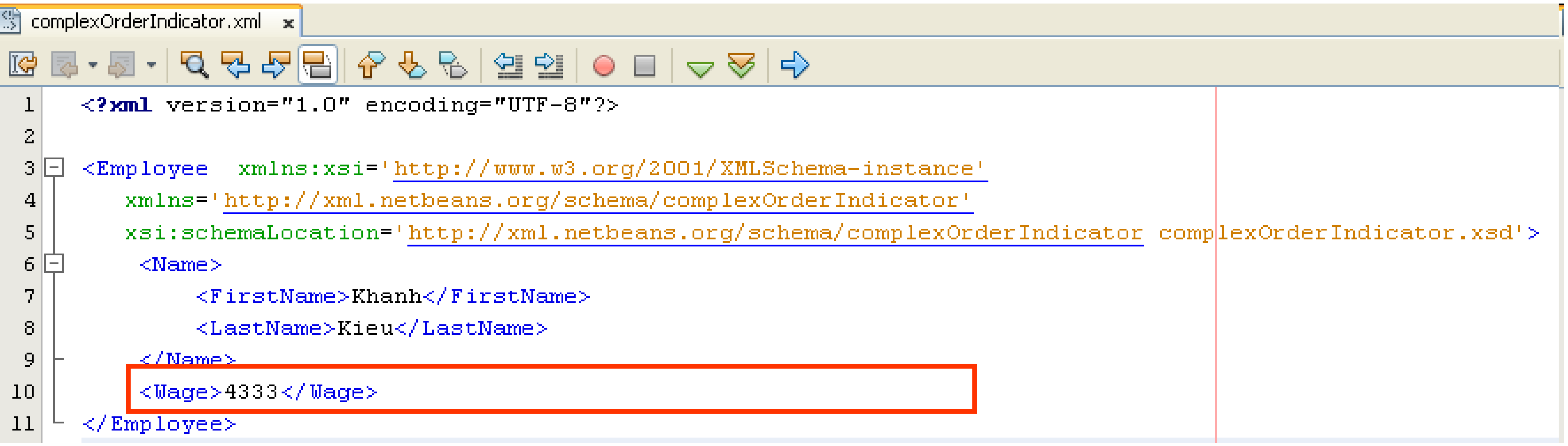
Schemas

Structuring – Order Indicators – Example



complexOrderIndicator.xml

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

<Employee  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/complexOrderIndicator'
   xsi:schemaLocation='http://xml.netbeans.org/schema/complexOrderIndicator complexOrderIndicator.xsd'>
    <Name>
        <FirstName>Khanh</FirstName>
        <LastName>Kieu</LastName>
    </Name>
    <Salary>45000</Salary>
</Employee>
```



complexOrderIndicator.xml

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

<Employee  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/complexOrderIndicator'
   xsi:schemaLocation='http://xml.netbeans.org/schema/complexOrderIndicator complexOrderIndicator.xsd'>
    <Name>
        <FirstName>Khanh</FirstName>
        <LastName>Kieu</LastName>
    </Name>
    <Wage>4333</Wage>
</Employee>
```

Schemas
Structuring – Occurrence Indicators

- Are used to **define how often** an element can **occur**
- Are same ideas as the DTD that uses the markers *, ?, and + to indicate the number of time a particular child element
- The **default value** of indicators for **maxOccurs** and **minOccurs** is **1**
- **maxOccurs** indicator
  - **Specifies** the **maximum number of times** an element can occur
- **minOccurs** indicator
  - Specifies the **minimum number of times** an element can occur

Schemas

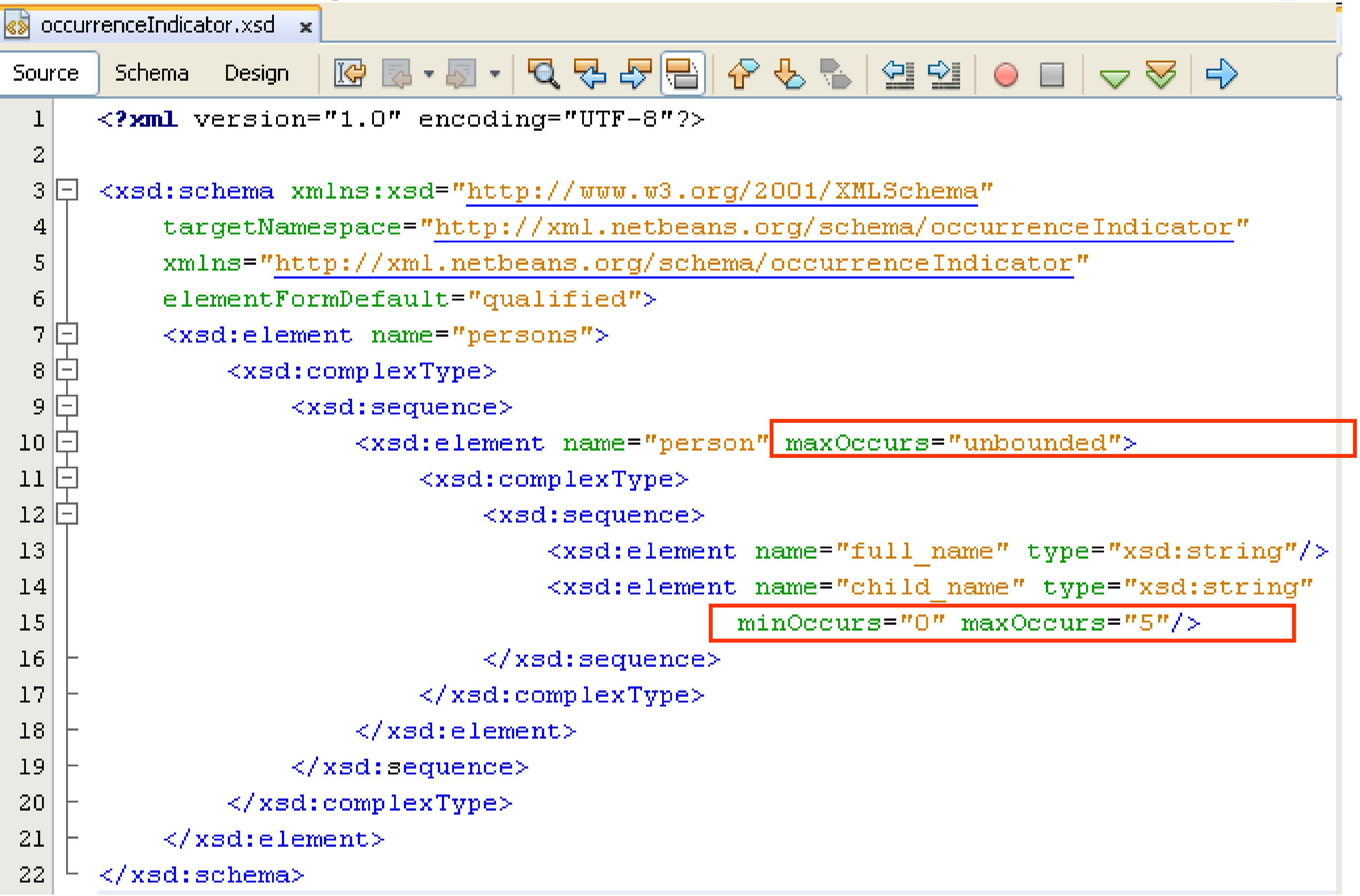
Structuring – Occurrence Indicators

The relationship between minOccurs and maxOccurs attributes

minOccurs	**maxOccurs**	**Number of times an element can occur**
0	1	0 or 1
1	1	1
0	unbounded	Infinite – at least zero
1	unbounded	At least one
>0	unbounded	At least minOccurs times
>maxOccurs	Any value	0 – **error**
Any value	<minOccurs	0 – **error**

Schemas

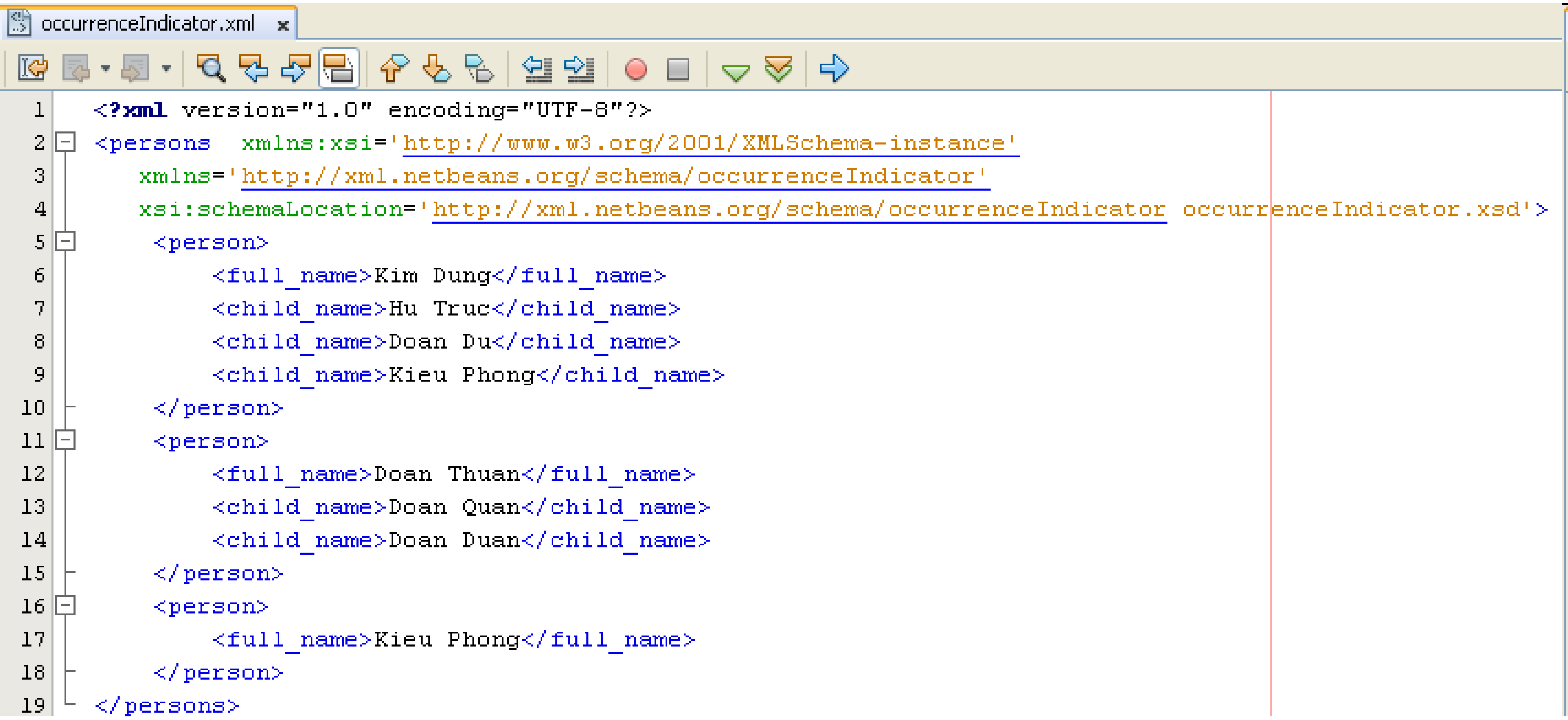
Structuring – Occurrence Indicators – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/occurrenceIndicator"
    xmlns="http://xml.netbeans.org/schema/occurrenceIndicator"
    elementFormDefault="qualified">
    <xsd:element name="persons">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="person" maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="full_name" type="xsd:string"/>
                            <xsd:element name="child_name" type="xsd:string"
                                minOccurs="0" maxOccurs="5"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas

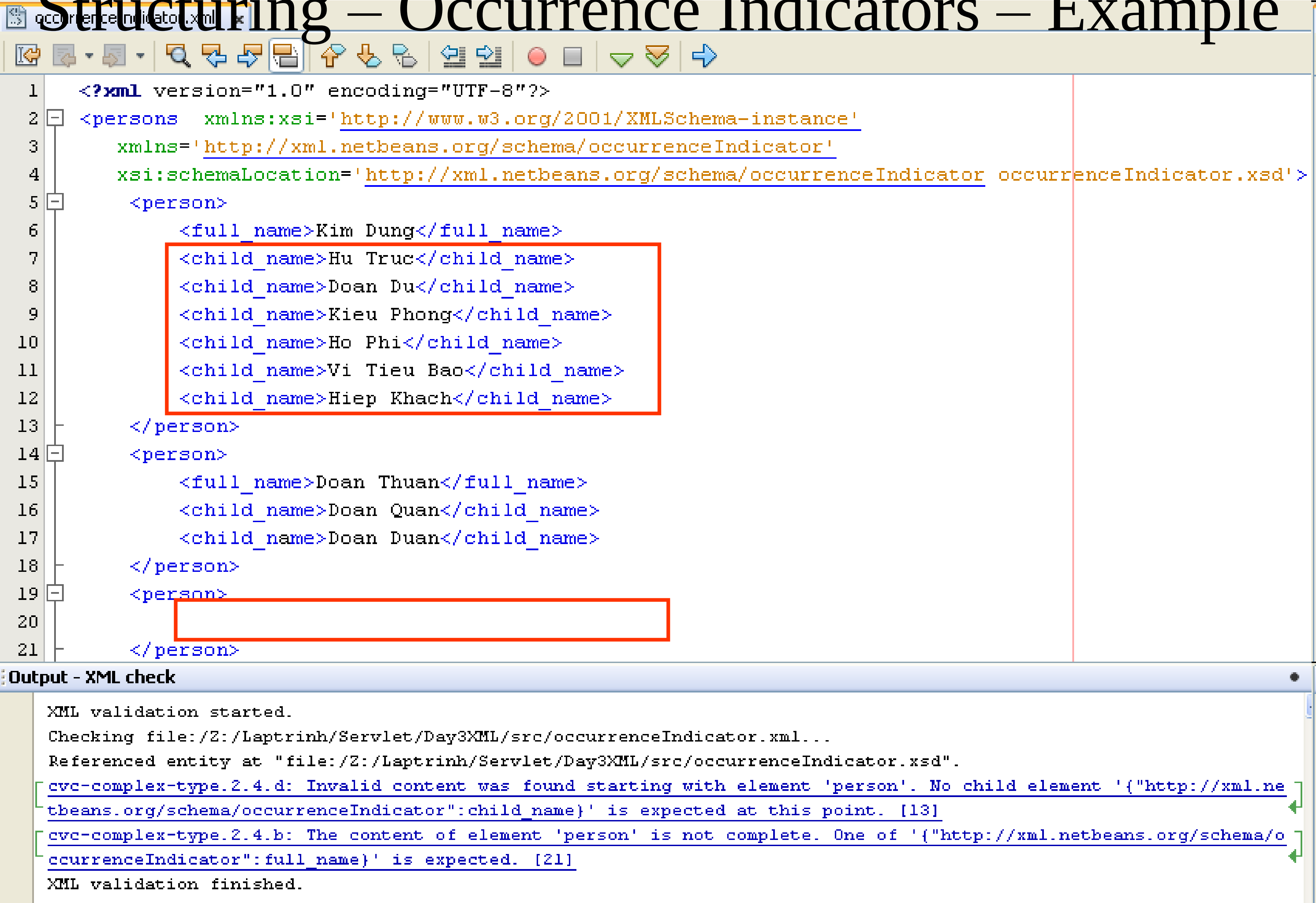
Structuring – Occurrence Indicators – Example



occurrenceIndicator.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persons  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/occurrenceIndicator'
   xsi:schemaLocation='http://xml.netbeans.org/schema/occurrenceIndicator occurrenceIndicator.xsd'>
    <person>
        <full_name>Kim Dung</full_name>
        <child_name>Hu Truc</child_name>
        <child_name>Doan Du</child_name>
        <child_name>Kieu Phong</child_name>
    </person>
    <person>
        <full_name>Doan Thuan</full_name>
        <child_name>Doan Quan</child_name>
        <child_name>Doan Duan</child_name>
    </person>
    <person>
        <full_name>Kieu Phong</full_name>
    </person>
</persons>
```

Schemas
Structuring – Occurrence Indicators – Example

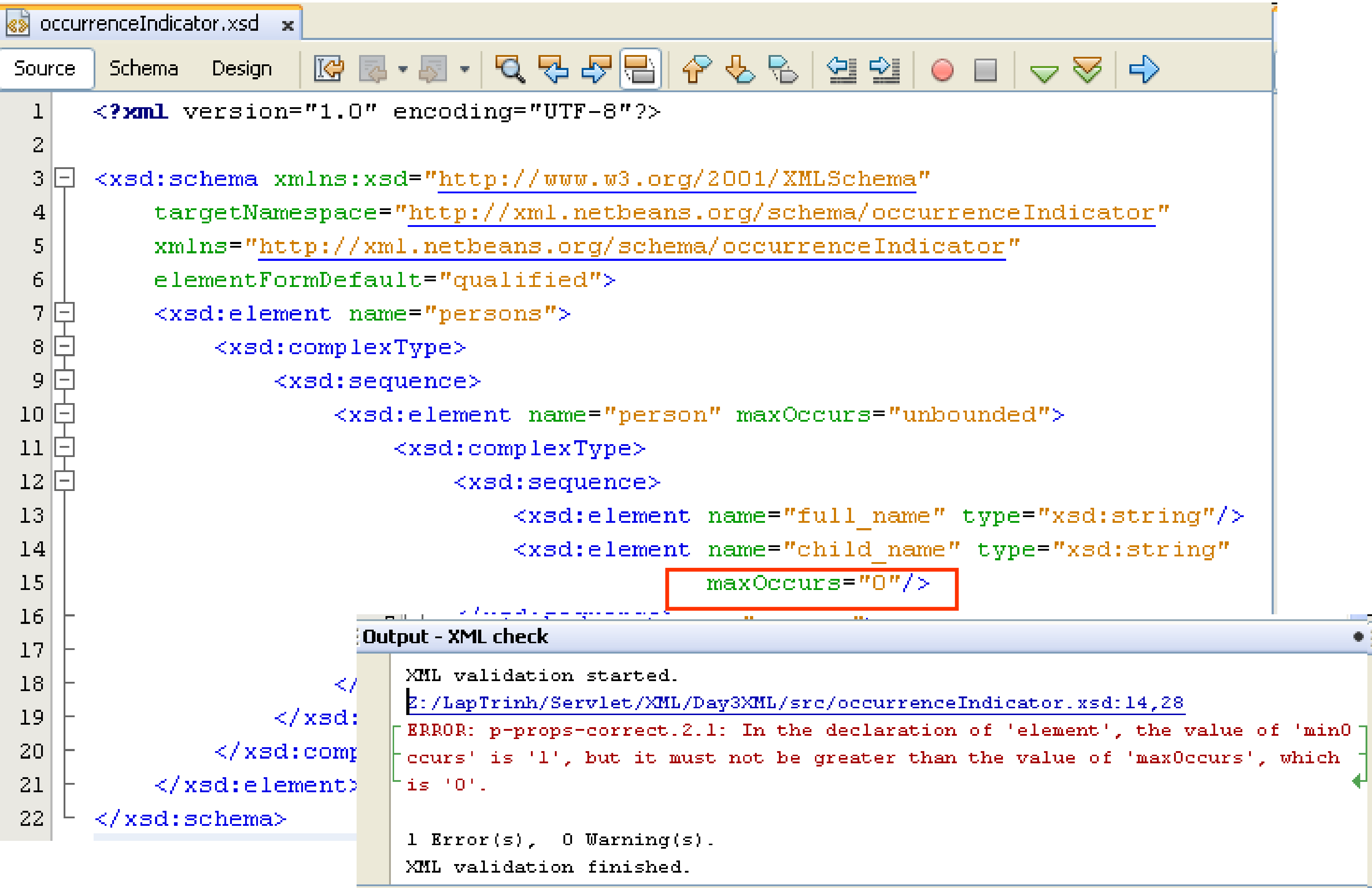


```
<?xml version="1.0" encoding="UTF-8"?>
<persons  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/occurrenceIndicator'
   xsi:schemaLocation='http://xml.netbeans.org/schema/occurrenceIndicator occurrenceIndicator.xsd'>
    <person>
        <full_name>Kim Dung</full_name>
        <child_name>Hu Truc</child_name>
        <child_name>Doan Du</child_name>
        <child_name>Kieu Phong</child_name>
        <child_name>Ho Phi</child_name>
        <child_name>Vi Tieu Bao</child_name>
        <child_name>Hiep Khach</child_name>
    </person>
    <person>
        <full_name>Doan Thuan</full_name>
        <child_name>Doan Quan</child_name>
        <child_name>Doan Duan</child_name>
    </person>
    <person>

    </person>
```

Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/occurrenceIndicator.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/occurrenceIndicator.xsd".
cvc-complex-type.2.4.d: Invalid content was found starting with element 'person'. No child element '{"http://xml.ne
tbeans.org/schema/occurrenceIndicator":child_name}' is expected at this point. [13]
cvc-complex-type.2.4.b: The content of element 'person' is not complete. One of '{"http://xml.netbeans.org/schema/o
ccurrenceIndicator":full_name}' is expected. [21]
XML validation finished.
```
]

