

XML валидация чрез XML Schema

Цели на упражнението:

1. Създаване на XML схема
2. Валидация на XML документ с XML Schema
3. Дефиниране на прости и комплексни типове
4. Задаване на ограничения върху предефинираните типове
5. Използване на разширени типове и предефиниране на типове
6. Използване на анотации, нотации, регулярни изрази
7. Импортиране на XML схеми от други пространства от имена

Средства за XML валидация чрез XML Schema:

За реализация на това упражнения могат да бъдат използвани някои от следните инструменти:

[XML Validator](#)

[XML Validator Online](#)

[CoreFiling XML Schema Validator](#)

**Задача 1: Превърнете дадения по-долу DTD документ в XML Schema.
Създайте XML екземпляр на тази схема и я валидирайте.**

```
<!ELEMENT collection (description,recipe*)>
<!ELEMENT description ANY>
<!ELEMENT recipe (title,ingredient*,preparation,comment?,nutrition)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT ingredient (preparation?)>
<!-- ingredient name CDATA #REQUIRED
      amount CDATA #IMPLIED
      unit CDATA #IMPLIED -->
<!ELEMENT preparation (step*)>
<!ELEMENT step (#PCDATA)>
<!ELEMENT comment (#PCDATA)>
<!ELEMENT nutrition EMPTY>
<!-- nutrition protein CDATA #REQUIRED
      carbohydrates CDATA #REQUIRED
      fat CDATA #REQUIRED -->
```

calories CDATA #REQUIRED
alcohol CDATA #IMPLIED>

Упътване

Решение

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="collection">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="description"/>
        <xs:element ref="recipe" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="description">
    <xs:complexType mixed="true">
      <xs:sequence>
        <xs:any minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="recipe">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="title"/>
        <xs:element ref="ingredient" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="preparation"/>
        <xs:element ref="comment" minOccurs="0"/>
        <xs:element ref="nutrition"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="title">
    <xs:complexType mixed="true"/>
  </xs:element>
  <xs:element name="ingredient">
    <xs:complexType>
      <xs:sequence minOccurs="0">
        <xs:element ref="ingredient" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="preparation"/>
      </xs:sequence>
      <xs:attribute name="name" use="required"/>
      <xs:attribute name="amount"/>
      <xs:attribute name="unit"/>
    </xs:complexType>
  </xs:element>
```

```

<xs:element name="preparation">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="step" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="step">
  <xs:complexType mixed="true"/>
</xs:element>
<xs:element name="comment">
  <xs:complexType mixed="true"/>
</xs:element>
<xs:element name="nutrition">
  <xs:complexType>
    <xs:attribute name="protein" use="required"/>
    <xs:attribute name="carbohydrates" use="required"/>
    <xs:attribute name="fat" use="required"/>
    <xs:attribute name="calories" use="required"/>
    <xs:attribute name="alcohol"/>
  </xs:complexType>
</xs:element>
</xs:schema>

```

```

-----
<?xml version="1.0" encoding="UTF-8"?>
<collection xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XXXX.xsd">
  <description>description.....</description>
  <recipe>
    <title>Collection 1</title>
    <ingredient name="all">
      <preparation>
        <step>Step 1</step>
        <step>Step 2</step>
        <step>Step 3</step>
      </preparation>
    </ingredient>
    <preparation>
      <step>Stage 1</step>
      <step>Stage 2</step>
      <step>Stage 3</step>
    </preparation>
    <comment>No comments....</comment>
    <nutrition fat="12%" protein="10%" calories="123" carbohydrates="N.A."/>
  </recipe>
</collection>

```

Задача 2: Даденият по-долу XML документ описва типовете сметки, които поддържа една примерна банка, нейните клиенти и сметките, които те имат. За този XML документ създайте XML Schema, която изпълнява следните условия:

- Редът на срещане на под-елементите на bank (accounts, customers и customer_accounts) и accounts (saving_accounts и checking_accounts) няма значение
- Всяка сметка има уникален идентификатор
- Всеки клиент има уникален идентификатор
- Балансът на сметката не може да бъде по-малък от -5000 - за това условие използвайте рестрикция на съществуващите предефинирани типове
- Атрибутът c_id реферира към съответния клиент, а ac_id към съответната му сметка
- Дефинирайте елементите customers и saving_account като комплексен глобален тип, а елемента accounts като комплексен локален тип

