

# Лабораторная работа №4

## Схема данных

# Модели данных

- Иерархическая
- Сетевая
- Реляционная
- Объектная
- XML

# Иерархическая модель

Отдел

010	05	1000000
-----	----	---------

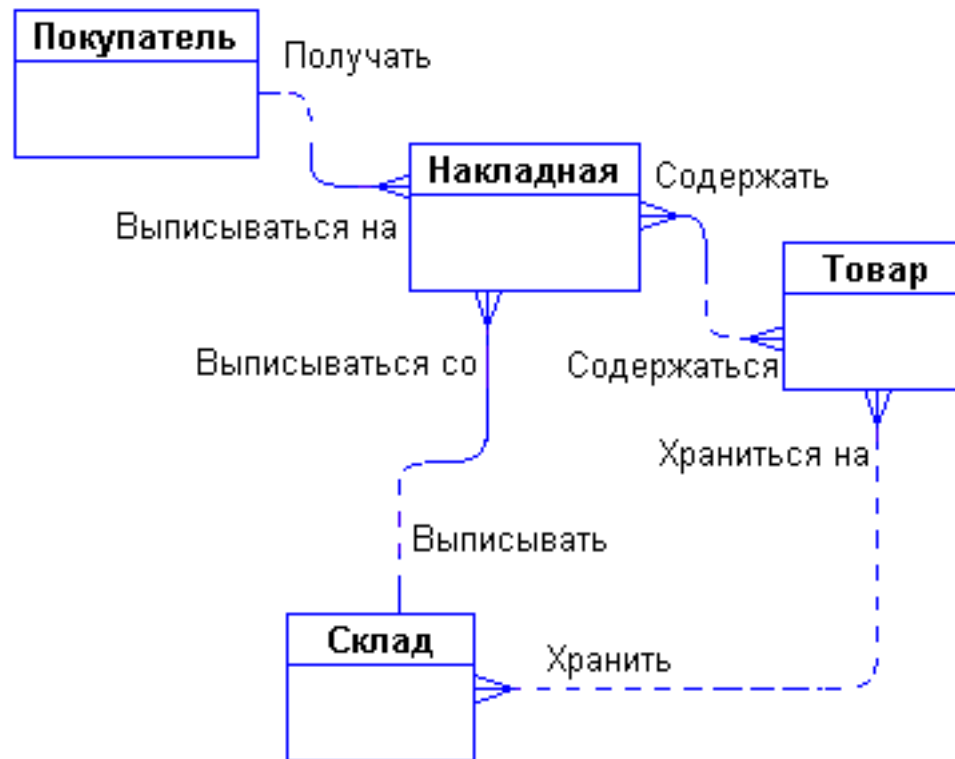
Сотрудники

84	Иванов	100000
	Сидоров	100010

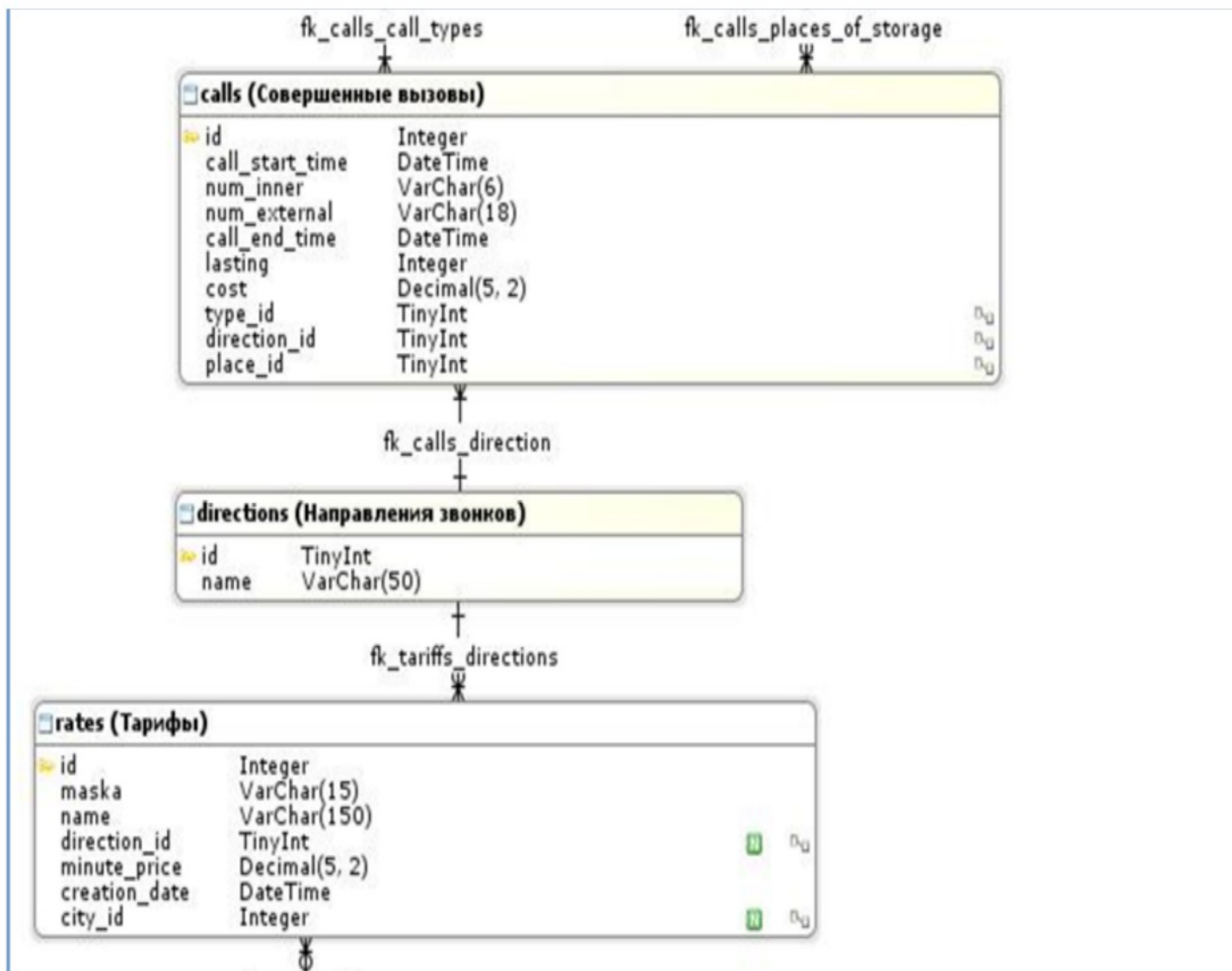
# Сетевая модель



# Реляционная модель



# Схема БД

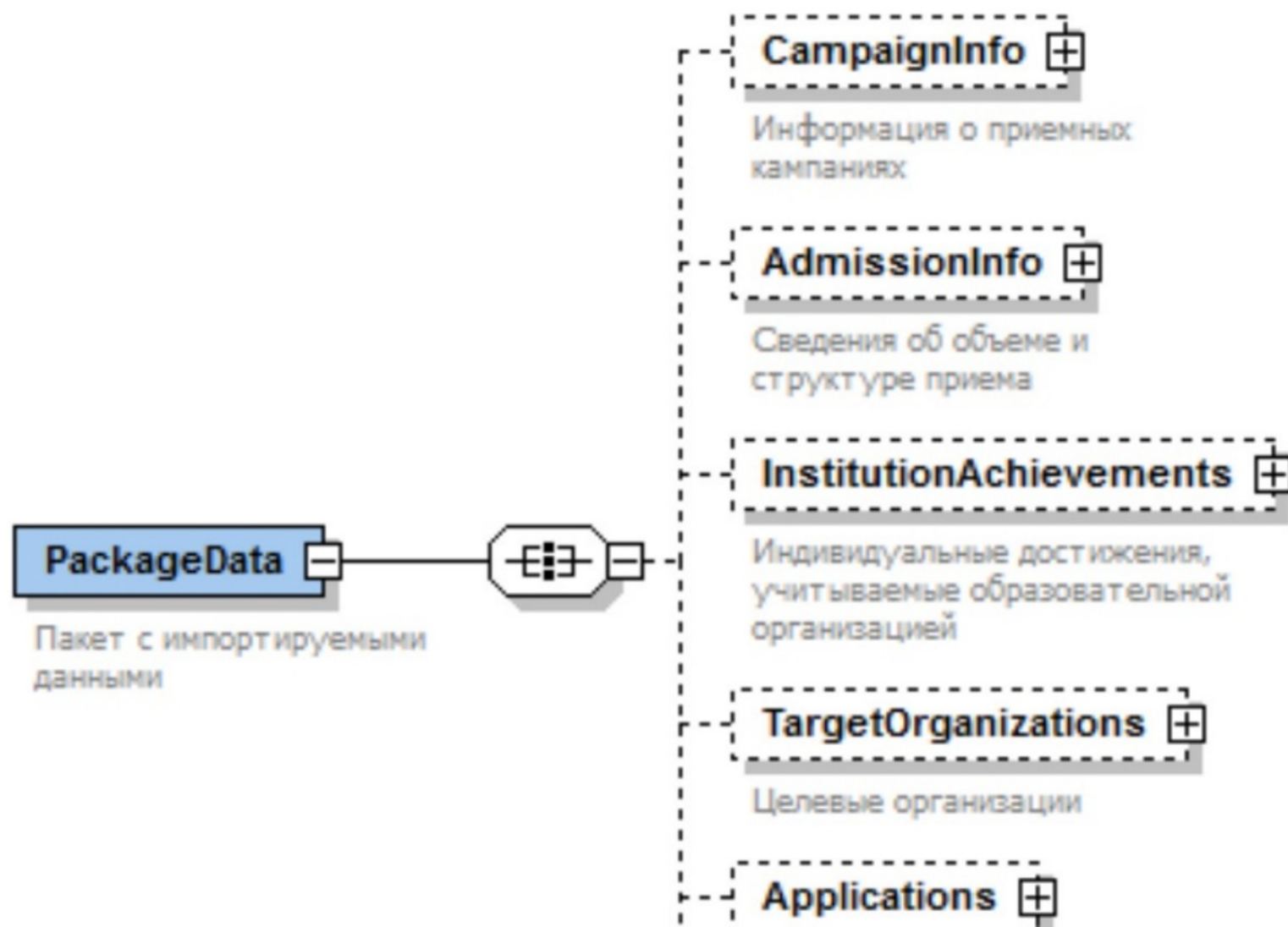


# XML

```
запрос-авторизация.xsd ✖
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:element name="AuthData">
    <xs:annotation>
      <xs:documentation>Блок авторизации</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:all>
        <xs:element name="Login" minOccurs="1">
          <xs:annotation>
            <xs:documentation>Номер заявления ОУ</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="50" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Pass" minOccurs="1">
          <xs:annotation>
            <xs:documentation>Пароль</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="50" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="InstitutionID" minOccurs="0" type="xs:unsignedInt">
          <xs:annotation>
```

# XML

Структура элемента «PackageData» (Пакет с импортируемыми данными):





# XML Schema

XML Schema используется для определения содержимого и структуры документов. В чем-то она схожа с DTD. Но есть усовершенствования: она позволяет определить наборы символов и их взаимосвязи, обязательность и необязательность значения, множественность.