Schemas
Structuring – Occurrence Indicators – Example

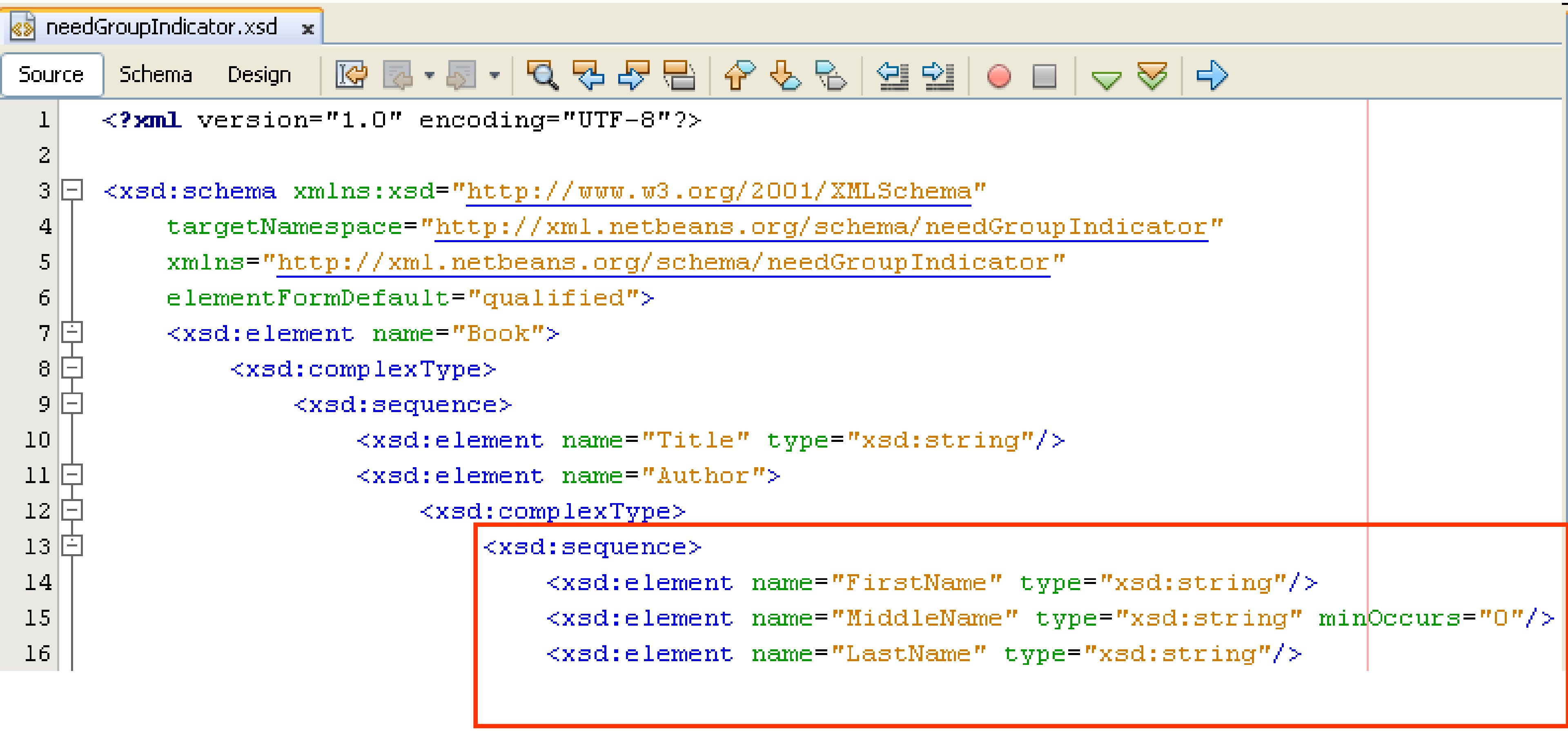


```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/occurrenceIndicator"
    xmlns="http://xml.netbeans.org/schema/occurrenceIndicator"
    elementFormDefault="qualified">
    <xsd:element name="persons">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="person" maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="full_name" type="xsd:string"/>
                            <xsd:element name="child_name" type="xsd:string"
                                    maxOccurs="0"/>

            </
        </xsd:
    </xsd:comp
    </xsd:element>
</xsd:schema>
```

Output - XML check

```
XML validation started.
Z:/LapTrinh/Servlet/XML/Day3XML/src/occurrenceIndicator.xsd:14,28
ERROR: p-props-correct.2.1: In the declaration of 'element', the value of 'minO
ccurs' is '1', but it must not be greater than the value of 'maxOccurs', which
is '0'.

1 Error(s),  0 Warning(s).
XML validation finished.
```

Schemas
Structuring – Need for Group Indicators

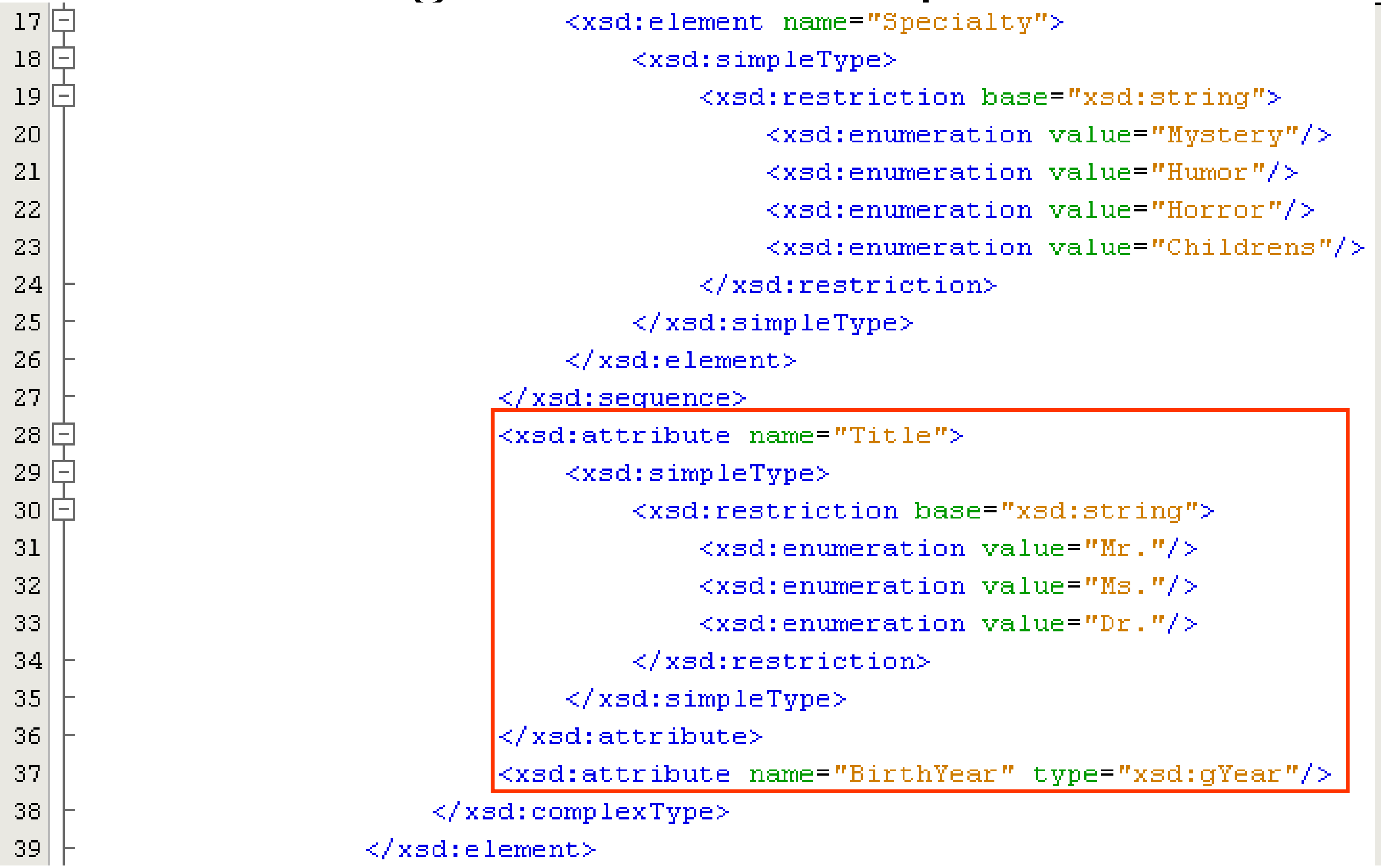


needGroupIndicator.xsd

Source Schema Design

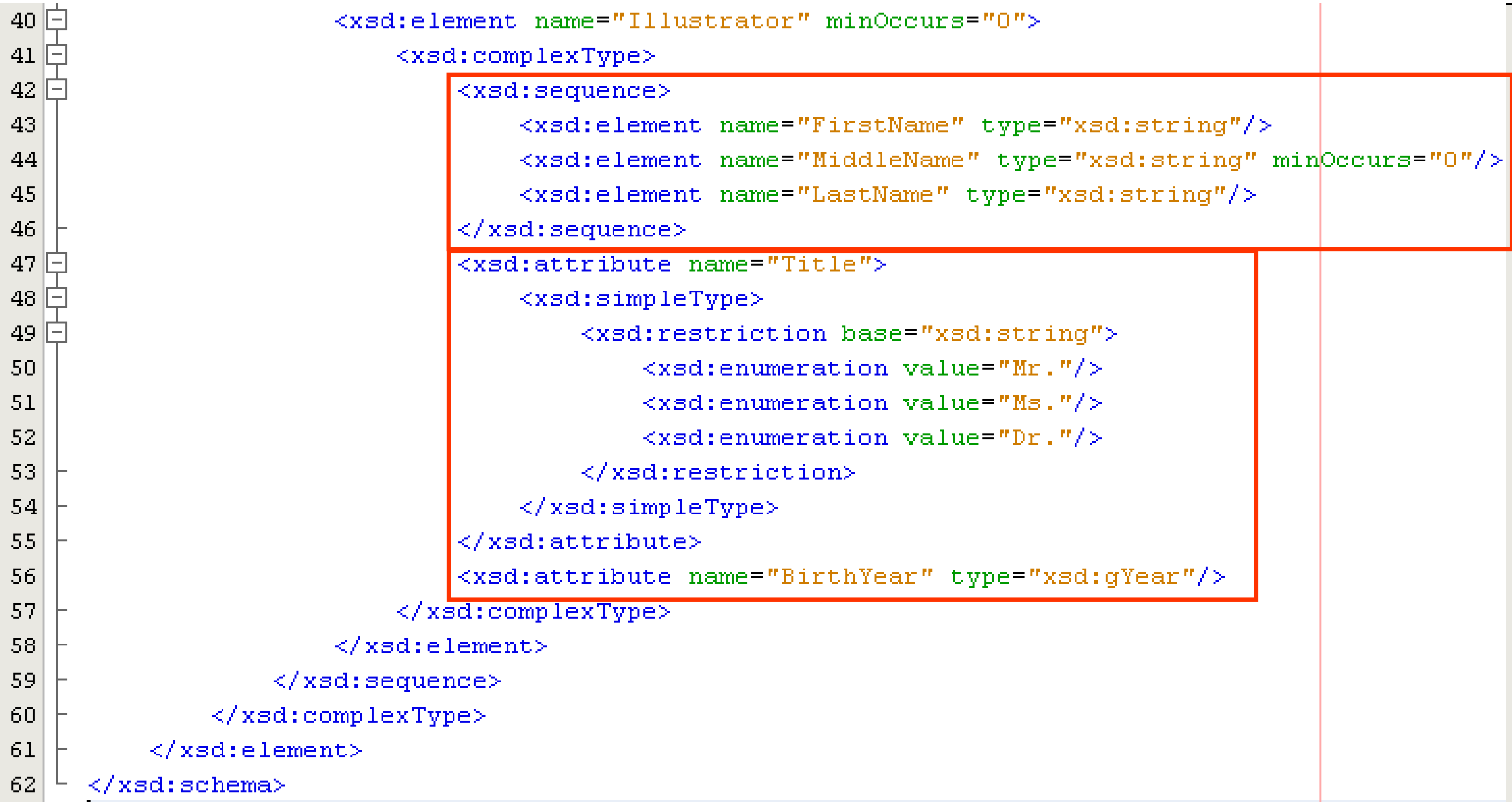
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/needGroupIndicator"
    xmlns="http://xml.netbeans.org/schema/needGroupIndicator"
    elementFormDefault="qualified">
    <xsd:element name="Book">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Title" type="xsd:string"/>
                <xsd:element name="Author">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="FirstName" type="xsd:string"/>
                            <xsd:element name="MiddleName" type="xsd:string" minOccurs="0"/>
                            <xsd:element name="LastName" type="xsd:string"/>
```

Schemas
Structuring – Need for Group Indicators



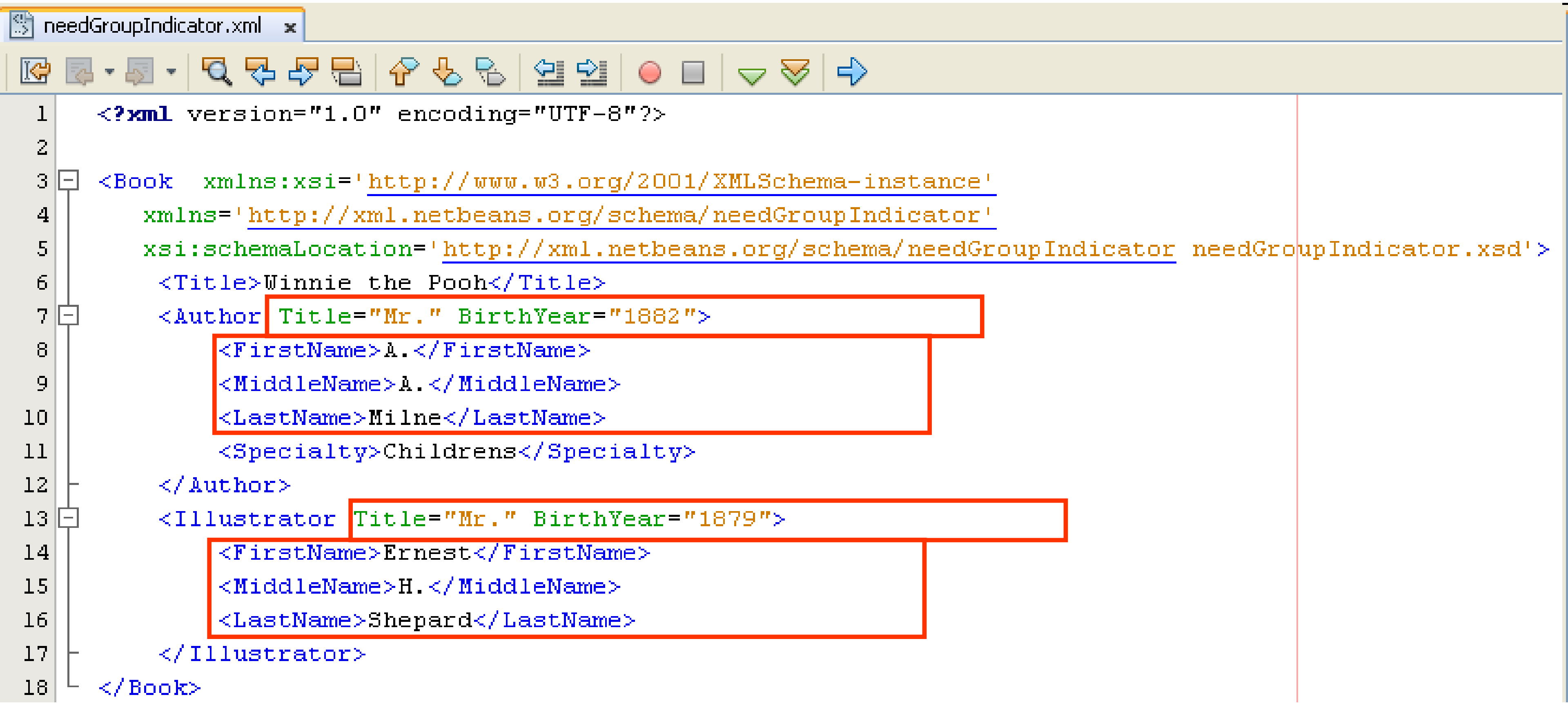
```
                    <xsd:element name="Specialty">
                        <xsd:simpleType>
                            <xsd:restriction base="xsd:string">
                                <xsd:enumeration value="Mystery"/>
                                <xsd:enumeration value="Humor"/>
                                <xsd:enumeration value="Horror"/>
                                <xsd:enumeration value="Childrens"/>
                            </xsd:restriction>
                        </xsd:simpleType>
                    </xsd:element>
                </xsd:sequence>
                <xsd:attribute name="Title">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="Mr."/>
                            <xsd:enumeration value="Ms."/>
                            <xsd:enumeration value="Dr."/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
                <xsd:attribute name="BirthYear" type="xsd:gYear"/>
            </xsd:complexType>
        </xsd:element>
```

Schemas
Structuring – Need for Group Indicators



```
                    <xsd:element name="Illustrator" minOccurs="0">
                        <xsd:complexType>
                            <xsd:sequence>
                                <xsd:element name="FirstName" type="xsd:string"/>
                                <xsd:element name="MiddleName" type="xsd:string" minOccurs="0"/>
                                <xsd:element name="LastName" type="xsd:string"/>
                            </xsd:sequence>
                            <xsd:attribute name="Title">
                                <xsd:simpleType>
                                    <xsd:restriction base="xsd:string">
                                        <xsd:enumeration value="Mr."/>
                                        <xsd:enumeration value="Ms."/>
                                        <xsd:enumeration value="Dr."/>
                                    </xsd:restriction>
                                </xsd:simpleType>
                            </xsd:attribute>
                            <xsd:attribute name="BirthYear" type="xsd:gYear"/>
                        </xsd:complexType>
                    </xsd:element>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
</xsd:schema>
```

Schemas
Structuring – Need for Group Indicators



needGroupIndicator.xml

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

<Book  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/needGroupIndicator'
   xsi:schemaLocation='http://xml.netbeans.org/schema/needGroupIndicator needGroupIndicator.xsd'>
    <Title>Winnie the Pooh</Title>
    <Author Title="Mr." BirthYear="1882">
        <FirstName>A.</FirstName>
        <MiddleName>A.</MiddleName>
        <LastName>Milne</LastName>
        <Specialty>Childrens</Specialty>
    </Author>
    <Illustrator Title="Mr." BirthYear="1879">
        <FirstName>Ernest</FirstName>
        <MiddleName>H.</MiddleName>
        <LastName>Shepard</LastName>
    </Illustrator>
</Book>
```

Schemas
Structuring – Group Indicators

- Are used to **define related sets of elements and/or attributes**
- Can be used to create a set structure **for reuse**
- **Syntax**
  - **Element** groups

    <xsd:group name="group_name">

    ...

    < /xsd:group>

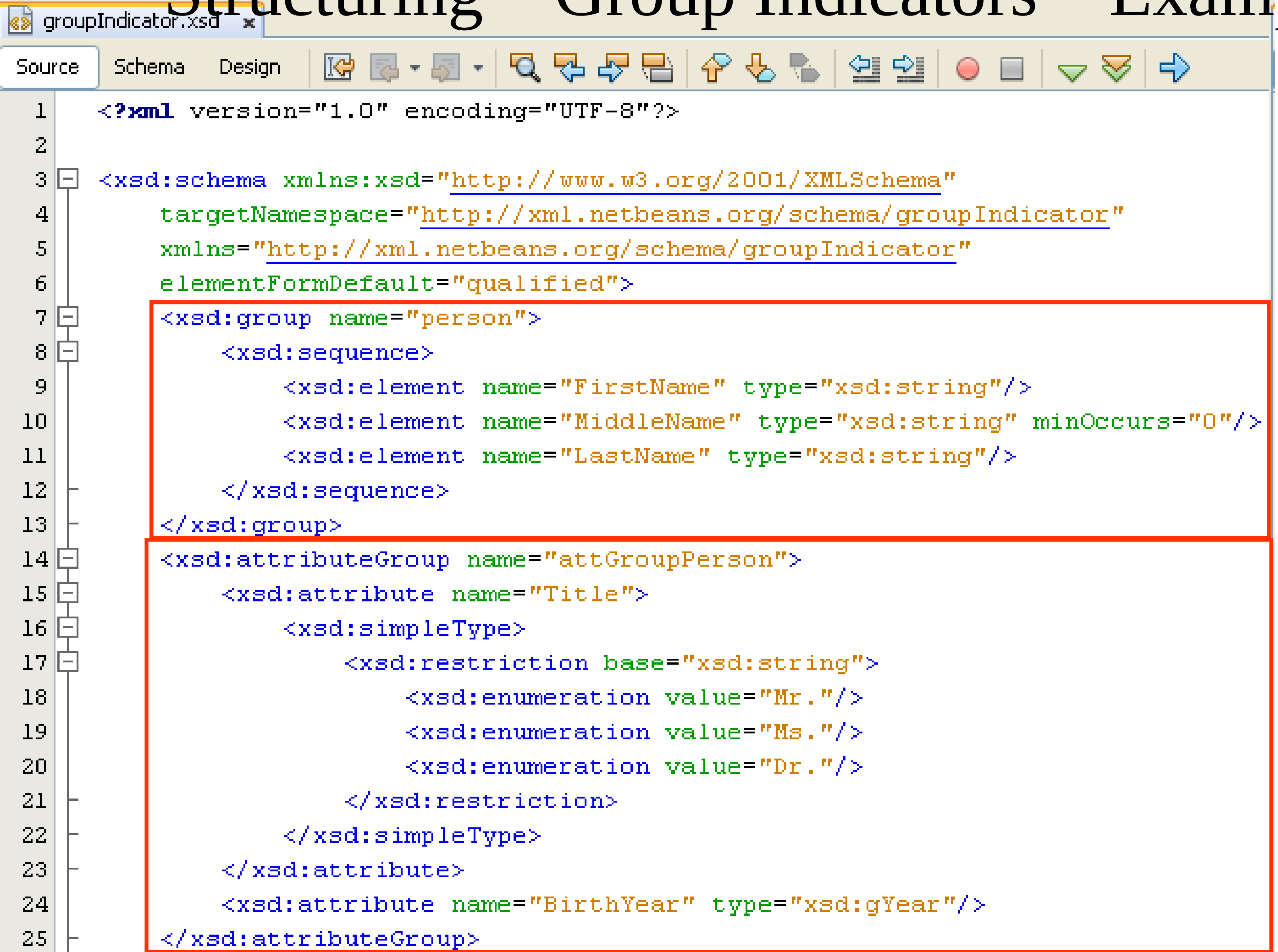
    - An **all, choice, or sequence element** must be **defined inside** the group declaration
  - **Attribute** groups

    <xsd:attributeGroup name="group_name">

    ...