```
<?xml version="1.0" encoding="UTF-8"?>
<bank>
  <accounts>
    <saving_accounts>
      <saving_account id="a1" interest="0.03">
        <balance>2500</balance>
      </saving_account>
      <saving_account id="a2" interest="0.03">
        <balance>15075</balance>
      </saving_account>
    </saving_accounts>
    <checking_accounts>
      <checking_account id="a3">
        <balance>4025</balance>
      </checking_account>
      <checking_account id="a4">
        <balance>-125</balance>
      </checking_account>
      <checking_account id="a5">
        <balance>325</balance>
      </checking_account>
    </checking_accounts>
  </accounts>
  <customers>
    <customer id="c1">
      <name>Ben Richerdson</name>
      <address>Park Drive 2</address>
    </customer>
    <customer id="c2">
      <name>Marc Wretcher</name>
      <address>Mill Drive 75</address>
    </customer>
    <customer id="c3">
```

```

    <name>Angel Steady</name>
    <address>Lake Sight 15</address>
  </customer>
</customers>
<customer_accounts>
  <customer_account c_id="c1" ac_id="a2"/>
  <customer_account c_id="c1" ac_id="a3"/>
  <customer_account c_id="c2" ac_id="a4"/>
  <customer_account c_id="c3" ac_id="a1"/>
  <customer_account c_id="c3" ac_id="a5"/>
</customer_accounts>
</bank>

```

Решение

```

<?xml version="1.0" encoding="UTF-8"?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bank">
    <xs:complexType>
      <xs:all>
        <xs:element name="accounts" minOccurs="1">
          <xs:complexType>
            <xs:all>
              <xs:element ref="saving_accounts"/>
              <xs:element ref="checking_accounts"/>
            </xs:all>
          </xs:complexType>
        </xs:element>
        <xs:element ref="customers" minOccurs="1"/>
        <xs:element ref="customer_accounts" minOccurs="1"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="name" type="xs:string"/>
  <xs:element name="address" type="xs:string"/>
  <xs:element name="balance">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="-5000"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="customer">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="name"/>
        <xs:element ref="address"/>
      </xs:sequence>
      <xs:attribute name="id" use="required" type="xs:ID"/>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

```
</xs:element>
<xs:complexType name="customers">
  <xs:sequence>
    <xs:element ref="customer" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="customers" type="customers"/>
<xs:element name="saving_account">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="balance"/>
    </xs:sequence>
    <xs:attribute name="id" use="required" type="xs:ID"/>
    <xs:attribute name="interest" type="xs:decimal"/>
  </xs:complexType>
</xs:element>
<xs:element name="checking_account">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="balance"/>
    </xs:sequence>
    <xs:attribute name="id" use="required" type="xs:ID"/>
  </xs:complexType>
</xs:element>
<xs:element name="customer_account">
  <xs:complexType>
    <xs:attribute name="c_id" use="required" type="xs:IDREF"/>
    <xs:attribute name="ac_id" use="required" type="xs:IDREF"/>
  </xs:complexType>
</xs:element>
<xs:element name="saving_accounts">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="saving_account" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checking_accounts">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="checking_account" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="customer_accounts">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="customer_account" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
</xs:complexType>
</xs:element>
</xs:schema>
```

Задача 3: В XML схемата от задача 2 направете следните промени:

- Дефинирайте два прости типа, задаващи горна и долна граница на стойността на елемента balance. След това променете дефиницията на елемента balance, като го представите като обединение на тези два прости типа
- Използвайки разширени типове на XML Schema, създайте нов тип customerExt, който разширява дефиницията на типа customer като добавя нов негов под-елемент contacts, който от своя страна се състои от 2 под-елемента - email и telephone. Използвайте новия тип customerExt вместо customer и запишете новата XML схема под името bank.xsd
- Създайте нова XML схема, която предефинира типа balance и customerExt от външната за нея XML схема - bank.xsd. Новият тип balance не трябва да бъде по-малък от -5200, а новият тип customerExt съдържа допълнително нов под-елемент image от тип base64Binary, който има един атрибут src от тип string. Запишете новата XML схема под името bankExt.xsd
- Добавете няколко анотации в по-горе създадената XML схема (bank.xsd) към избрани от вас комплексните типове, описващи тяхното предназначение

Упътване

Решение

bank.xsd:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      This document defines an example of XML Schema.
    </xs:documentation>
  </xs:annotation>
  <xs:element name="bank">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        This element is the root.
      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:all>
        <xs:element name="accounts" minOccurs="1">
          <xs:complexType>
            <xs:all>
              <xs:element ref="saving_accounts"/>
              <xs:element ref="checking_accounts"/>
            </xs:all>
          </xs:complexType>
        </xs:element>
```

```

        <xs:element ref="customers" minOccurs="1"/>
        <xs:element ref="customer_accounts" minOccurs="1"/>
    </xs:all>
</xs:complexType>
</xs:element>
<xs:element name="name" type="xs:string"/>
<xs:element name="address" type="address"/>
<xs:simpleType name="address">
    <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:element name="balance" type="balance"/>
<xs:simpleType name="balance">
    <xs:union memberTypes="minBalance maxBalance"/>
</xs:simpleType>
<xs:simpleType name="maxBalance">
    <xs:restriction base="xs:positiveInteger">
        <xs:maxInclusive value="1000000"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="minBalance">
    <xs:restriction base="xs:negativeInteger">
        <xs:minInclusive value="-5500"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="customer">
    <xs:sequence>
        <xs:element ref="name"/>
        <xs:element ref="address"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
</xs:complexType>
<xs:element name="customer" type="customer"/>
<xs:complexType name="customerExt">
    <xs:complexContent>
        <xs:extension base="customer">
            <xs:sequence>
                <xs:element ref="contacts" minOccurs="1"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:element name="customerExt" type="customerExt"/>
<xs:complexType name="contacts">
    <xs:sequence>
        <xs:element ref="email"/>
        <xs:element ref="telephone"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="contacts" type="contacts"/>

```



```
<xs:element name="email" type="xs:string"/>
<xs:element name="telephone" type="xs:string"/>
<xs:complexType name="customers">
  <xs:sequence>
    <xs:element ref="customerExt" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="customers" type="customers"/>
<xs:element name="saving_account">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="balance"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
    <xs:attribute name="interest" type="xs:decimal"/>
  </xs:complexType>
</xs:element>
<xs:element name="checking_account">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="balance"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="customer_account">
  <xs:complexType>
    <xs:attribute name="c_id" type="xs:IDREF" use="required"/>
    <xs:attribute name="ac_id" type="xs:IDREF" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="saving_accounts">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="saving_account" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checking_accounts">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="checking_account" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="customer_accounts">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="customer_account" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

