1)

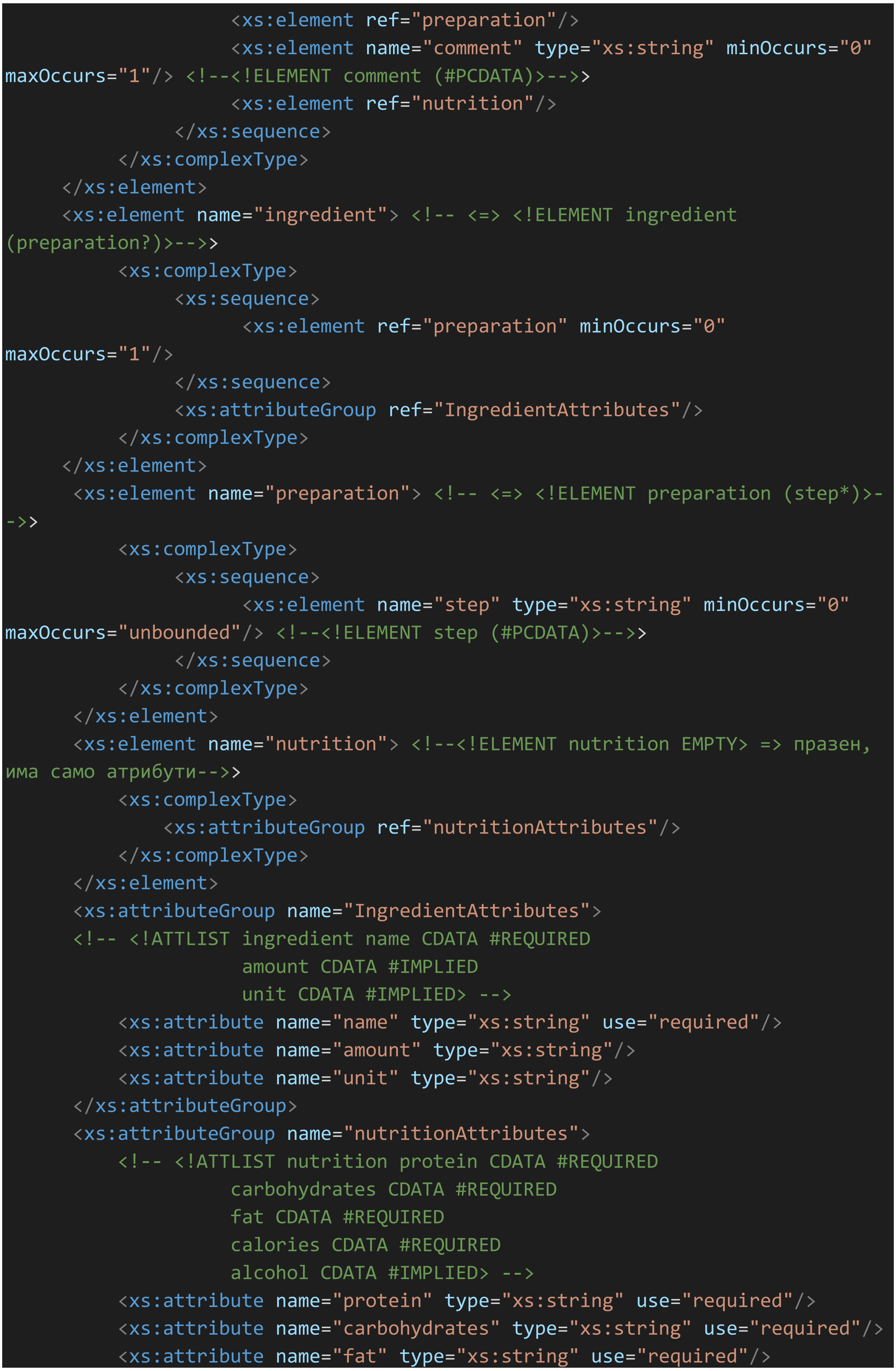
XML екземпляр:

```
<?xml version="1.0" encoding="UTF-8"?>
<collection xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="task1_xsd.xml">
    <description/>
    <recipe>
        <title>Title</title>
        <preparation>
            <step>woow</step>
        </preparation>
        <nutrition protein="20" carbohydrates="40" fat="10" calories="200"
alcohol="100"></nutrition>
    </recipe>
</collection>
```

XML схема:

task1_xsd.xml:

```
<?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"> <!-- <=> <!ELEMENT collection
(description,recipe*)> -->
        <xs:complexType>
            <xs:sequence>
                <xs:element name="description"> <!-- <=> <!ELEMENT
description ANY>-->>
                    <xs:complexType>
                        <xs:sequence minOccurs="1" maxOccurs="1">
                            <xs:any namespace="##any"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="recipe" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="recipe"> <!-- <=> <!ELEMENT recipe
(title,ingredient*,preparation,comment?,nutrition)> -->
        <xs:complexType>
            <xs:sequence>
                <xs:element name="title" type="xs:string"/> <!-- <=>
<!ELEMENT title (#PCDATA)>-->>
                <xs:element ref="ingredient" minOccurs="0"
maxOccurs="unbounded"/>
```