    < /xsd:attributeGroup>
- **After** the group have been **defined**, **it** can **be referenced** in **another definition**
  - **Syntax**: **<xsd:group ref=“group_name”/> or <xsd:attributeGroup ref=“”/>**

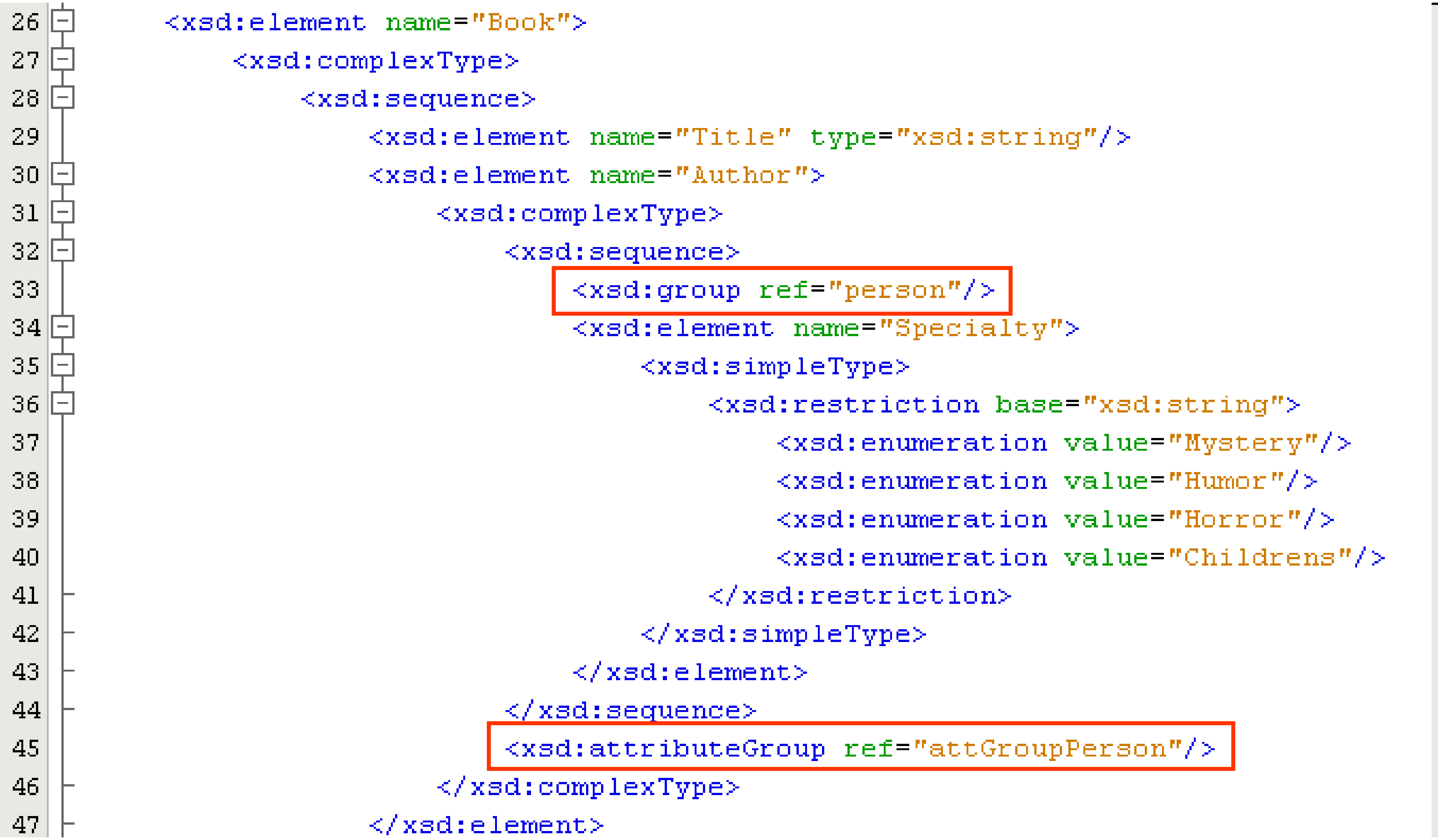
Schemas
Structuring – Group Indicators – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/groupIndicator"
    xmlns="http://xml.netbeans.org/schema/groupIndicator"
    elementFormDefault="qualified">
    <xsd:group name="person">
        <xsd:sequence>
            <xsd:element name="FirstName" type="xsd:string"/>
            <xsd:element name="MiddleName" type="xsd:string" minOccurs="0"/>
            <xsd:element name="LastName" type="xsd:string"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:attributeGroup name="attGroupPerson">
        <xsd:attribute name="Title">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Mr."/>
                    <xsd:enumeration value="Ms."/>
                    <xsd:enumeration value="Dr."/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:attribute>
        <xsd:attribute name="BirthYear" type="xsd:gYear"/>
    </xsd:attributeGroup>
```

Schemas

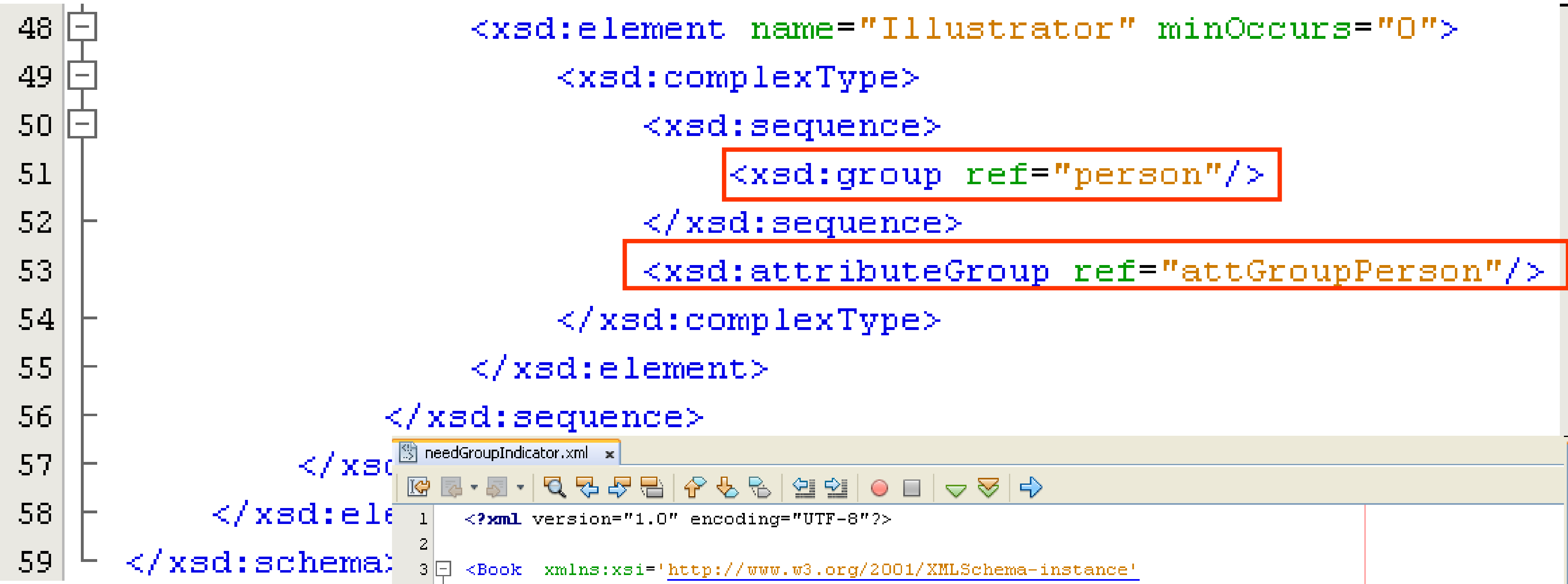
Structuring – Group Indicators – Example



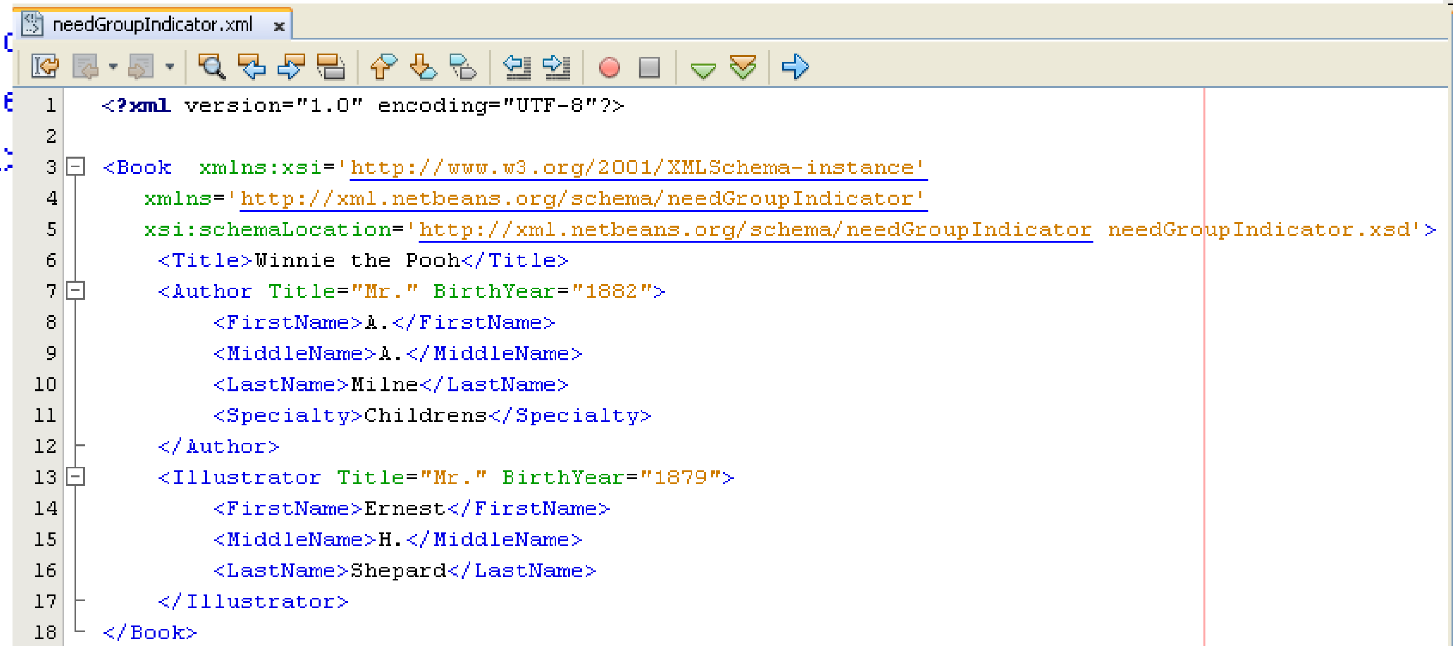
```
<xsd:element name="Book">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="Title" type="xsd:string"/>
            <xsd:element name="Author">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:group ref="person"/>
                        <xsd:element name="Specialty">
                            <xsd:simpleType>
                                <xsd:restriction base="xsd:string">
                                    <xsd:enumeration value="Mystery"/>
                                    <xsd:enumeration value="Humor"/>
                                    <xsd:enumeration value="Horror"/>
                                    <xsd:enumeration value="Childrens"/>
                                </xsd:restriction>
                            </xsd:simpleType>
                        </xsd:element>
                    </xsd:sequence>
                    <xsd:attributeGroup ref="attGroupPerson"/>
                </xsd:complexType>
            </xsd:element>
```

Schemas

Structuring – Group Indicators – Example



```
            <xsd:element name="Illustrator" minOccurs="0">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:group ref="person"/>
                    </xsd:sequence>
                    <xsd:attributeGroup ref="attGroupPerson"/>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xs
</xsd:ele
</xsd:schema
```



needGroupIndicator.xml

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

<Book  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/needGroupIndicator'
   xsi:schemaLocation='http://xml.netbeans.org/schema/needGroupIndicator needGroupIndicator.xsd'>
    <Title>Winnie the Pooh</Title>
    <Author Title="Mr." BirthYear="1882">
        <FirstName>A.</FirstName>
        <MiddleName>A.</MiddleName>
        <LastName>Milne</LastName>
        <Specialty>Childrens</Specialty>
    </Author>
    <Illustrator Title="Mr." BirthYear="1879">
        <FirstName>Ernest</FirstName>
        <MiddleName>H.</MiddleName>
        <LastName>Shepard</LastName>
    </Illustrator>
</Book>
```

Schemas

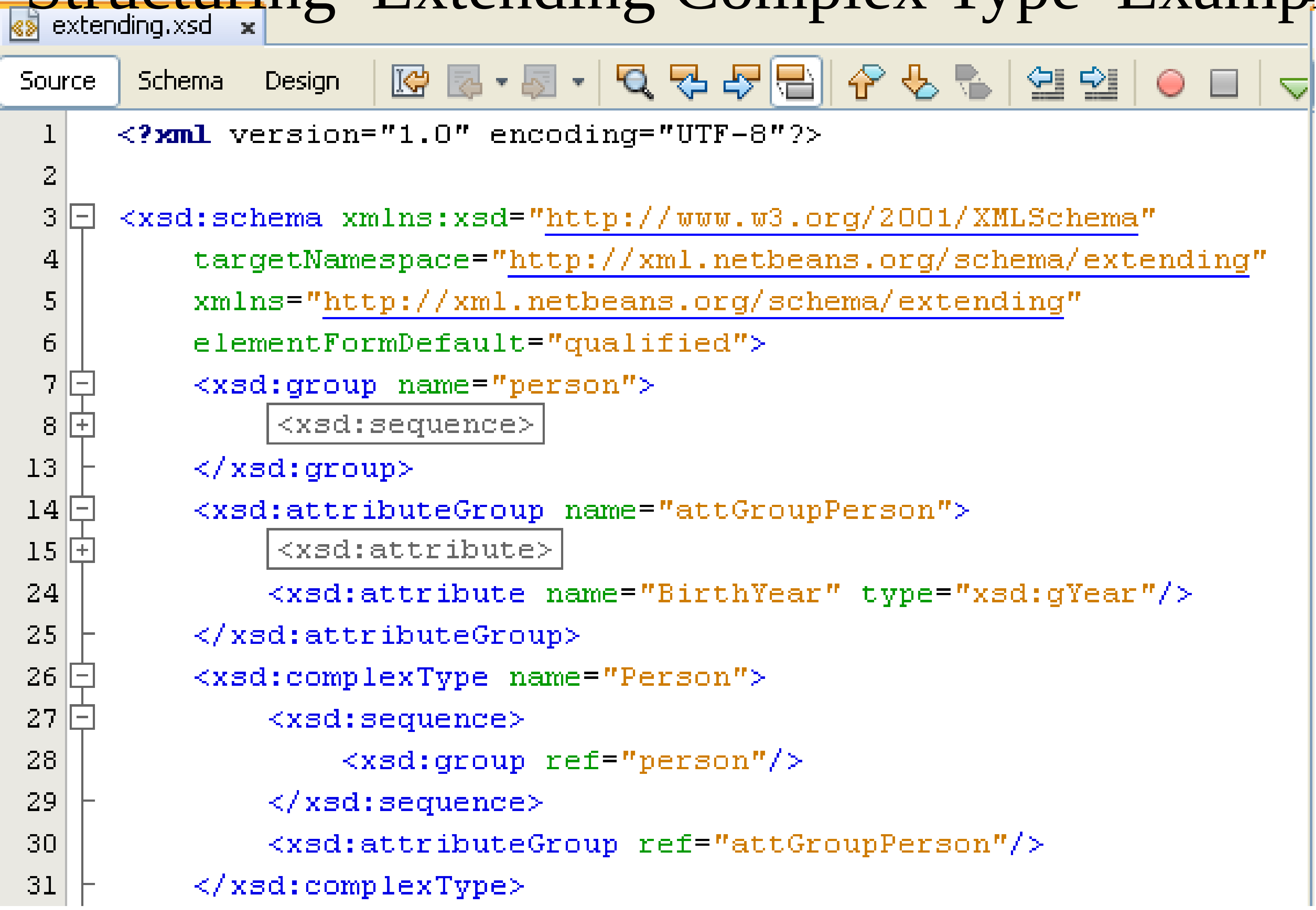
Structuring – Extending Complex Type

- **New** complex types can **be derived** by **extending existing complex types**
  - **Both elements and attributes** can be **added** in the **new type**, but **nothing** in the existing type can **be overridden**
  - **New** elements **are appended to** the **content model**, such that the original elements and new elements act as two groups that must appear in sequence
- **Syntax**

```
<xsd:extension base="base_type">
   …
</xsd:extension>
```