bankExt.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:redefine schemaLocation="bank.xsd">
    <xs:simpleType name="minBalance">
      <xs:restriction base="minBalance">
        <xs:minInclusive value="-5200"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="customerExt">
      <xs:complexContent>
        <xs:extension base="customerExt">
          <xs:sequence>
            <xs:element name="image" type="image"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:redefine>
  <xs:complexType name="image">
    <xs:simpleContent>
      <xs:extension base="xs:base64Binary">
        <xs:attribute name="src" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:schema>
```

Задача 4: Редактирайте XML схемата от задача 3 (bank.xsd), като добавите елемент postalCode към комплексния тип customer и включите регулярен израз за:

- пощенски код (четири цифрено число, например:1000)
- телефон (например в следния формат:+359-02-989-14-04)
- електронна поща

Упътване

Решение

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      This document defines an example of XML Schema.
    </xs:documentation>
  </xs:annotation>
  <xs:element name="bank">
    <xs:annotation>
      <xs:documentation xml:lang="en">
        This element is the root.
      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:all>
        <xs:element name="accounts" minOccurs="1">
          <xs:complexType>
            <xs:all>
              <xs:element ref="saving_accounts"/>
              <xs:element ref="checking_accounts"/>
            </xs:all>
          </xs:complexType>
        </xs:element>
        <xs:element ref="customers" minOccurs="1"/>
        <xs:element ref="customer_accounts" minOccurs="1"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="name" type="xs:string"/>
  <xs:element name="address" type="address"/>
  <xs:simpleType name="address">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:element name="balance" type="balance"/>
  <xs:simpleType name="balance">
    <xs:union memberTypes="minBalance maxBalance"/>
  </xs:simpleType>
  <xs:simpleType name="maxBalance">
    <xs:restriction base="xs:positiveInteger">
      <xs:maxInclusive value="1000000"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="minBalance">
    <xs:restriction base="xs:negativeInteger">
      <xs:minInclusive value="-5500"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="customer">
    <xs:sequence>
```

```

    <xs:element ref="name"/>
    <xs:element ref="address"/>
    <xs:element ref="postalCode"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="required"/>
</xs:complexType>
<xs:element name="customer" type="customer"/>
<xs:complexType name="customerExt">
  <xs:complexContent>
    <xs:extension base="customer">
      <xs:sequence>
        <xs:element ref="contacts" minOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="customerExt" type="customerExt"/>
<xs:complexType name="contacts">
  <xs:sequence>
    <xs:element ref="email"/>
    <xs:element ref="telephone"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="contacts" type="contacts"/>
<xs:element name="email" type="email"/>
<xs:element name="postalCode" type="postalCode"/>
<xs:element name="telephone" type="telephone"/>
<xs:complexType name="customers">
  <xs:sequence>
    <xs:element ref="customerExt" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="customers" type="customers"/>
<xs:element name="saving_account">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="balance"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
    <xs:attribute name="interest" type="xs:decimal"/>
  </xs:complexType>
</xs:element>
<xs:element name="checking_account">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="balance"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
  </xs:complexType>

```

```

</xs:element>
<xs:element name="customer_account">
  <xs:complexType>
    <xs:attribute name="c_id" type="xs:IDREF" use="required"/>
    <xs:attribute name="ac_id" type="xs:IDREF" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="saving_accounts">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="saving_account" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="checking_accounts">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="checking_account" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="customer_accounts">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="customer_account" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:simpleType name="postalCode">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]{4}"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="email">
  <xs:restriction base="xs:token">
    <xs:pattern value="[_\\-a-zA-Z0-9\\.\\+]+@[a-zA-Z0-9](\\.?[\\-a-zA-Z0-9]*[a-zA-Z0-9])*/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="telephone">
  <xs:restriction base="xs:string">
    <xs:pattern value="\\+[0-9]{2,3}-[0-9]{2}-[0-9]{3}-[0-9]{2}-[0-9]{2}"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>

```

Задача 5: Редактирайте XML схемата от задача 3 (bankExt.xsd), като добавите към елемента image атрибут type от тип нотация. Създайте XML инстанция

на новата схема и валидирайте.