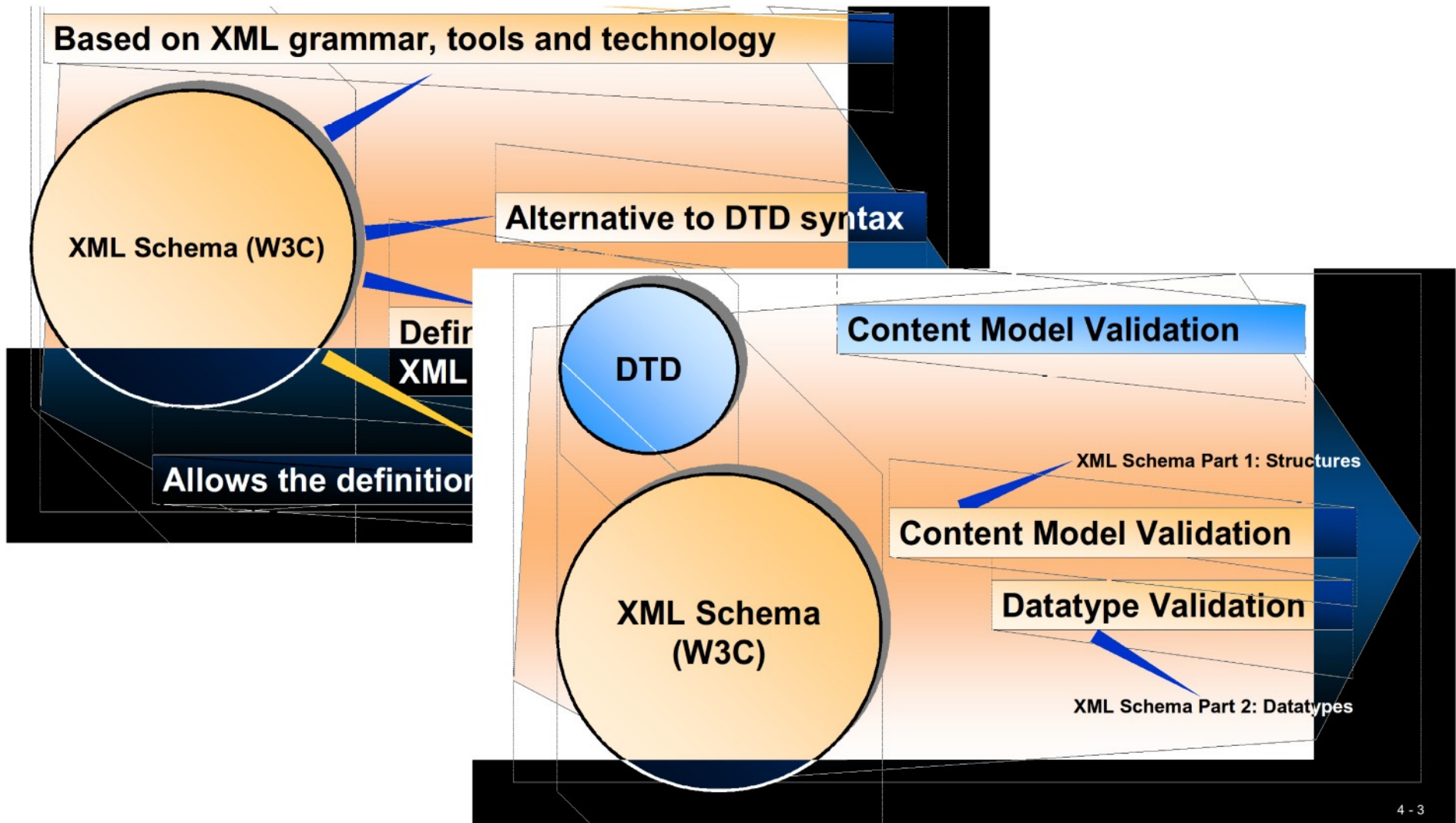
XML-документы делятся на совместимые по схеме и несовместимые по схеме. Несовместимый по схеме XML-документ при этом может оставаться совместимым по типу.