Schemas

Structuring–Extending Complex Type–Example



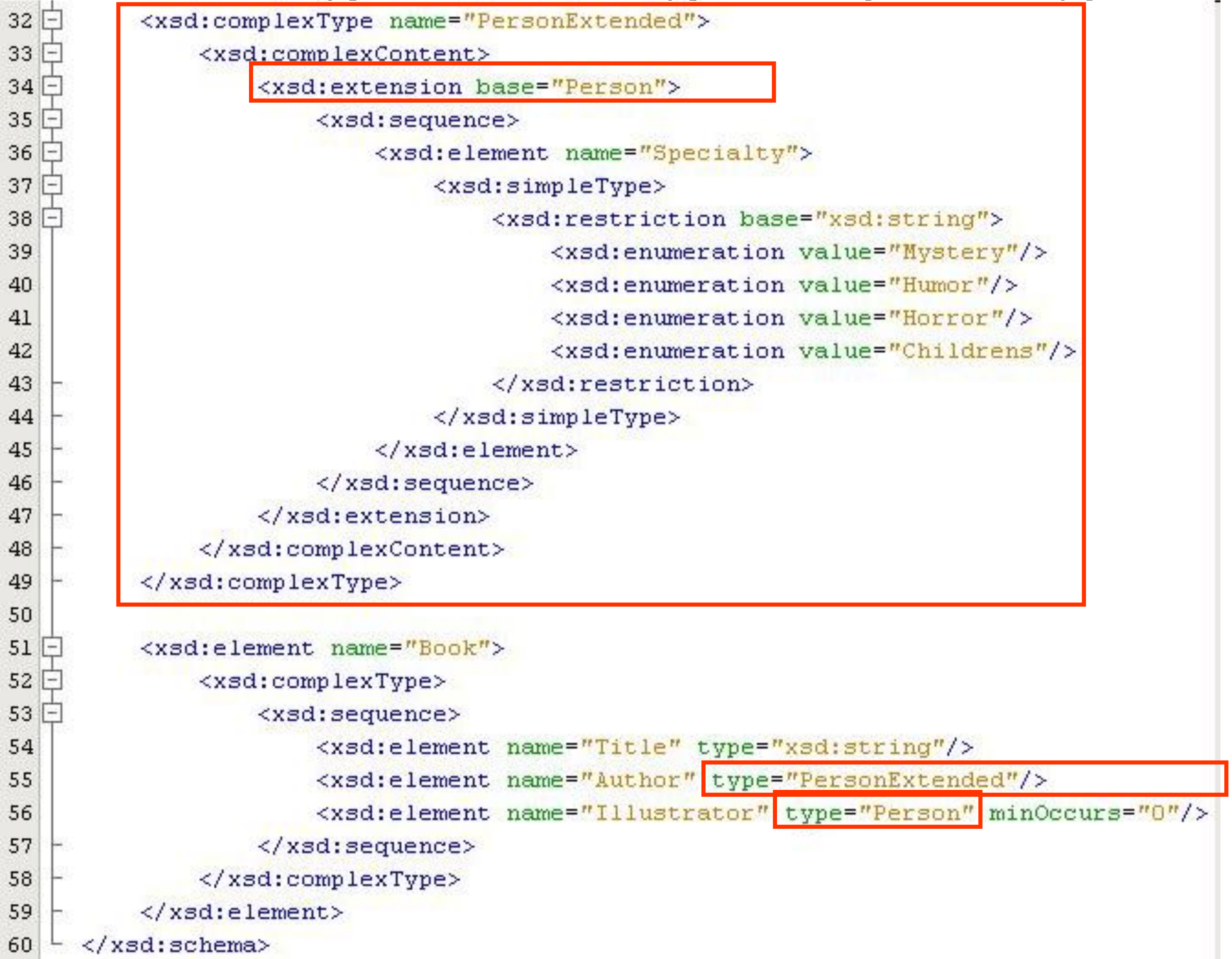
extending.xsd

Source Schema Design

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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/extending"
    xmlns="http://xml.netbeans.org/schema/extending"
    elementFormDefault="qualified">
    <xsd:group name="person">
        <xsd:sequence>
    </xsd:group>
    <xsd:attributeGroup name="attGroupPerson">
        <xsd:attribute>
        <xsd:attribute name="BirthYear" type="xsd:gYear"/>
    </xsd:attributeGroup>
    <xsd:complexType name="Person">
        <xsd:sequence>
            <xsd:group ref="person"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="attGroupPerson"/>
    </xsd:complexType>
```

Schemas

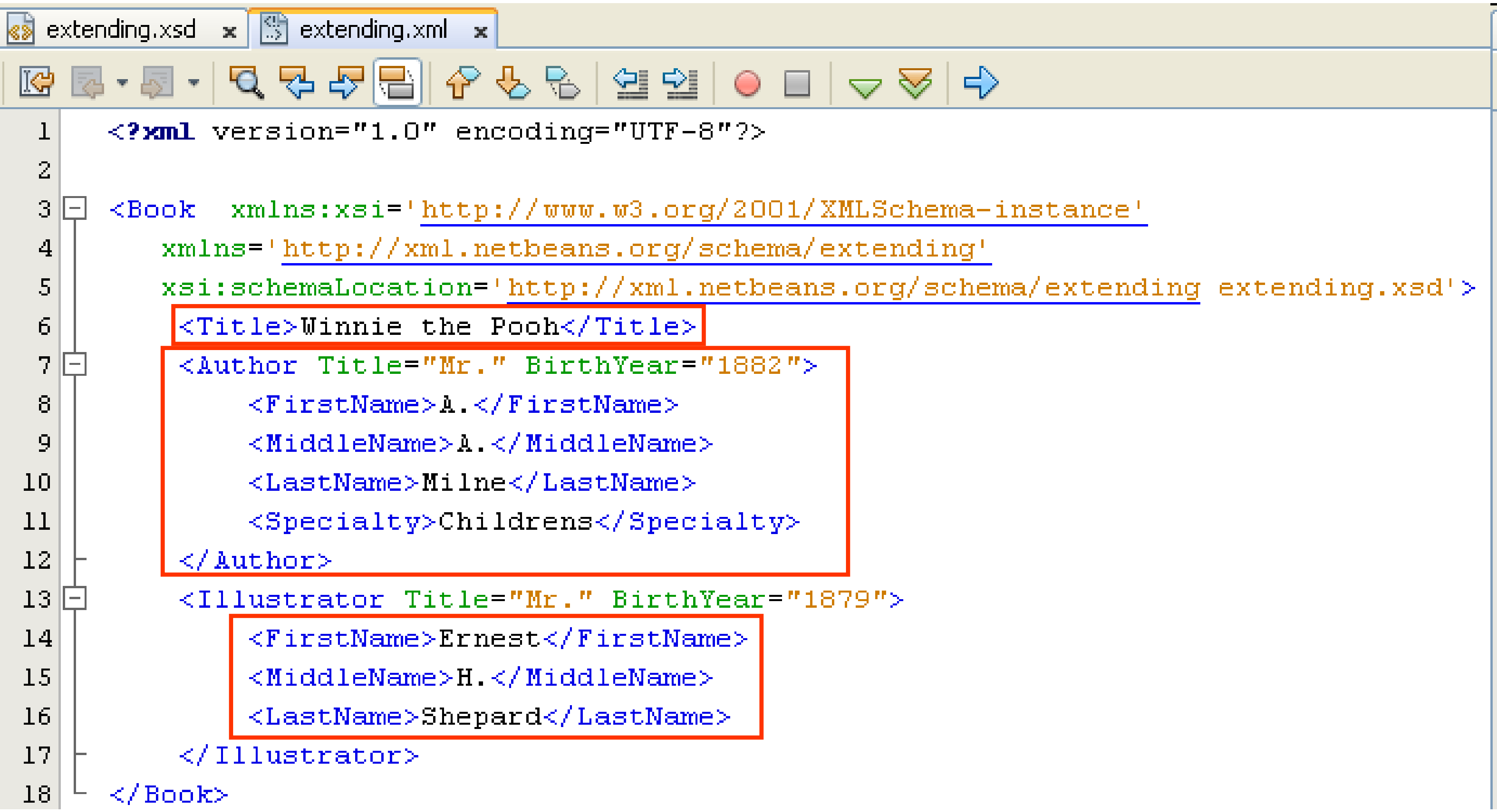
Structuring–Extending Complex Type–Example



```
    <xsd:complexType name="PersonExtended">
        <xsd:complexContent>
            <xsd:extension base="Person">
                <xsd:sequence>
                    <xsd:element name="Specialty">
                        <xsd:simpleType>
                            <xsd:restriction base="xsd:string">
                                <xsd:enumeration value="Mystery"/>
                                <xsd:enumeration value="Humor"/>
                                <xsd:enumeration value="Horror"/>
                                <xsd:enumeration value="Childrens"/>
                            </xsd:restriction>
                        </xsd:simpleType>
                    </xsd:element>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:element name="Book">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Title" type="xsd:string"/>
                <xsd:element name="Author" type="PersonExtended"/>
                <xsd:element name="Illustrator" type="Person" minOccurs="0"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas

Structuring–Extending Complex Type–Example



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

<Book  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/extending'
   xsi:schemaLocation='http://xml.netbeans.org/schema/extending extending.xsd'>
    <Title>Winnie the Pooh</Title>
    <Author Title="Mr." BirthYear="1882">
        <FirstName>A.</FirstName>
        <MiddleName>A.</MiddleName>
        <LastName>Milne</LastName>
        <Specialty>Childrens</Specialty>
    </Author>
    <Illustrator Title="Mr." BirthYear="1879">
        <FirstName>Ernest</FirstName>
        <MiddleName>H.</MiddleName>
        <LastName>Shepard</LastName>
    </Illustrator>
</Book>
```

Schemas
Structuring – Abstract Types

- Is used to **define** the **arbitrary complex type** that will be **defined** at **used time**
- When a type is made abstract, it **cannot** be **used directly** in an XML instance
  - One of its **derived types** must be **used** instead
- The **derived** type is **identified** in the **instance** document using the **xsi:type attribute**
- **Syntax**
  - Abstract Type Declaration

<xsd:complexType name="name_type" abstract="true">

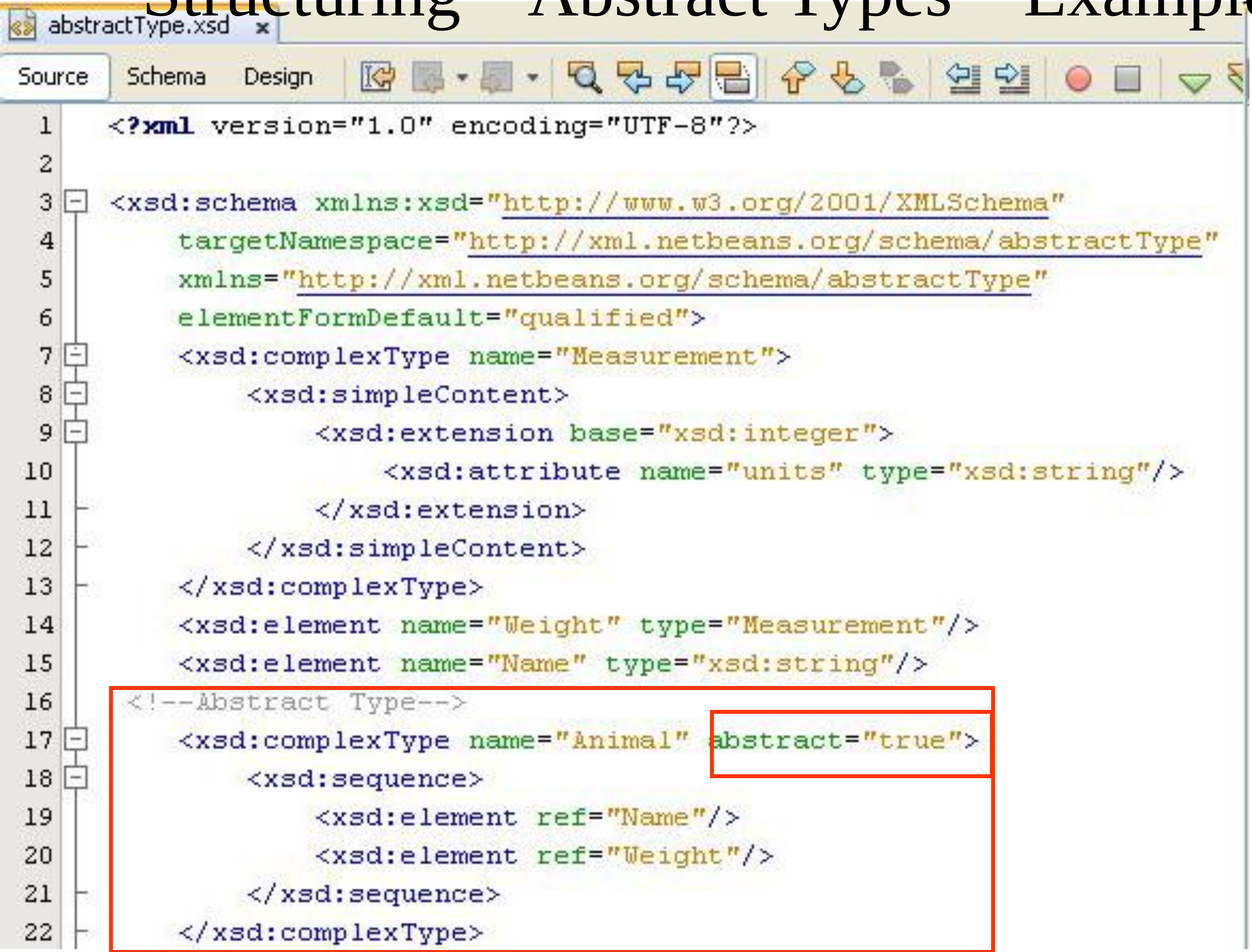
…

</xsd:complexType>

  - Derived Type Declaration from Abstract Type: using **extending complex type** mechanism
  - **Using in XML document:**

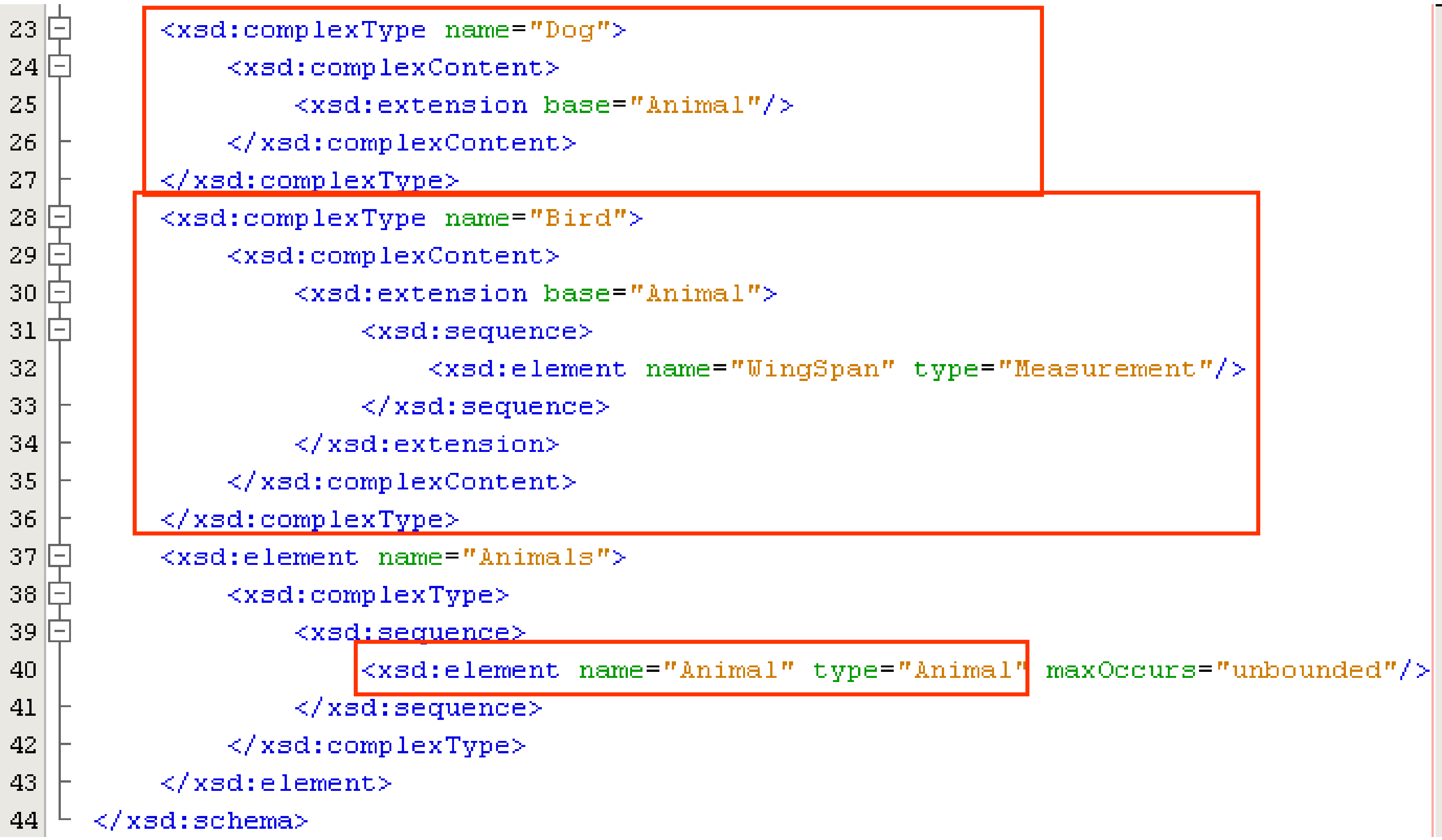
<element_Name xsi:type="derivedType">…</element_Name>

Schemas
Structuring – Abstract Types – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/abstractType"
    xmlns="http://xml.netbeans.org/schema/abstractType"
    elementFormDefault="qualified">
    <xsd:complexType name="Measurement">
        <xsd:simpleContent>
            <xsd:extension base="xsd:integer">
                <xsd:attribute name="units" type="xsd:string"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:element name="Weight" type="Measurement"/>
    <xsd:element name="Name" type="xsd:string"/>
  <!--Abstract Type-->
    <xsd:complexType name="Animal" abstract="true">
        <xsd:sequence>
            <xsd:element ref="Name"/>
            <xsd:element ref="Weight"/>
        </xsd:sequence>
    </xsd:complexType>
```

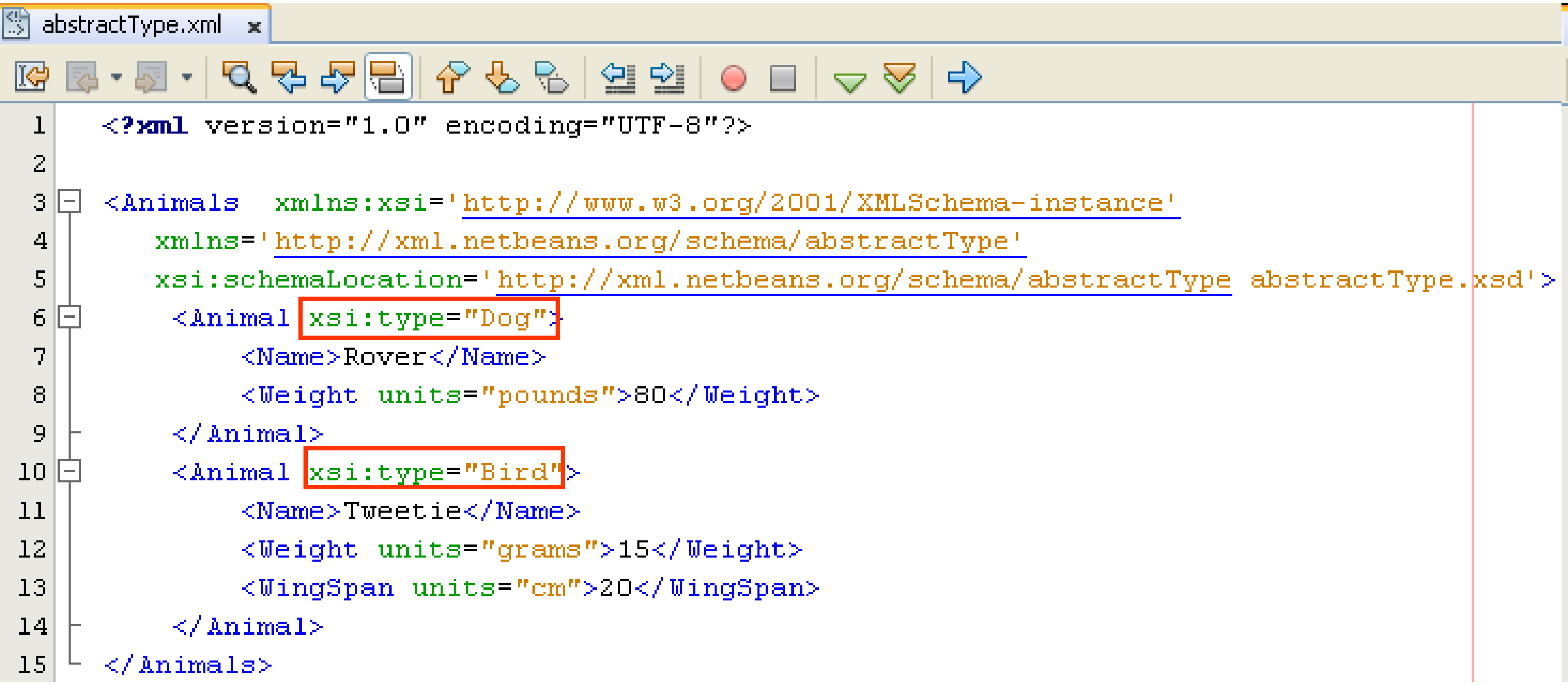
Schemas
Structuring – Abstract Types – Example



```
    <xsd:complexType name="Dog">
        <xsd:complexContent>
            <xsd:extension base="Animal"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="Bird">
        <xsd:complexContent>
            <xsd:extension base="Animal">
                <xsd:sequence>
                    <xsd:element name="WingSpan" type="Measurement"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="Animals">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Animal" type="Animal" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

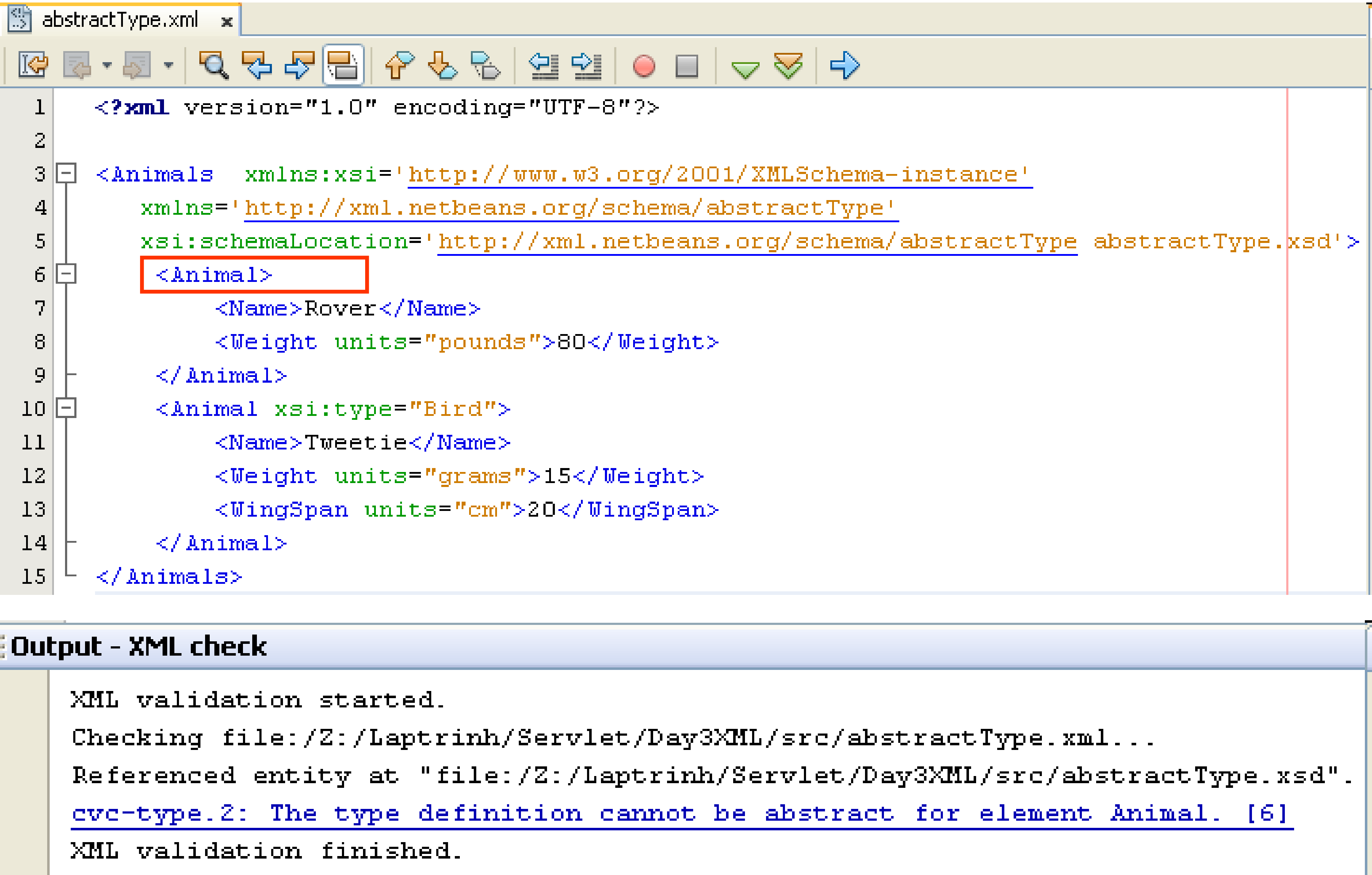
Schemas

Structuring – Abstract Types – Example



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

<Animals  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/abstractType'
   xsi:schemaLocation='http://xml.netbeans.org/schema/abstractType abstractType.xsd'>
    <Animal xsi:type="Dog">
        <Name>Rover</Name>
        <Weight units="pounds">80</Weight>
    </Animal>
    <Animal xsi:type="Bird">
        <Name>Tweetie</Name>
        <Weight units="grams">15</Weight>
        <WingSpan units="cm">20</WingSpan>
    </Animal>
</Animals>
```