Упътване

Решение

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:redefine schemaLocation="bank.xsd">
    <xs:simpleType name="minBalance">
      <xs:restriction base="minBalance">
        <xs:minInclusive value="-5200"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="customerExt">
      <xs:complexContent>
        <xs:extension base="customerExt">
          <xs:sequence>
            <xs:element name="image" type="image"/>
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:redefine>
  <xs:notation name="gif" public="image/gif" system="view.exe"/>
  <xs:notation name="jpeg" public="image/jpeg" system="view.exe"/>
  <xs:complexType name="image">
    <xs:simpleContent>
      <xs:extension base="xs:base64Binary">
        <xs:attribute name="src" type="xs:string" use="required"/>
        <xs:attribute name="type" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:NOTATION">
              <xs:enumeration value="gif"/>
              <xs:enumeration value="jpeg"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:schema>
```

Задача 6: Съставете XML Schema, която включва схеми от други пространства от имена. Създайте XML екземляр (инстанция) на тази схема и го валидирайте.

Task6.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Task6_4.xsd">
  <order xmlns="http://example.org/ord" xmlns:prod="http://example.org/prod">
    <number>123ABBCC123</number>
    <items>
      <product xmlns="http://example.org/prod">
        <number prod:id="prod557">557</number>
        <name xmlns="">Short-Sleeved Linen Blouse</name>
        <prod:size system="US-DRESS">10</prod:size>
        <prod:color xmlns:prod="http://example.org/prod2" prod:value="blue"/>
      </product>
    </items>
  </order>
</envelope>
```

Task6_1.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://example.org/prod" xmlns="http://example.org/prod" xmlns:prod2="
http://example.org/prod2" elementFormDefault="qualified">
  <xsd:import namespace="http://example.org/prod2" schemaLocation="Task6_1.xsd"/>
  <xsd:element name="product" type="ProductType"/>
  <xsd:complexType name="ProductType">
    <xsd:sequence>
      <xsd:element name="number" type="ProdNumType"/>
      <xsd:element name="name" type="xsd:string" form="unqualified"/>
      <xsd:element name="size" type="SizeType"/>
      <xsd:element ref="prod2:color"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="ProdNumType">
    <xsd:simpleContent>
      <xsd:extension base="xsd:integer">
        <xsd:attribute name="id" type="xsd:ID" form="qualified" use="required"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="SizeType">
    <xsd:simpleContent>
      <xsd:extension base="xsd:integer">
        <xsd:attribute name="system" type="xsd:string"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
```

```
</xsd:simpleContent>
</xsd:complexType>
</xsd:schema>
```

Task6_2.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://example.org/prod" xmlns="http://example.org/prod"
xmlns:prod2="http://example.org/prod2" elementFormDefault="qualified">
<xsd:import namespace="http://example.org/prod2" schemaLocation="Task6_1.xsd"/>
<xsd:element name="product" type="ProductType"/>
<xsd:complexType name="ProductType">
<xsd:sequence>
<xsd:element name="number" type="ProdNumType"/>
<xsd:element name="name" type="xsd:string" form="unqualified"/>
<xsd:element name="size" type="SizeType"/>
<xsd:element ref="prod2:color"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ProdNumType">
<xsd:simpleContent>
<xsd:extension base="xsd:integer">
<xsd:attribute name="id" type="xsd:ID" form="qualified" use="required"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="SizeType">
<xsd:simpleContent>
<xsd:extension base="xsd:integer">
<xsd:attribute name="system" type="xsd:string"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:schema>
```

Task6_3.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://example.org/ord" xmlns:ord="http://example.org/ord"
xmlns:prod="http://example.org/prod" elementFormDefault="qualified">
<xsd:import namespace="http://example.org/prod" schemaLocation="Task6_2.xsd"/>
<xsd:element name="order" type="ord:OrderType"/>
<xsd:complexType name="OrderType">
<xsd:sequence>
<xsd:element name="number" type="xsd:string"/>
<xsd:element name="items" type="ord:ItemsType"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ItemsType">
```



```
<xsd:sequence>
  <xsd:element ref="prod:product" maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

Task6_4.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ord="http://example.org/ord">
  <xsd:import namespace="http://example.org/ord" schemaLocation="Task6_3.xsd"/>
  <xsd:element name="envelope" type="EnvelopeType"/>
  <xsd:complexType name="EnvelopeType">
    <xsd:sequence>
      <xsd:element ref="ord:order" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```