# XML Schema

Поддержка типов данных обеспечивает:

- более легкое описание возможное содержание документа;
- более легкую проверку корректности данных;
- более легкую работу с данными из базы данных;
- более легкое определение области значений данных (ограничения данных);
- более легкое определение шаблонов данных (форматов данных);
- более легкое преобразование данных между различными типами данных.

# XML Schema



# XML Schema

Когда данные посылаются от отправителя к получателю очень важно то, что они имеют одинаковое представление о содержимом. Посредством XML Схем отправитель может описать данные таким образом, что получатель будет понимать их.

Например, дата представленная как 1999-03-11 может в ряде стран интерпретироваться как 3 ноября, в других странах как 11 марта. Однако, XML element с типом данных: `<date type="date">1999-03-11</date>` позволит однозначно понимать содержимое, т.к. тип данных XML даты описывается в формате `ССУУ-ММ-ДД`.

# XML Schema

Простой пример XML Schema, расположенный в файле "country.xsd", описывает данные о стране:

```
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="страна" type="страна"/>
  <xs:complexType name="страна">
    <xs:sequence>
      <xs:element name="название" type="xs:string"/>
      <xs:element name="население" type="xs:decimal"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Пример документа, отвечающего этой схеме:

```
<страна
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="country.xsd">
  <название>Франция</название>
  <население>59.7</население>
</страна>
```

# Включение схемы

Если не определено пространство имен

```
<building xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="task05.xsd">
```

Если есть пространство имен

```
<note
xmlns="http://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3schools.com note.xsd">
```

# Схема

```
<?xml version="1.0"?>
```

```
<xs:schema  
xmlns:xs="http://www.w3.org/2001/XMLSchema"  
targetNamespace="http://www.w3schools.com"  
xmlns="http://www.w3schools.com">
```