Schemas
Structuring – Abstract Types – Example



abstractType.xml

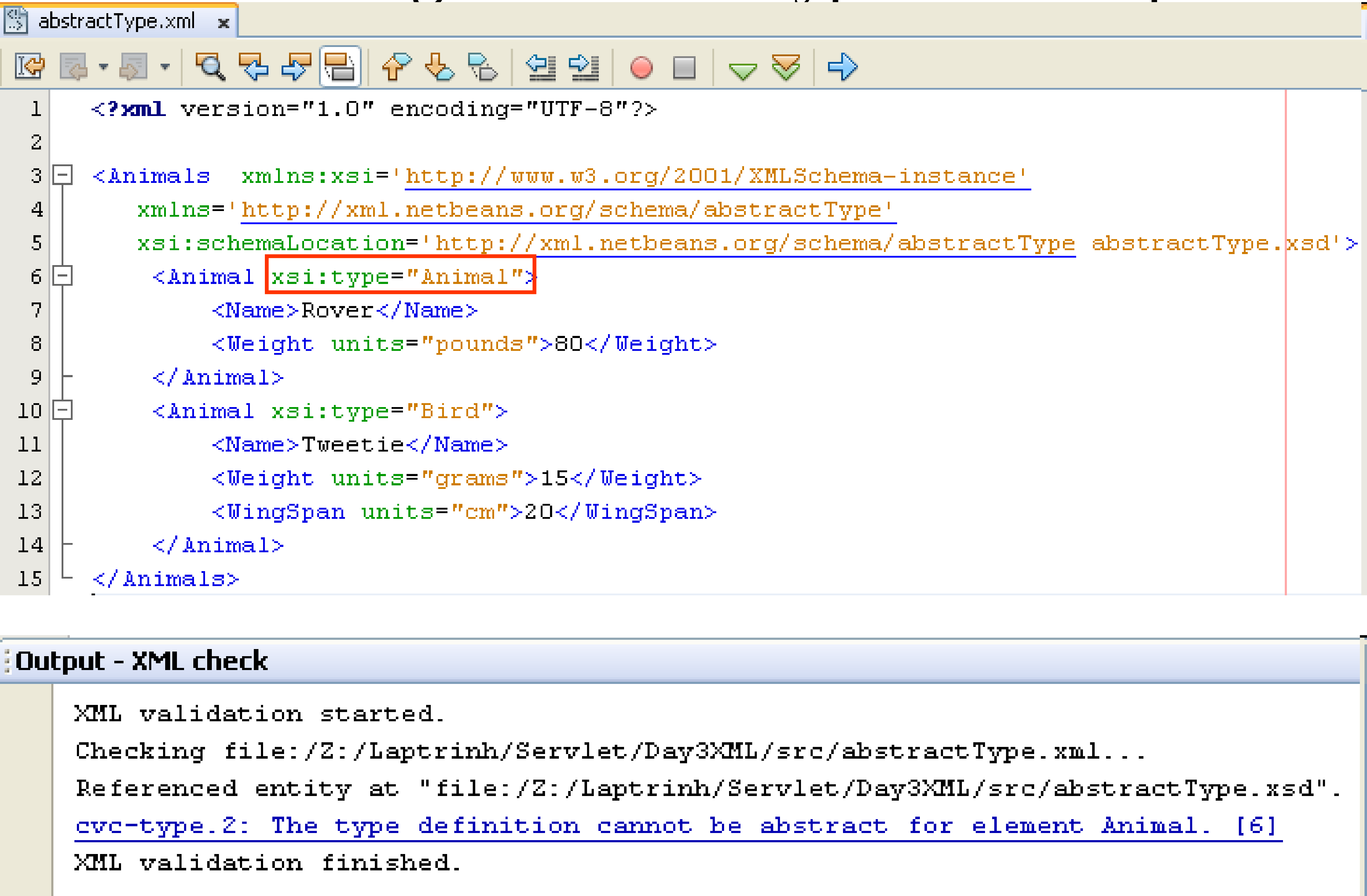
```
<?xml version="1.0" encoding="UTF-8"?>

<Animals  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/abstractType'
   xsi:schemaLocation='http://xml.netbeans.org/schema/abstractType abstractType.xsd'>
    <Animal>
        <Name>Rover</Name>
        <Weight units="pounds">80</Weight>
    </Animal>
    <Animal xsi:type="Bird">
        <Name>Tweetie</Name>
        <Weight units="grams">15</Weight>
        <WingSpan units="cm">20</WingSpan>
    </Animal>
</Animals>
```

Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/abstractType.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/abstractType.xsd".
cvc-type.2: The type definition cannot be abstract for element Animal. [6]
XML validation finished.
```

Schemas

Structuring – Abstract Types – Example



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

<Animals  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/abstractType'
   xsi:schemaLocation='http://xml.netbeans.org/schema/abstractType abstractType.xsd'>
    <Animal xsi:type="Animal">
        <Name>Rover</Name>
        <Weight units="pounds">80</Weight>
    </Animal>
    <Animal xsi:type="Bird">
        <Name>Tweetie</Name>
        <Weight units="grams">15</Weight>
        <WingSpan units="cm">20</WingSpan>
    </Animal>
</Animals>
```

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/abstractType.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/abstractType.xsd".
cvc-type.2: The type definition cannot be abstract for element Animal. [6]
XML validation finished.
```

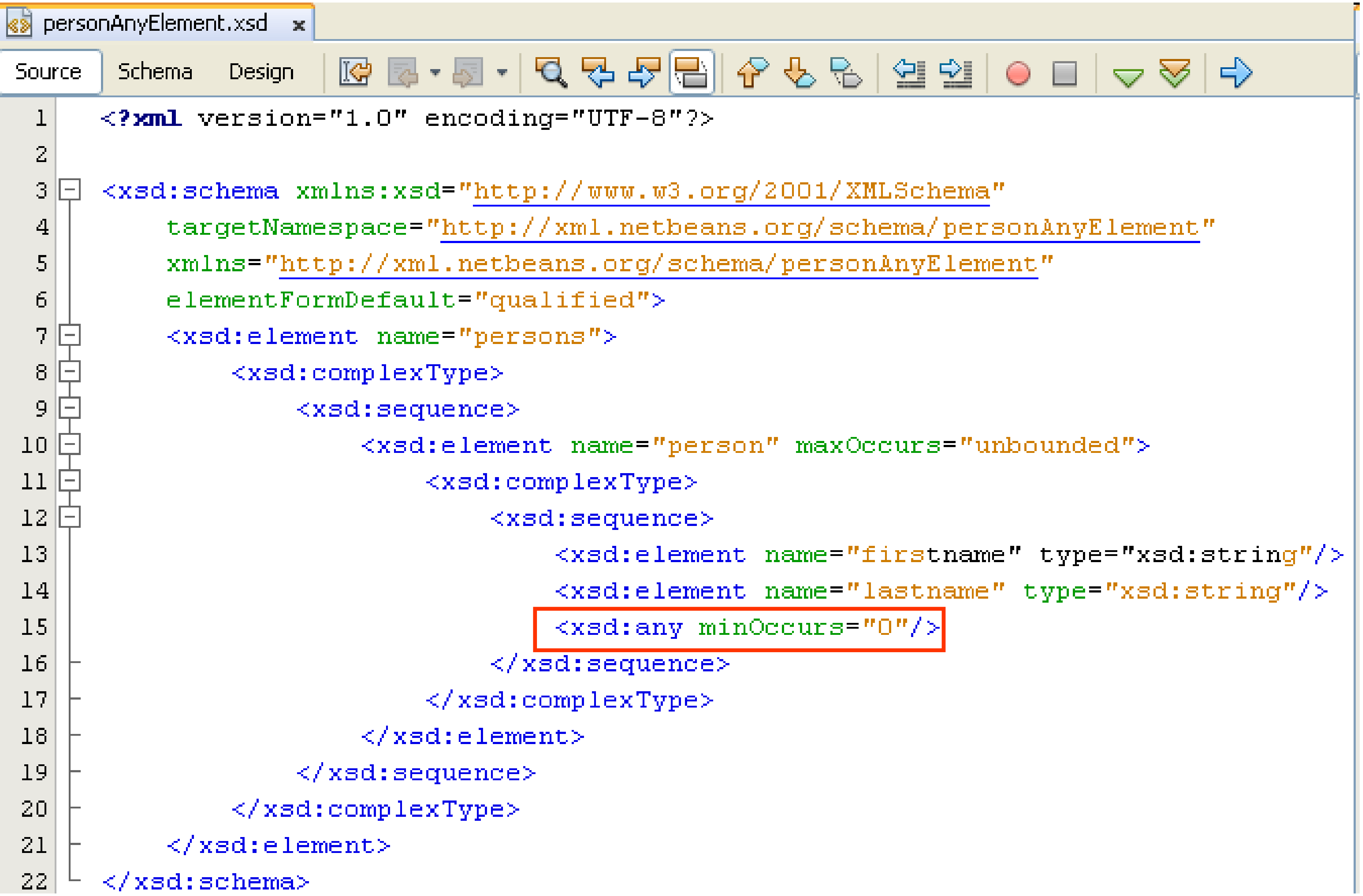
Schemas
Structuring – Any Elements

- **Allows** to **extend** the XML document with **elements** or **attributes not specified** by the schema
- **Allows** documents to **contain additional elements or attributes** that are **not declared** in the main XML schema
- **Syntax**
  - **Elements**

    <xsd:any/>
  - **Attributes**

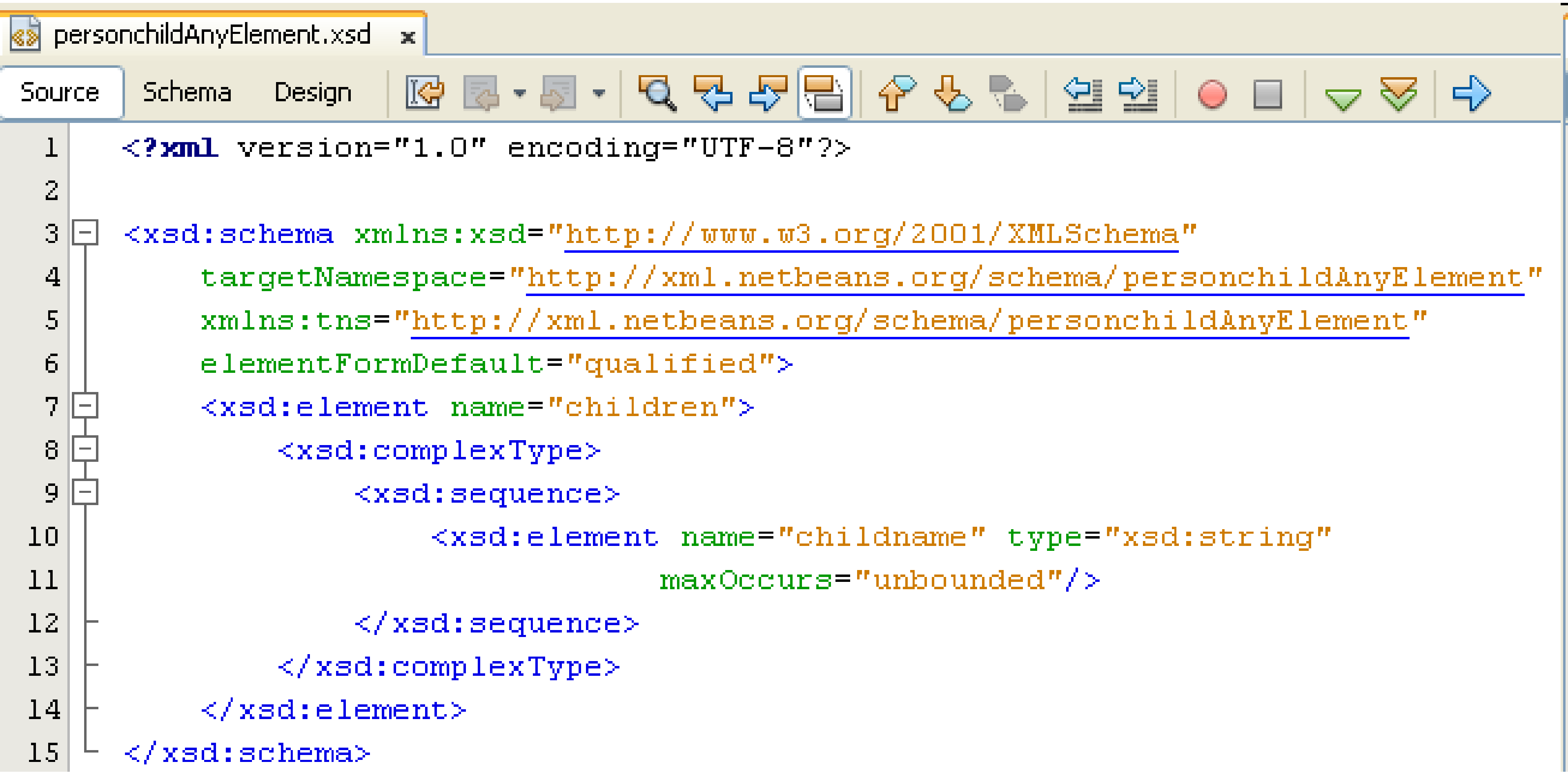
    <xsd:anyAttribute/>

Schemas
Structuring – Any Elements – Example



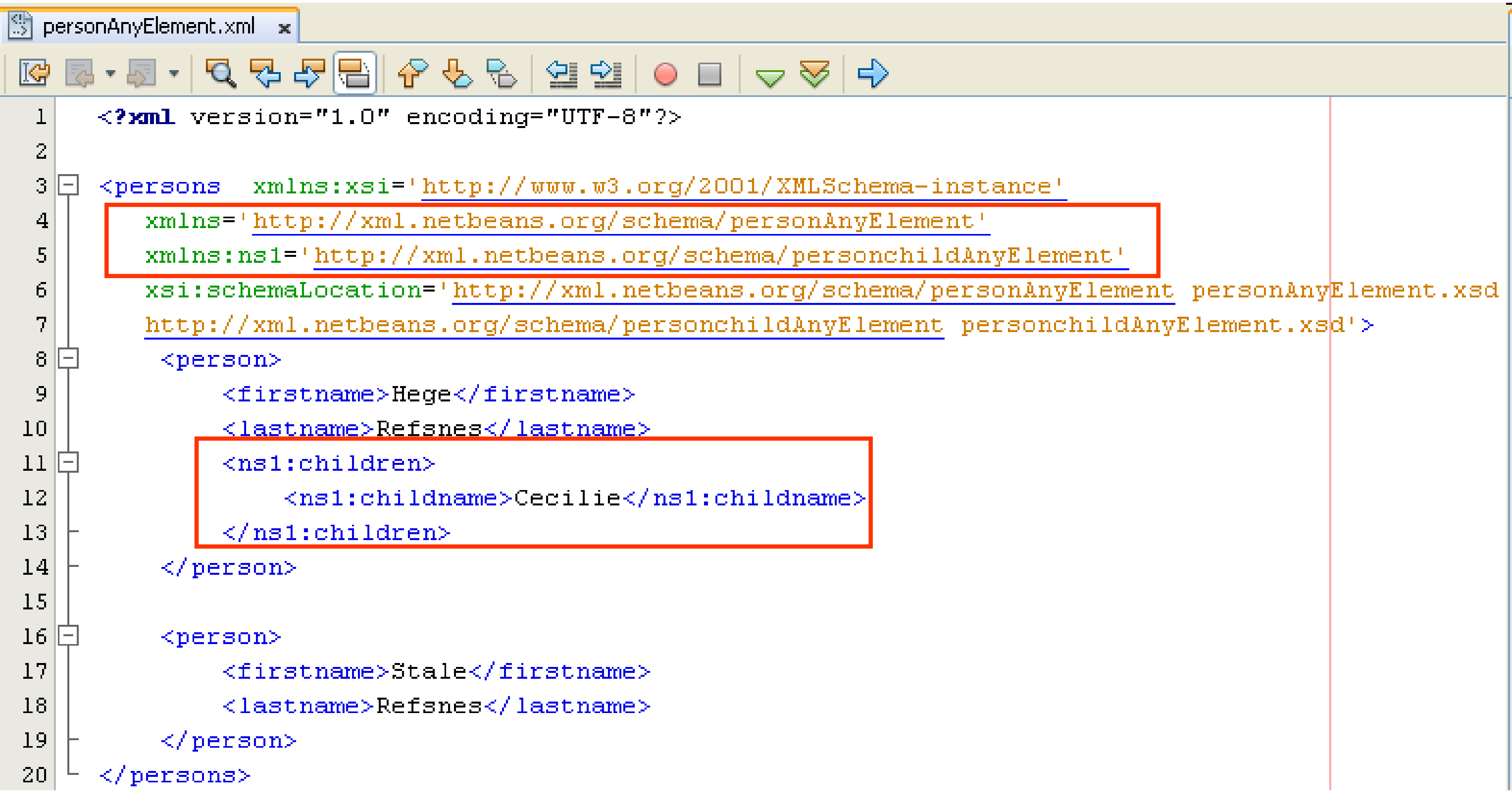
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/personAnyElement"
    xmlns="http://xml.netbeans.org/schema/personAnyElement"
    elementFormDefault="qualified">
    <xsd:element name="persons">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="person" maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="firstname" type="xsd:string"/>
                            <xsd:element name="lastname" type="xsd:string"/>
                            <xsd:any minOccurs="0"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas
Structuring – Any Elements – Example



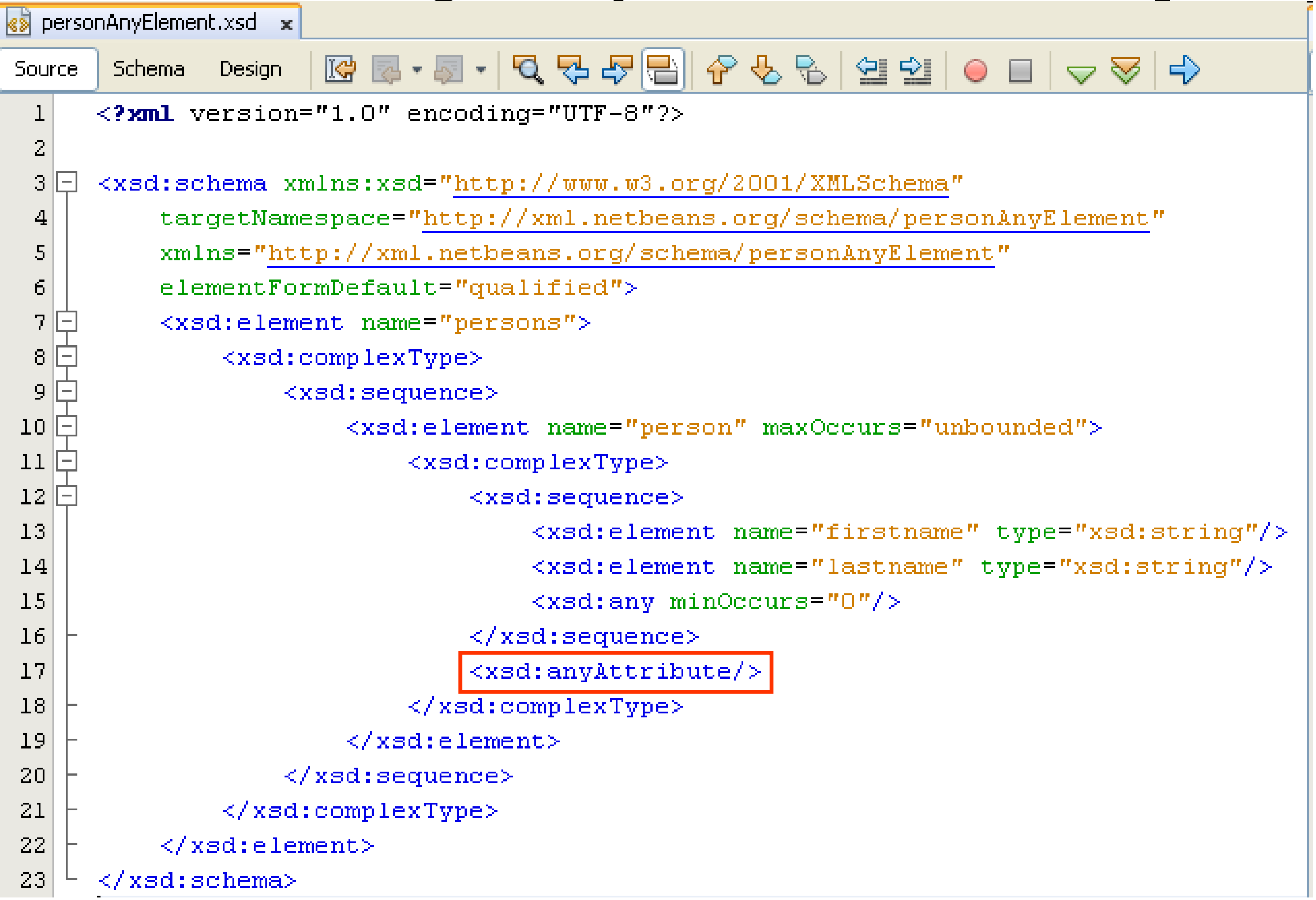
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/personchildAnyElement"
    xmlns:tns="http://xml.netbeans.org/schema/personchildAnyElement"
    elementFormDefault="qualified">
    <xsd:element name="children">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="childname" type="xsd:string"
                            maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas
Structuring – Any Elements – Example



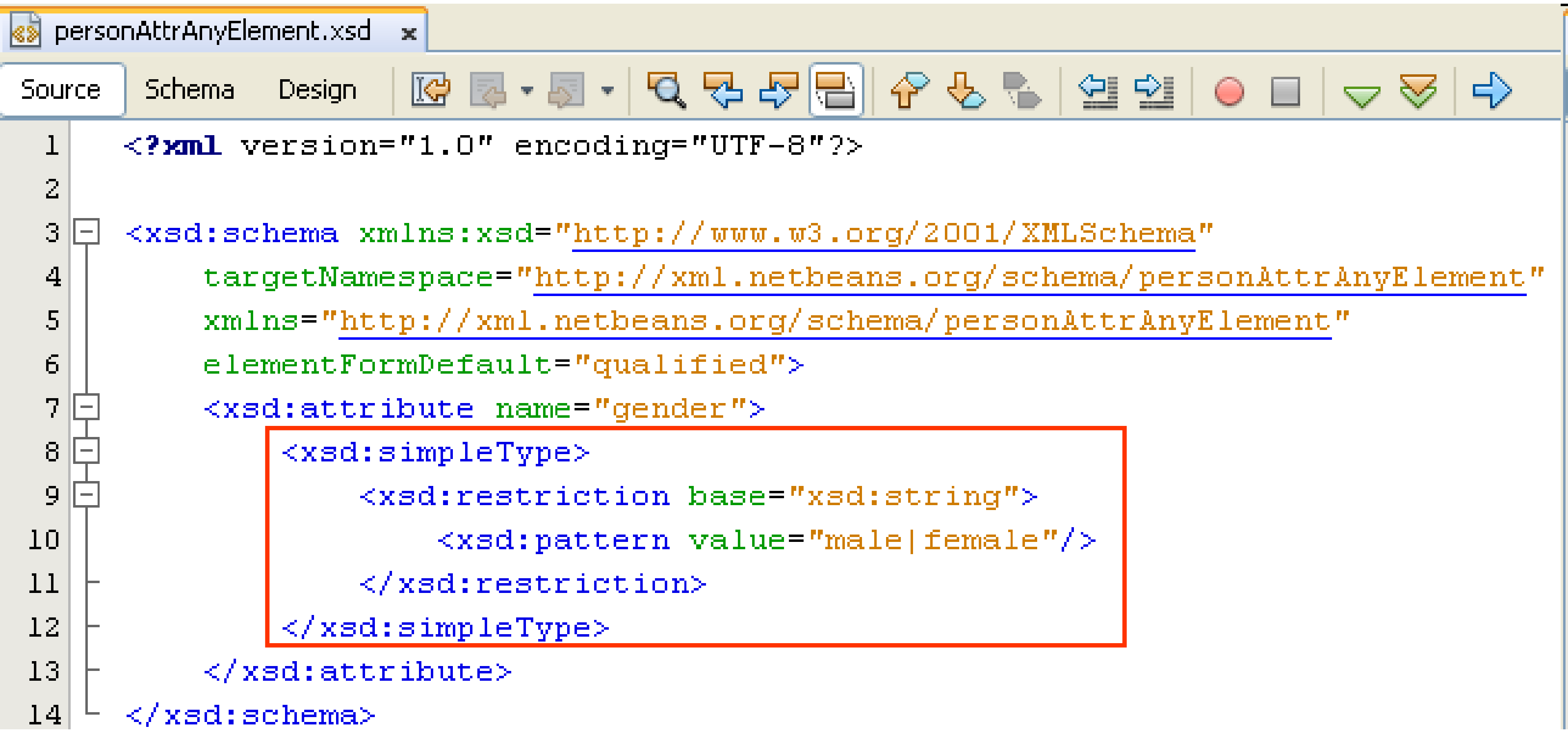
```
<?xml version="1.0" encoding="UTF-8"?>

<persons  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/personAnyElement'
   xmlns:ns1='http://xml.netbeans.org/schema/personchildAnyElement'
   xsi:schemaLocation='http://xml.netbeans.org/schema/personAnyElement personAnyElement.xsd
   http://xml.netbeans.org/schema/personchildAnyElement personchildAnyElement.xsd'>
    <person>
        <firstname>Hege</firstname>
        <lastname>Refsnes</lastname>
        <ns1:children>
            <ns1:childname>Cecilie</ns1:childname>
        </ns1:children>
    </person>

    <person>
        <firstname>Stale</firstname>
        <lastname>Refsnes</lastname>
    </person>
</persons>
```

Schemas
Structuring – Any Elements – Example



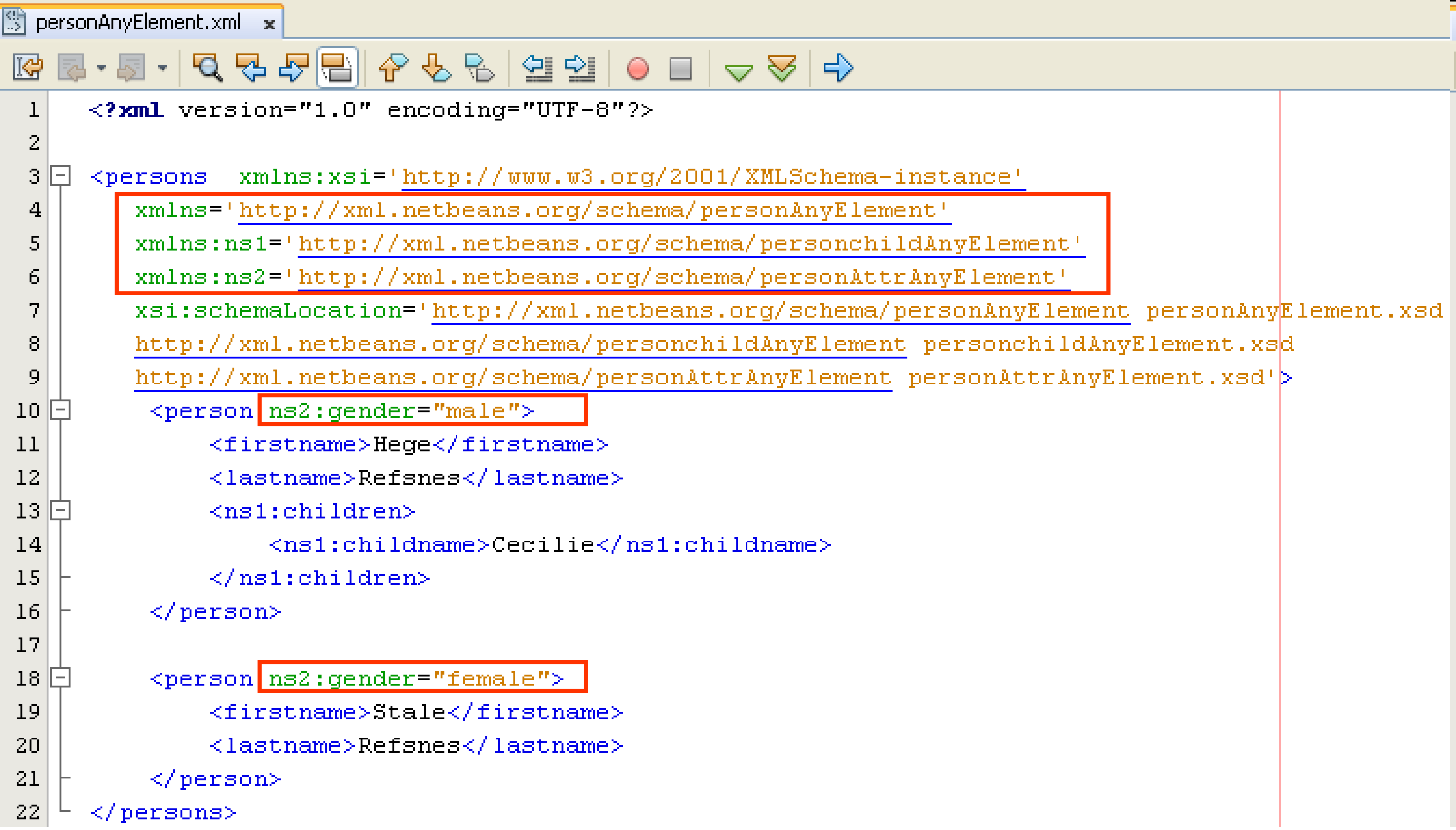
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/personAnyElement"
    xmlns="http://xml.netbeans.org/schema/personAnyElement"
    elementFormDefault="qualified">
    <xsd:element name="persons">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="person" maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="firstname" type="xsd:string"/>
                            <xsd:element name="lastname" type="xsd:string"/>
                            <xsd:any minOccurs="0"/>
                        </xsd:sequence>
                        <xsd:anyAttribute/>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas
Structuring – Any Elements – Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/personAttrAnyElement"
    xmlns="http://xml.netbeans.org/schema/personAttrAnyElement"
    elementFormDefault="qualified">
    <xsd:attribute name="gender">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:pattern value="male|female"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:schema>
```

Schemas
Structuring – Any Elements – Example



personAnyElement.xml

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

<persons  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/personAnyElement'
   xmlns:ns1='http://xml.netbeans.org/schema/personchildAnyElement'
   xmlns:ns2='http://xml.netbeans.org/schema/personAttrAnyElement'
   xsi:schemaLocation='http://xml.netbeans.org/schema/personAnyElement personAnyElement.xsd
   http://xml.netbeans.org/schema/personchildAnyElement personchildAnyElement.xsd
   http://xml.netbeans.org/schema/personAttrAnyElement personAttrAnyElement.xsd'>
    <person ns2:gender="male">
        <firstname>Hege</firstname>
        <lastname>Refsnes</lastname>
        <ns1:children>
            <ns1:childname>Cecilie</ns1:childname>
        </ns1:children>
    </person>

    <person ns2:gender="female">
        <firstname>Stale</firstname>
        <lastname>Refsnes</lastname>
    </person>
</persons>
```

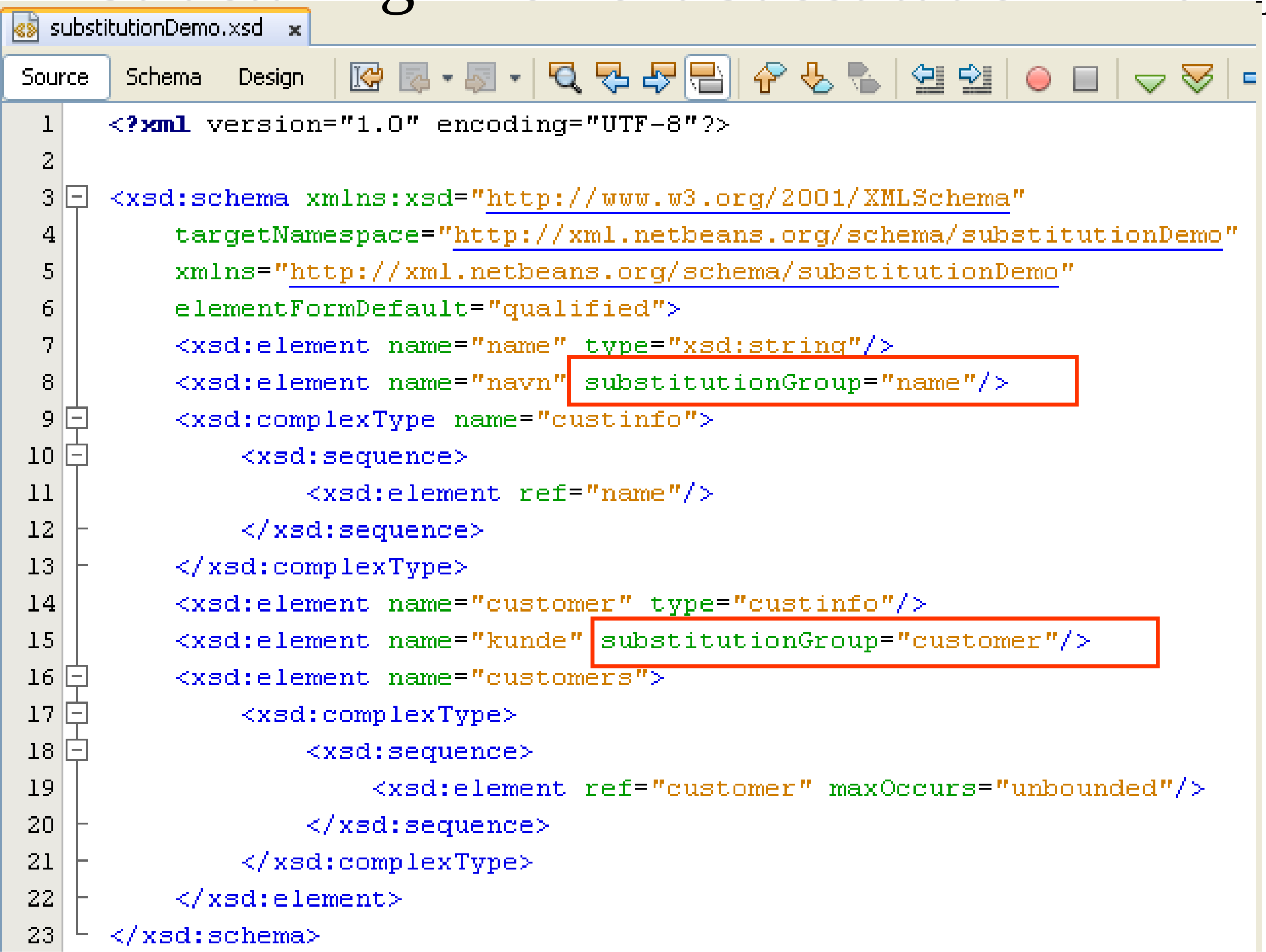
Schemas
Structuring – XSD Element Substitution

- One element can **substitute** another element
- A **substitutionGroup** is a solution
  - First, a **head element** is **declared**
  - Then, the **other elements** which **state** that they are **substitutable for** the **head element** are **declared**
- **Syntax**
  - **First:** declare the **key** element

    <xsd:element name="element_name" type="data_type" [block="substitution"]/>
    - **block** keyword is used to block element substitution
  - **Second:** declare the **substitution element with substitutionGroup** attribute

    <xsd:element name="alternative_name" substitutionGroup="element_name" />
- **Notes**
  - The **type** of the **substitutable elements** must be the **same as**, or **derived from**, the type of the head element.
  - All elements in the **substitutionGroup** (the head element and the substitutable elements) must be **declared** as **global elements**

Schemas
Structuring–Element Substitution–Example

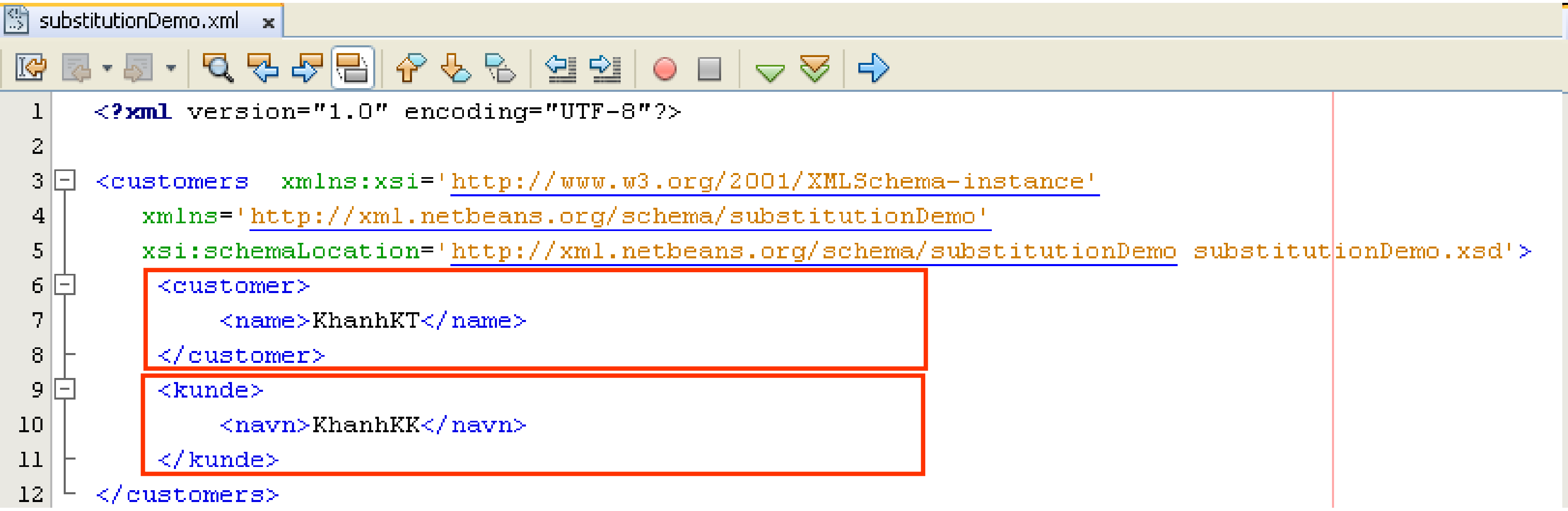


substitutionDemo.xsd

Source Schema Design

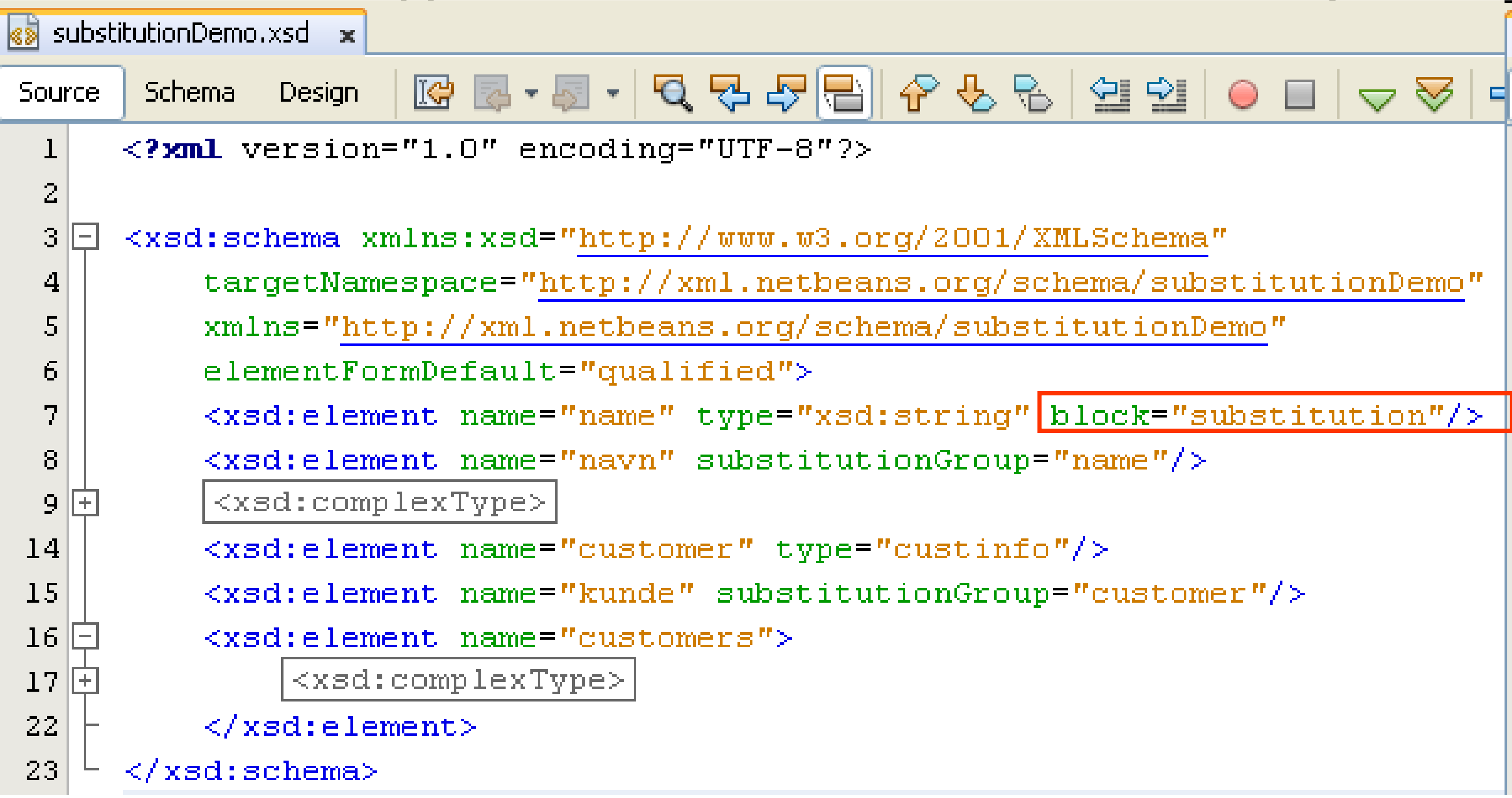
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/substitutionDemo"
    xmlns="http://xml.netbeans.org/schema/substitutionDemo"
    elementFormDefault="qualified">
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="navn" substitutionGroup="name"/>
    <xsd:complexType name="custinfo">
        <xsd:sequence>
            <xsd:element ref="name"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="customer" type="custinfo"/>
    <xsd:element name="kunde" substitutionGroup="customer"/>
    <xsd:element name="customers">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="customer" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas
Structuring–Element Substitution–Example


substitutionDemo.xml

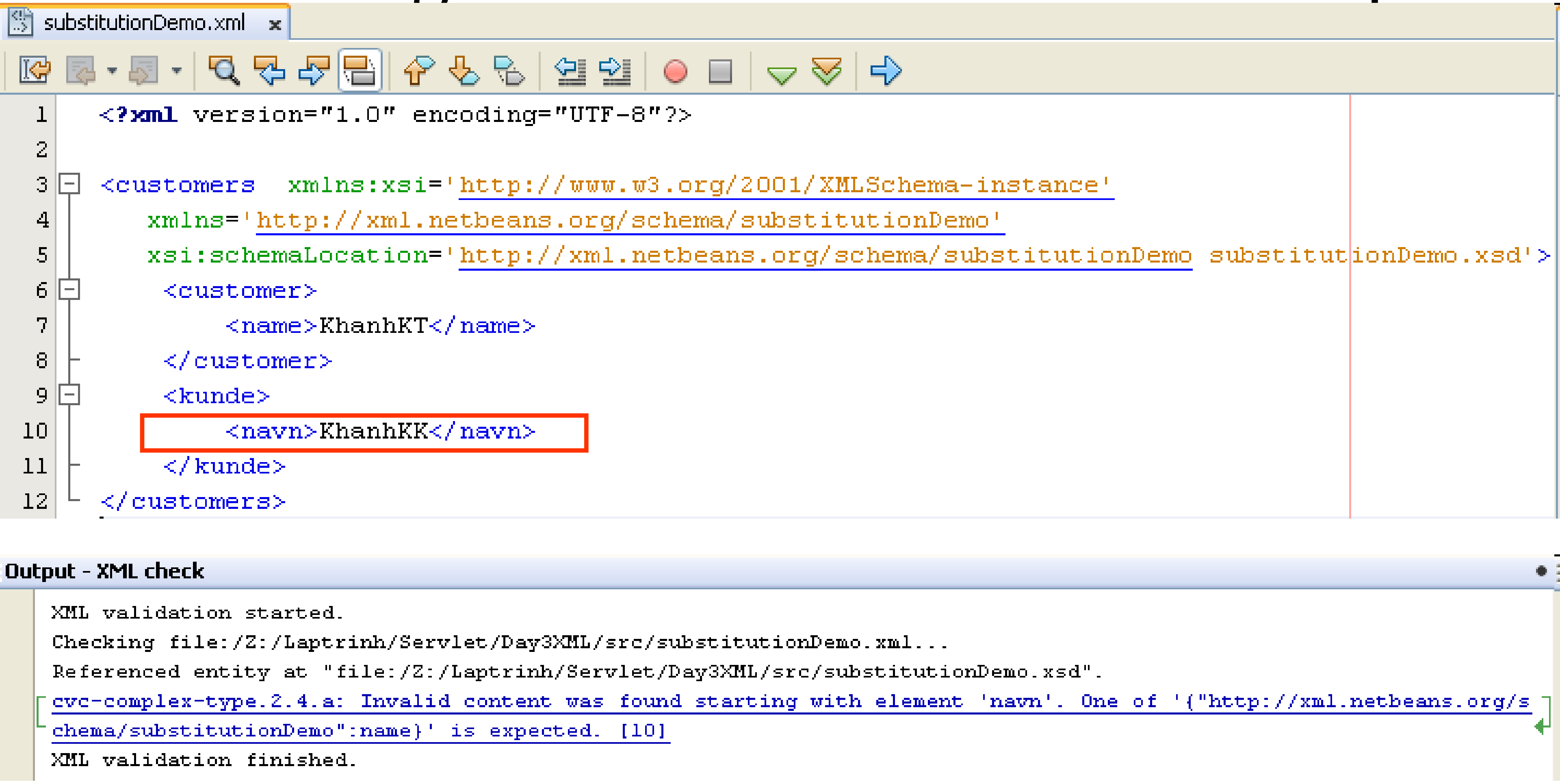
```
<?xml version="1.0" encoding="UTF-8"?>