```
...
```

```
...
```

```
</xs:schema>
```

# Простой элемент

```
<xs:element name="xxx" type="yyy"/>
```

```
<lastname>Refsnes</lastname>
```

```
<age>36</age>
```

```
<dateborn>1970-03-27</dateborn>
```

```
<xs:element name="lastname" type="xs:string"/>
```

```
<xs:element name="age" type="xs:integer"/>
```

```
<xs:element name="dateborn" type="xs:date"/>
```

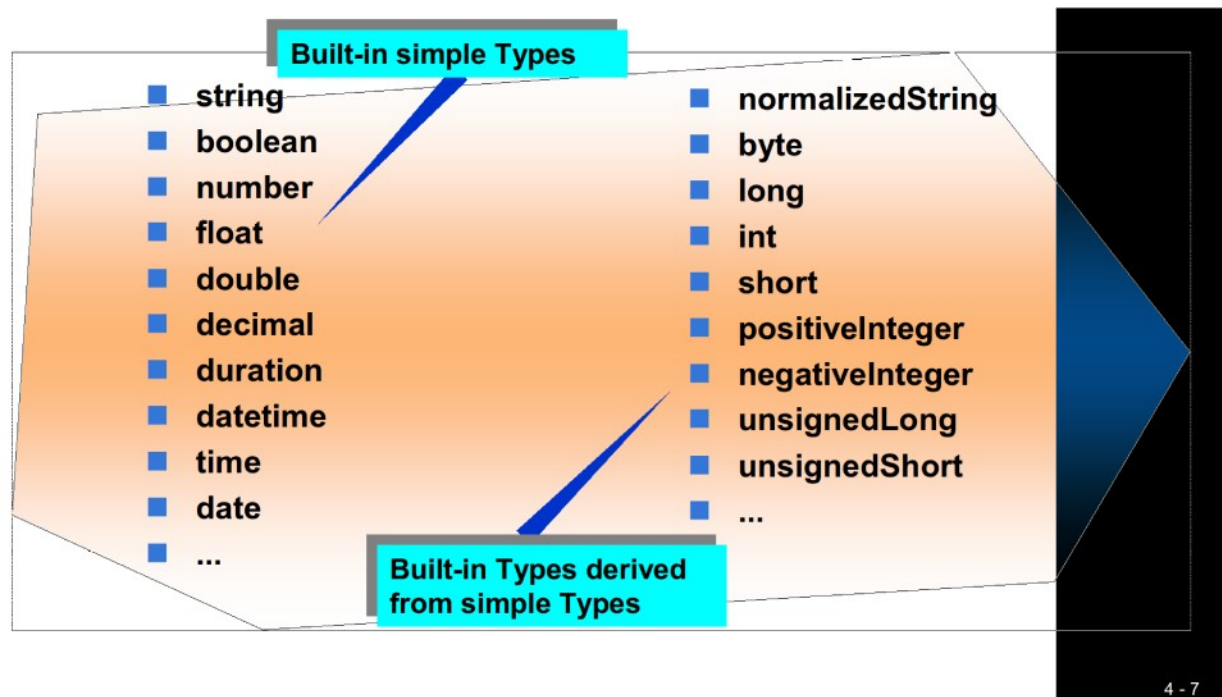


# Значения

**<xs:element name="color" type="xs:string" default="red"/>**

**<xs:element name="color" type="xs:string" fixed="red"/>**

# Типы данных



# Атрибуты

**<xs:attribute name="xxx" type="yyy"/>**

<lastname lang="EN">Smith</lastname>

<xs:attribute name="lang" type="xs:string"/>

<xs:attribute name="lang" type="xs:string" default="EN"/>

<xs:attribute name="lang" type="xs:string" fixed="EN"/>

<xs:attribute name="lang" type="xs:string" use="required"/>

# Ограничения для простых типов

- **xsd:restriction**
  - ◆ **xsd:maxLength, xsd:minLength**
  - ◆ **xsd:minInclusive, xsd:maxInclusive**
  - ◆ **xsd:totalDigits, xsd:fractionDigits ...**
  - ◆ **xsd:pattern**
- **xsd:union**
- **xsd:list**
- **...**



Facets

# Ограничения

```
<xs:element name="age">  
  <xs:simpleType>  
    <xs:restriction base="xs:integer">  
      <xs:minInclusive value="0"/>  
      <xs:maxInclusive value="120"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```

# Ограничения

```
<xs:element name="car">  
  <xs:simpleType>  
    <xs:restriction base="xs:string">  
      <xs:enumeration value="Audi"/>  
      <xs:enumeration value="Golf"/>  
      <xs:enumeration value="BMW"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```

# Ограничения

```
<xs:element name="initials">  
  <xs:simpleType>  
    <xs:restriction base="xs:string">  
      <xs:pattern value="[a-zA-Z][a-zA-Z][a-zA-Z]"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```

# СЛОЖНЫЕ ТИПЫ

```
<employee>  
  <firstname>John</firstname>  
  <lastname>Smith</lastname>  
</employee>
```

```
<xs:element name="employee">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="firstname" type="xs:string"/>  
      <xs:element name="lastname" type="xs:string"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```



# XSD индикаторы

```
<xs:element name="person">  
  <xs:complexType>  
    <xs:all> - любой порядок для "детей", но каждый может появиться только раз  
      <xs:element name="firstname" type="xs:string"/>  
      <xs:element name="lastname" type="xs:string"/>  
    </xs:all>  
  </xs:complexType>  
</xs:element>
```

# XSD индикаторы

```
<xs:element name="person">  
  <xs:complexType>  
    <xs:choice> - может появиться какой-то один из "детей"  
      <xs:element name="employee" type="employee"/>  
      <xs:element name="member" type="member"/>  
    </xs:choice>  
  </xs:complexType>  
</xs:element>
```

# XSD индикаторы

```
<xs:element name="person">  
  <xs:complexType>  
    <xs:sequence> - определенная последовательность "детей"  
      <xs:element name="firstname" type="xs:string"/>  
      <xs:element name="lastname" type="xs:string"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