<customers  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/substitutionDemo'
    xsi:schemaLocation='http://xml.netbeans.org/schema/substitutionDemo substitutionDemo.xsd'>
      <customer>
          <name>KhanhKT</name>
      </customer>
      <kunde>
          <navn>KhanhKK</navn>
      </kunde>
</customers>
```

Schemas
Structuring–Element Substitution–Example



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/substitutionDemo"
    xmlns="http://xml.netbeans.org/schema/substitutionDemo"
    elementFormDefault="qualified">
    <xsd:element name="name" type="xsd:string" block="substitution"/>
    <xsd:element name="navn" substitutionGroup="name"/>
    <xsd:complexType>
    <xsd:element name="customer" type="custinfo"/>
    <xsd:element name="kunde" substitutionGroup="customer"/>
    <xsd:element name="customers">
        <xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas
Structuring–Element Substitution–Example



substitutionDemo.xml

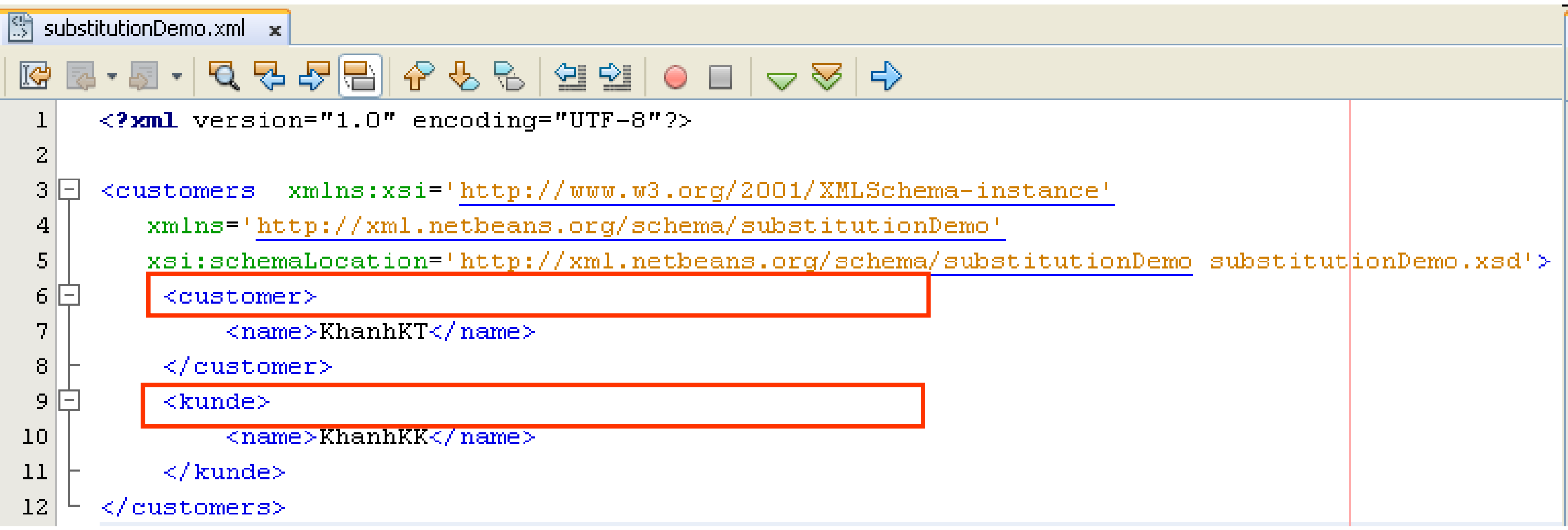
```
<?xml version="1.0" encoding="UTF-8"?>

<customers  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
   xmlns='http://xml.netbeans.org/schema/substitutionDemo'
   xsi:schemaLocation='http://xml.netbeans.org/schema/substitutionDemo substitutionDemo.xsd'>
    <customer>
        <name>KhanhKT</name>
    </customer>
    <kunde>
        <navn>KhanhKK</navn>
    </kunde>
</customers>
```

Output - XML check

```
XML validation started.
Checking file:/Z:/Laptrinh/Servlet/Day3XML/src/substitutionDemo.xml...
Referenced entity at "file:/Z:/Laptrinh/Servlet/Day3XML/src/substitutionDemo.xsd".
cvc-complex-type.2.4.a: Invalid content was found starting with element 'navn'. One of '{"http://xml.netbeans.org/s
chema/substitutionDemo":name}' is expected. [10]
XML validation finished.
```

Schemas

Structuring–Element Substitution–Example



substitutionDemo.xml

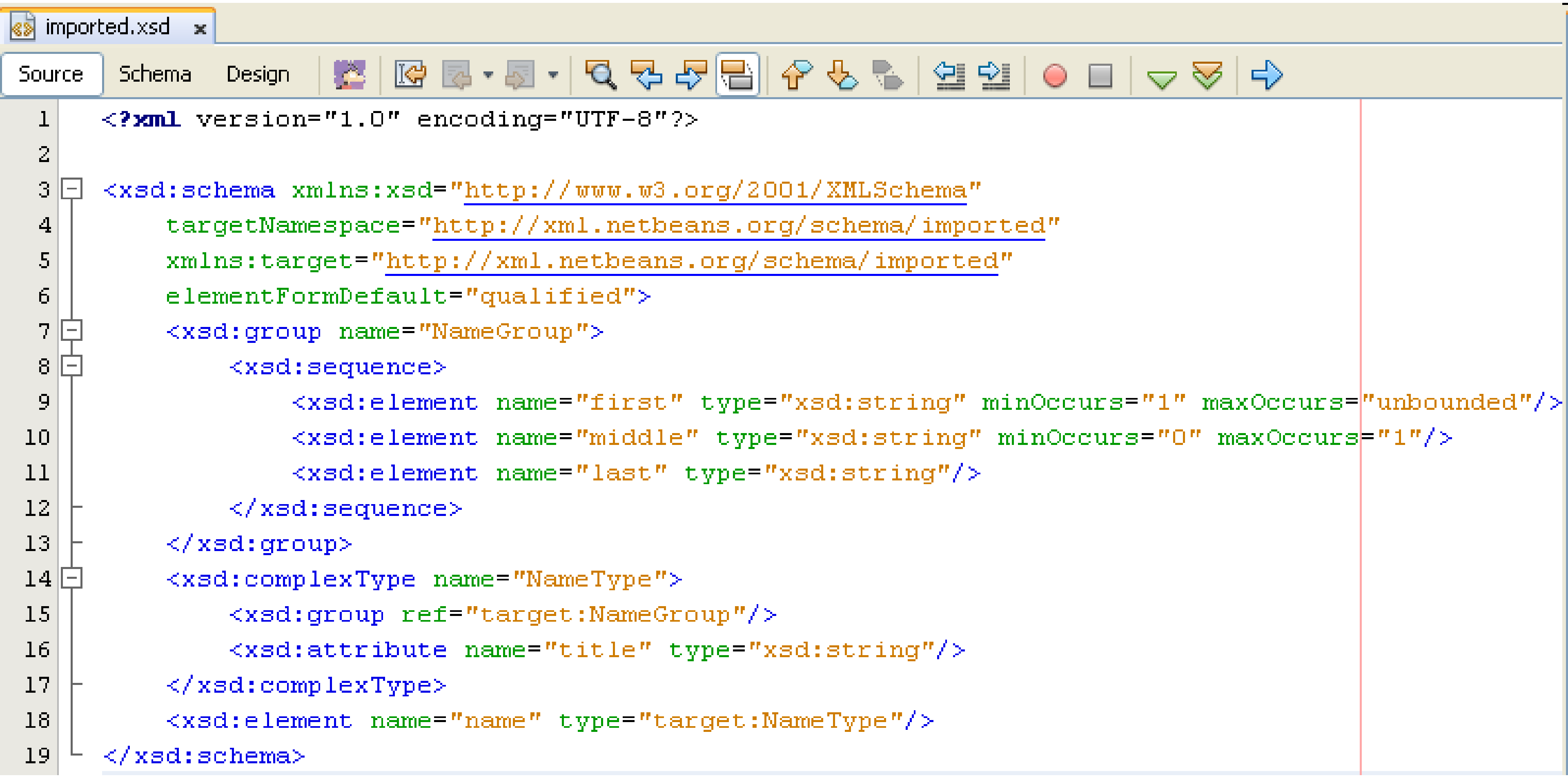
```
<?xml version="1.0" encoding="UTF-8"?>

<customers  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/substitutionDemo'
    xsi:schemaLocation='http://xml.netbeans.org/schema/substitutionDemo substitutionDemo.xsd'>
      <customer>
          <name>KhanhKT</name>
      </customer>
      <kunde>
          <name>KhanhKK</name>
      </kunde>
</customers>
```

Schemas
Creating schema from multiple documents

- This is **mechanisms** for **combining** XML **Schemas** and **reusing definitions**
- **Import** declaration
  - Allows to **import global declarations** from other XML Schemas.
  - Is used **primarily** for **combining** XML **Schemas** that have **different targetNamespaces**
  - Allows to **refer** to **declarations only within** other **XML Schemas**
  - **Must** be a **direct child** of the <**schema**> **element**
  - **Syntax**: **<xsd:import namespace="URI" schemaLocation="url"/>**
- **Include** declaration
  - Is very **similar** to the **import** declaration
  - Allows to **combine** XML **Schemas** that are designed for the **same targetNamespace** (or **no targetNamespace**) much more effectively
  - **Syntax**: **<xsd:include schemaLocation="url"/>**
  - Can be used only on **documents** with the **same targetNamespace, or no targetNamespace**

Schemas

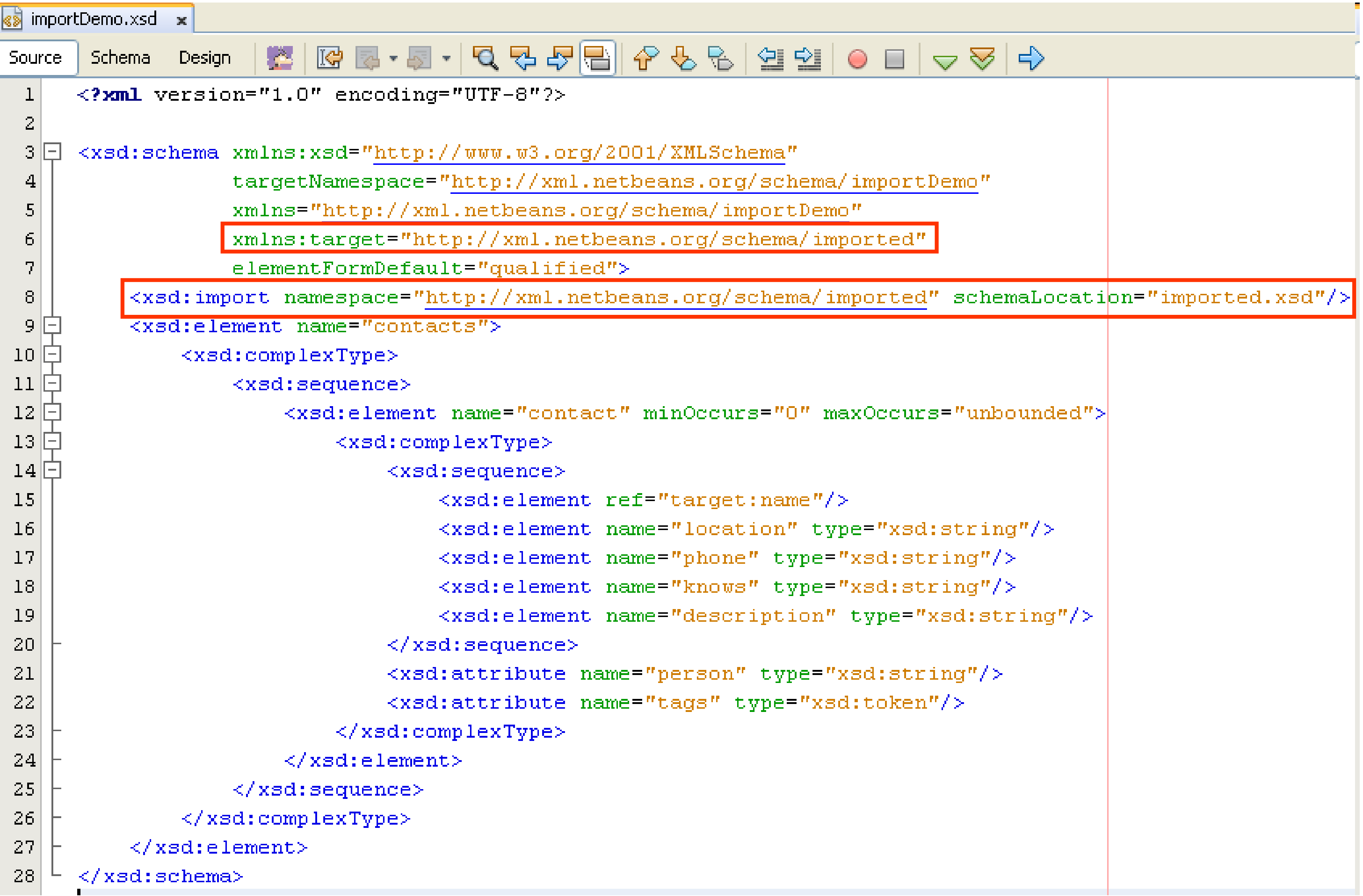
Creating schema from multiple documents



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/imported"
    xmlns:target="http://xml.netbeans.org/schema/imported"
    elementFormDefault="qualified">
    <xsd:group name="NameGroup">
        <xsd:sequence>
            <xsd:element name="first" type="xsd:string" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="middle" type="xsd:string" minOccurs="0" maxOccurs="1"/>
            <xsd:element name="last" type="xsd:string"/>
        </xsd:sequence>
    </xsd:group>
    <xsd:complexType name="NameType">
        <xsd:group ref="target:NameGroup"/>
        <xsd:attribute name="title" type="xsd:string"/>
    </xsd:complexType>
    <xsd:element name="name" type="target:NameType"/>
</xsd:schema>
```

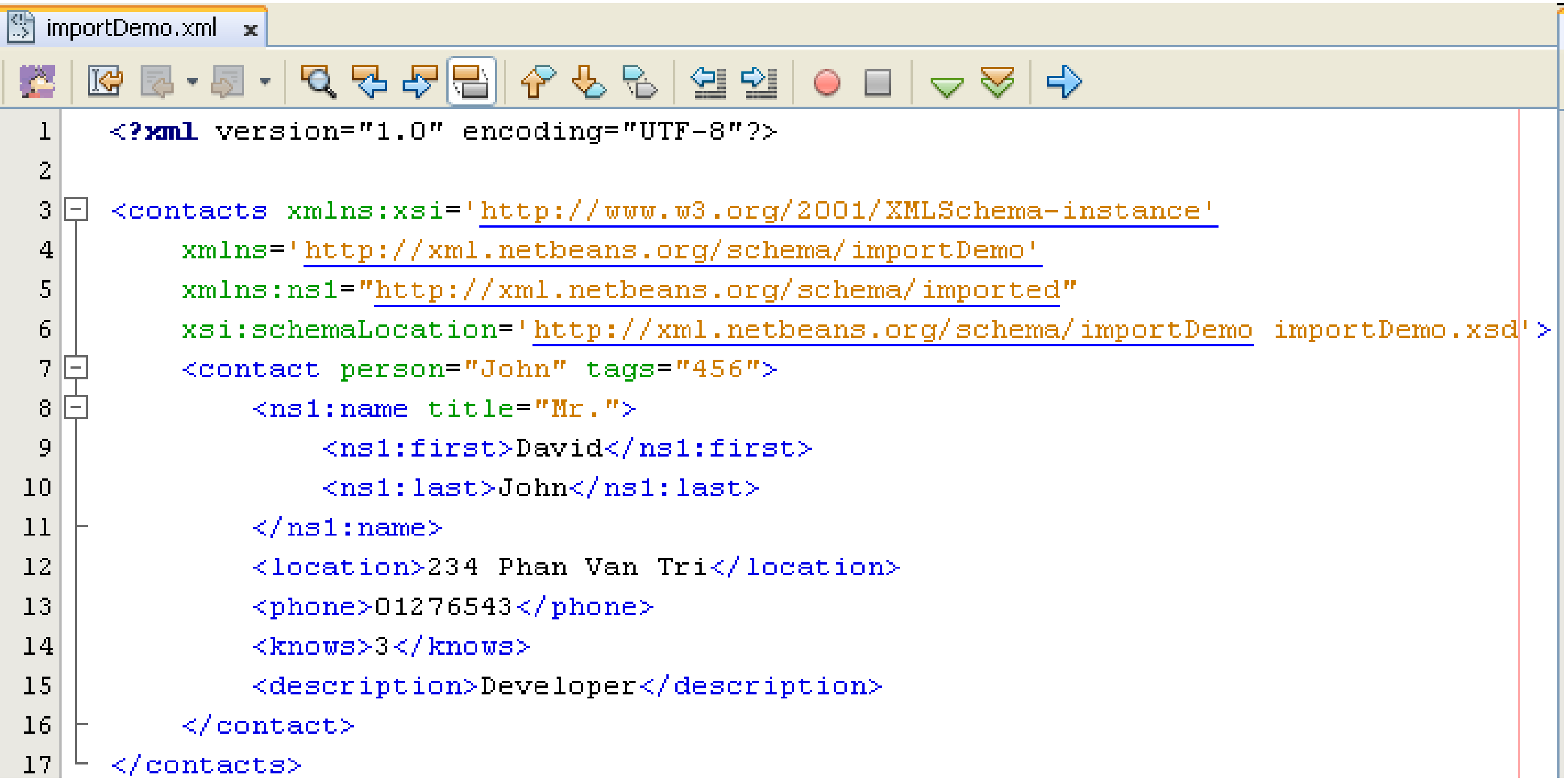
Schemas

Creating schema from multiple documents



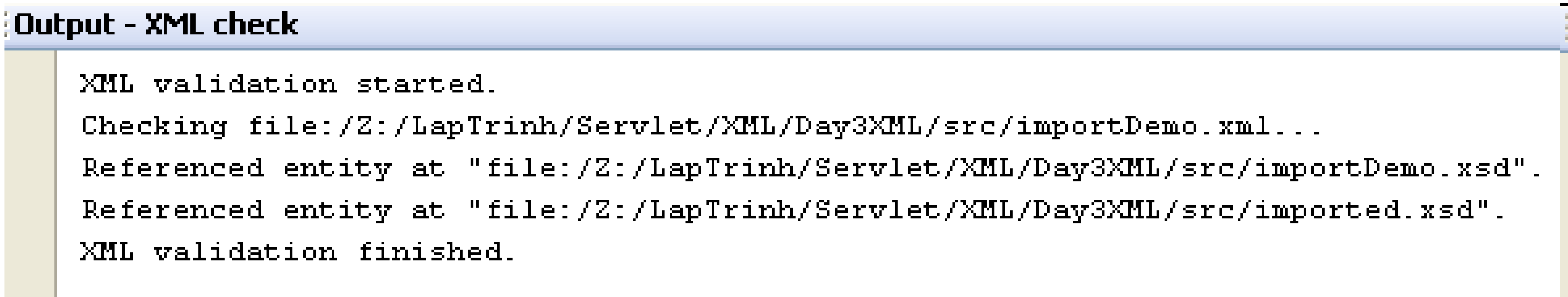
```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://xml.netbeans.org/schema/importDemo"
            xmlns="http://xml.netbeans.org/schema/importDemo"
            xmlns:target="http://xml.netbeans.org/schema/imported"
            elementFormDefault="qualified">
    <xsd:import namespace="http://xml.netbeans.org/schema/imported" schemaLocation="imported.xsd"/>
    <xsd:element name="contacts">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="contact" minOccurs="0" maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element ref="target:name"/>
                            <xsd:element name="location" type="xsd:string"/>
                            <xsd:element name="phone" type="xsd:string"/>
                            <xsd:element name="knows" type="xsd:string"/>
                            <xsd:element name="description" type="xsd:string"/>
                        </xsd:sequence>
                        <xsd:attribute name="person" type="xsd:string"/>
                        <xsd:attribute name="tags" type="xsd:token"/>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas
Creating schema from multiple documents



importDemo.xml

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

<contacts xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/importDemo'
    xmlns:ns1="http://xml.netbeans.org/schema/imported"
    xsi:schemaLocation='http://xml.netbeans.org/schema/importDemo importDemo.xsd'>
    <contact person="John" tags="456">
        <ns1:name title="Mr.">
            <ns1:first>David</ns1:first>
            <ns1:last>John</ns1:last>
        </ns1:name>
        <location>234 Phan Van Tri</location>
        <phone>01276543</phone>
        <knows>3</knows>
        <description>Developer</description>
    </contact>
</contacts>
```

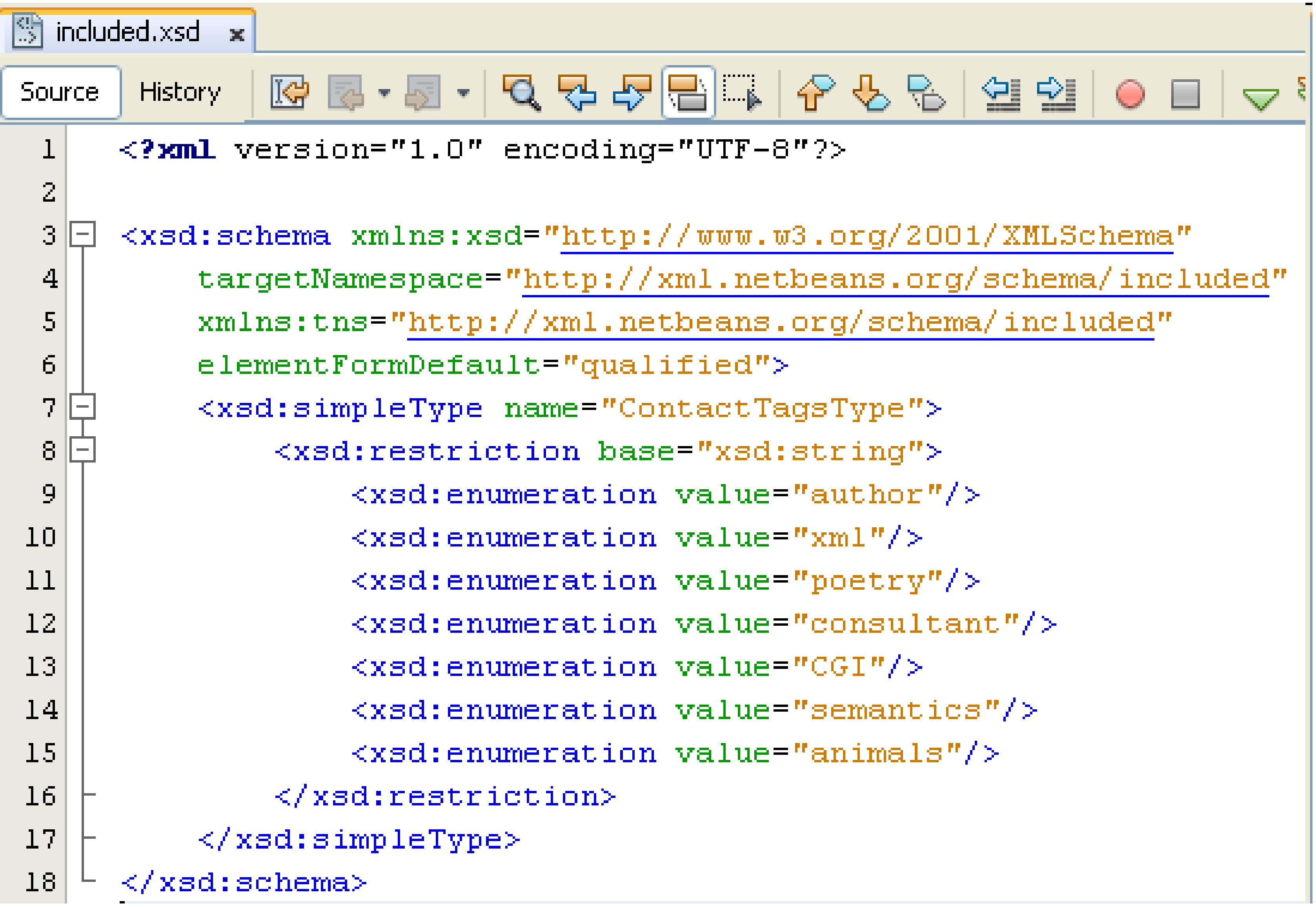


Output - XML check

```
XML validation started.
Checking file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/importDemo.xml...
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/importDemo.xsd".
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/imported.xsd".
XML validation finished.
```

Schemas

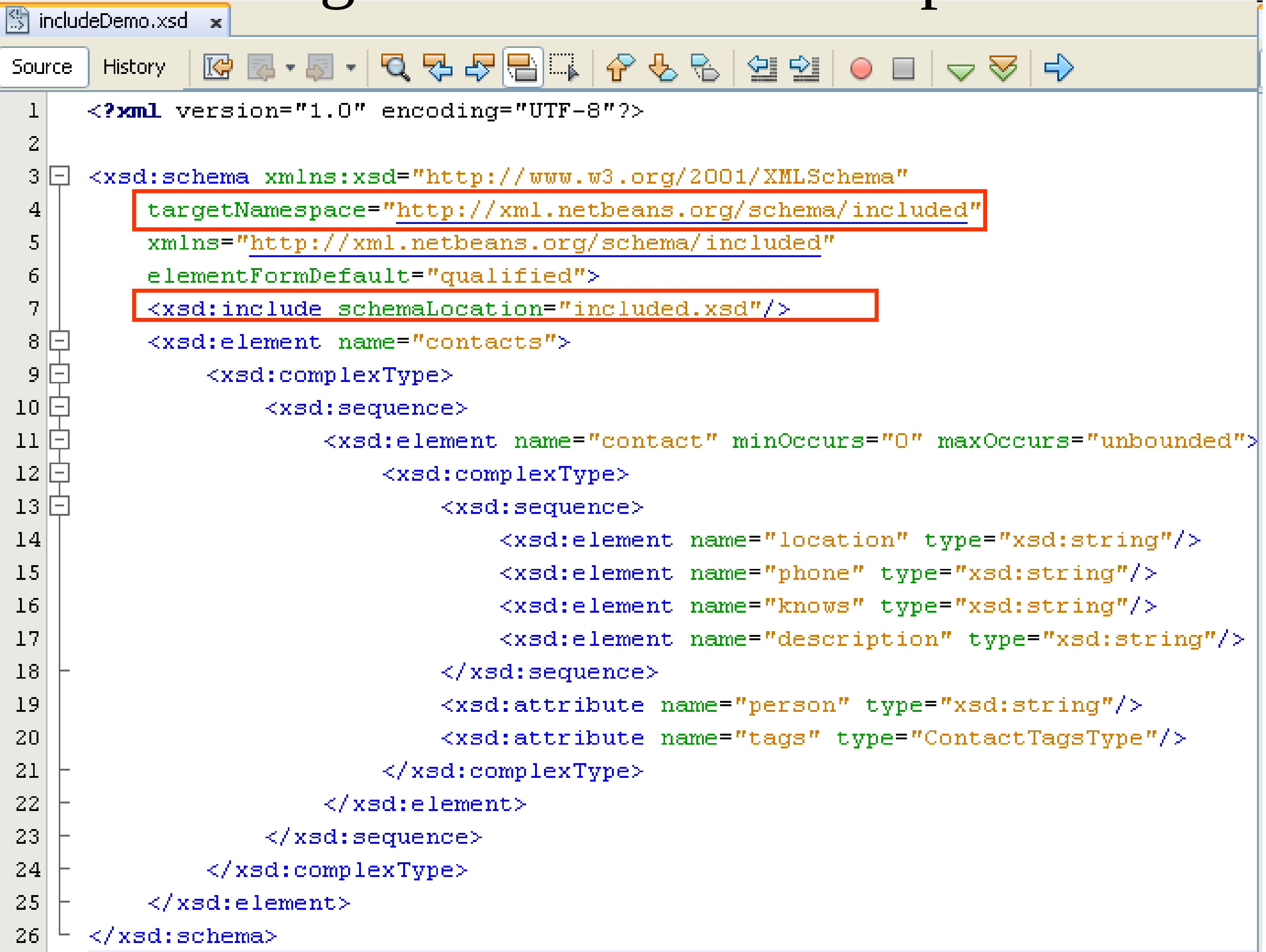
Creating schema from multiple documents



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/included"
    xmlns:tns="http://xml.netbeans.org/schema/included"
    elementFormDefault="qualified">
    <xsd:simpleType name="ContactTagsType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="author"/>
            <xsd:enumeration value="xml"/>
            <xsd:enumeration value="poetry"/>
            <xsd:enumeration value="consultant"/>
            <xsd:enumeration value="CGI"/>
            <xsd:enumeration value="semantics"/>
            <xsd:enumeration value="animals"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

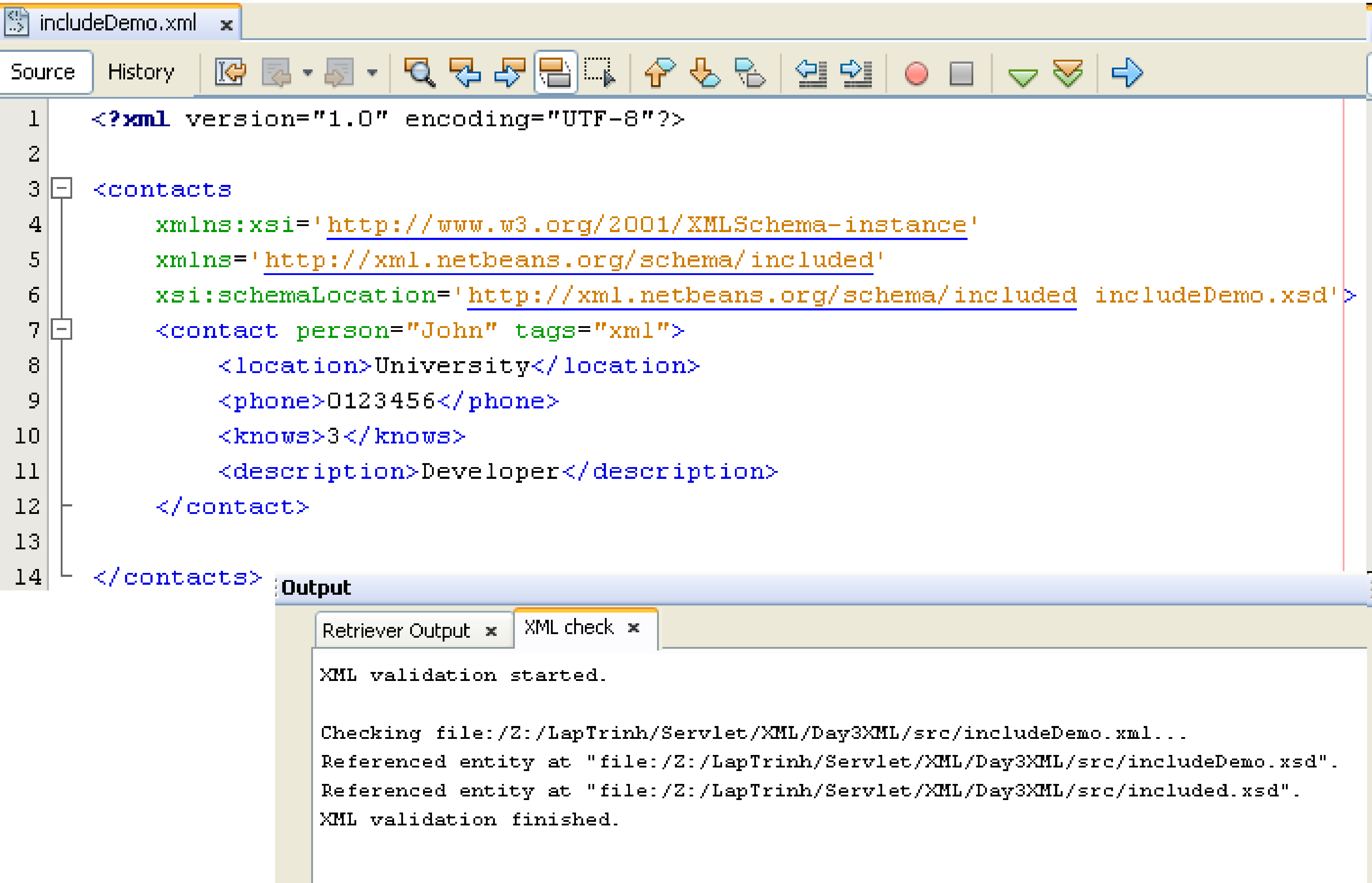
Schemas

Creating schema from multiple documents



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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://xml.netbeans.org/schema/included"
    xmlns="http://xml.netbeans.org/schema/included"
    elementFormDefault="qualified">
    <xsd:include schemaLocation="included.xsd"/>
    <xsd:element name="contacts">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="contact" minOccurs="0" maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="location" type="xsd:string"/>
                            <xsd:element name="phone" type="xsd:string"/>
                            <xsd:element name="knows" type="xsd:string"/>
                            <xsd:element name="description" type="xsd:string"/>
                        </xsd:sequence>
                        <xsd:attribute name="person" type="xsd:string"/>
                        <xsd:attribute name="tags" type="ContactTagsType"/>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Schemas
Creating schema from multiple documents



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

<contacts
    xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
    xmlns='http://xml.netbeans.org/schema/included'
    xsi:schemaLocation='http://xml.netbeans.org/schema/included includeDemo.xsd'>
    <contact person="John" tags="xml">
        <location>University</location>
        <phone>0123456</phone>
        <knows>3</knows>
        <description>Developer</description>
    </contact>

</contacts>
```

```
XML validation started.

Checking file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/includeDemo.xml...
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/includeDemo.xsd".
Referenced entity at "file:/Z:/LapTrinh/Servlet/XML/Day3XML/src/included.xsd".
XML validation finished.
```

Summary

- Introduction
- XSD Elements and Attributes
- XSD Data Types
- XSD Facets
- XSD Indicators
- Extending XSD

Q&A

Next Lecture

- Cascading Style Sheet (CSS)
- Extensible Style Language (XSL)