# XSD индикаторы

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="full_name" type="xs:string"/>
      <xs:element name="child_name" type="xs:string"
maxOccurs="10"/>
      <xs:element name="child_name" type="xs:string"
maxOccurs="10" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

# XSD индикаторы

Определение группы элементов (all, sequence, choice)

```
<xs:group name="groupname">
```

...

```
</xs:group>
```

Определение группы атрибутов

```
<xs:attributeGroup name="groupname">
```

...

```
</xs:attributeGroup>
```

# СЛОЖНЫЕ ТИПЫ

```
<xs:element name="employee" type="personinfo"/>  
<xs:element name="student" type="personinfo"/>  
<xs:element name="member" type="personinfo"/>
```

```
<xs:complexType name="personinfo">  
  <xs:sequence>  
    <xs:element name="firstname" type="xs:string"/>  
    <xs:element name="lastname" type="xs:string"/>  
  </xs:sequence>  
</xs:complexType>
```

# СЛОЖНЫЕ ТИПЫ

```
<xs:element name="employee" type="fullpersoninfo"/>
<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="fullpersoninfo">
  <xs:complexContent>
    <xs:extension base="personinfo">
      <xs:sequence>
        <xs:element name="address" type="xs:string"/>
        <xs:element name="city" type="xs:string"/>
        <xs:element name="country" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

# Пустые элементы

```
<product prodid="1345" />
```

```
<xs:element name="product">  
  <xs:complexType>  
    <xs:attribute name="prodid" type="xs:positiveInteger"/>  
  </xs:complexType>  
</xs:element>
```



# Сложные элементы

```
<xs:element name="somename">  
  <xs:complexType>  
    <xs:simpleContent>  
      <xs:extension base="basetype">  
        ....  
        ....  
      </xs:extension>  
    </xs:simpleContent>  
  </xs:complexType>  
</xs:element>
```

# Сложные элементы

```
<xs:element name="somename">  
  <xs:complexType>  
    <xs:simpleContent>  
      <xs:restriction base="basetype">  
        ....  
        ....  
      </xs:restriction>  
    </xs:simpleContent>  
  </xs:complexType>  
</xs:element>
```

# Смешанное содержимое

<letter>

Dear Mr.<name>John Smith</name>.

Your order <orderid>1032</orderid>

will be shipped on <shipdate>2001-07-13</shipdate>.

</letter>

```
<xs:element name="letter" type="lettertype"/>
```

```
<xs:complexType name="lettertype" mixed="true">
```

```
  <xs:sequence>
```

```
    <xs:element name="name" type="xs:string"/>
```

```
    <xs:element name="orderid" type="xs:positiveInteger"/>
```

```
    <xs:element name="shipdate" type="xs:date"/>
```

```
  </xs:sequence>
```

```
</xs:complexType>
```

# Any

`<xs:any minOccurs="0"/>`

`<xs:anyAttribute/>`

# Пример

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="book">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="title" type="xsd:string"/>
        <xsd:element name="author" type="xsd:string"/>
        <xsd:element name="character" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="name" type="xsd:string"/>
              <xsd:element name="friend-of" type="xsd:string"
                minOccurs="0" maxOccurs="unbounded"/>
              <xsd:element name="since" type="xsd:date"/>
              <xsd:element name="qualification" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="isbn" type="xsd:string"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Composer

Element of type  
complexType

# Задание 5

- Создать в файл task5.xsd и описать в нем схему документа для первой задачи task01.xml;
- Проверить документ на соответствие схеме.

# ИСТОЧНИКИ

1. Эдди С.Э. XML: справочник — СПб.: Питер, 2000
2. Материалы курса IT 581
3. Материалы курса по XML компьютерного центра обучения "Специалист" при МГТУ им. Н.Э. Баумана
4. Материалы W3Schools (<http://www.w3schools.com/xml/>)
5. Википедия (<http://ru.wikipedia.org>)
6. Наварро Э. XHTML: учебный курс — СПб.: Питер, 2001