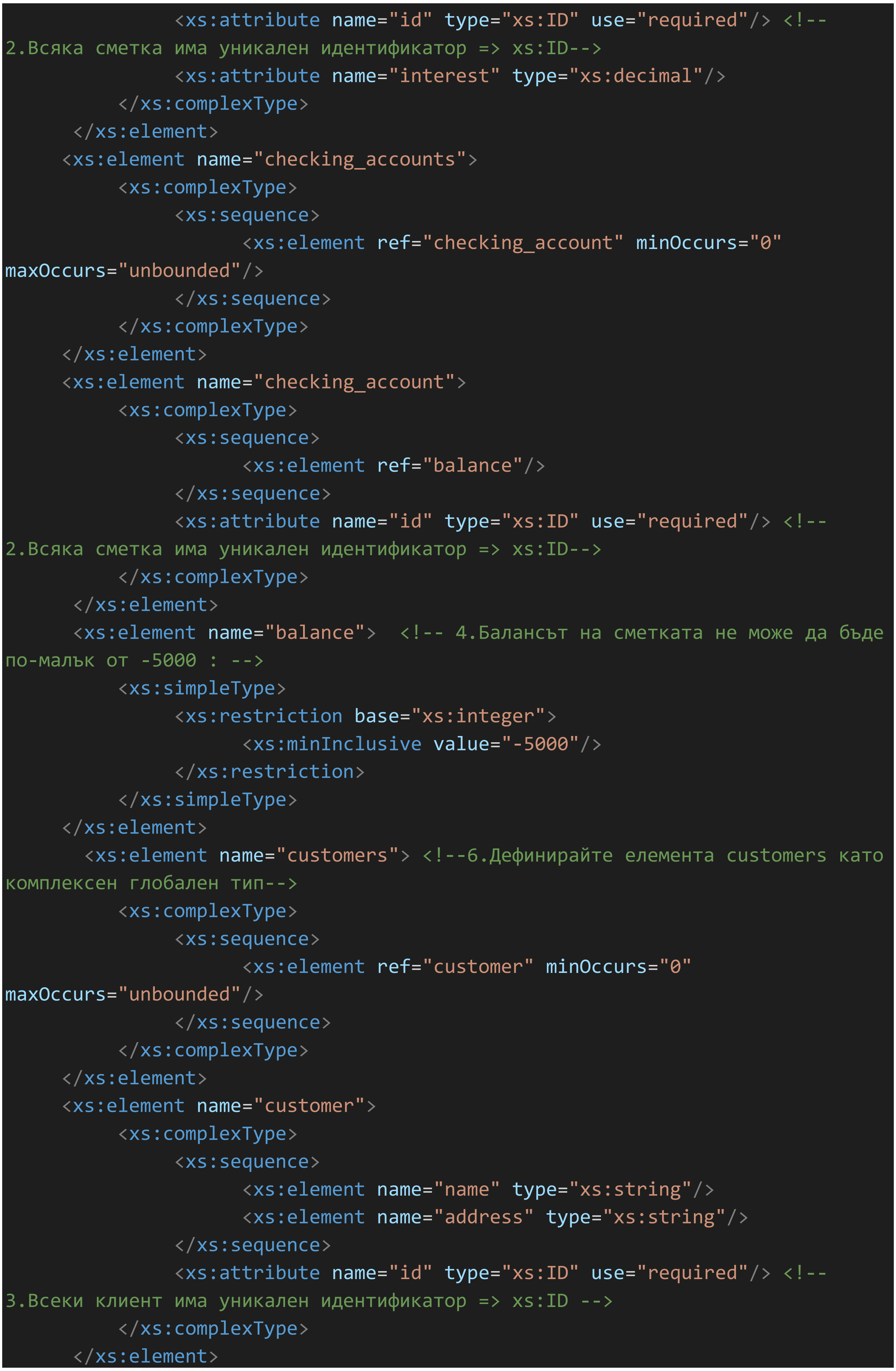


```
                    <xs:element ref="preparation"/>
                    <xs:element name="comment" type="xs:string" minOccurs="0"
maxOccurs="1"/> <!--<!ELEMENT comment (#PCDATA)>-->
                    <xs:element ref="nutrition"/>
               </xs:sequence>
          </xs:complexType>
     </xs:element>
     <xs:element name="ingredient"> <!-- <=> <!ELEMENT ingredient
(preparation?)>-->
          <xs:complexType>
               <xs:sequence>
                     <xs:element ref="preparation" minOccurs="0"
maxOccurs="1"/>
               </xs:sequence>
               <xs:attributeGroup ref="IngredientAttributes"/>
          </xs:complexType>
     </xs:element>
      <xs:element name="preparation"> <!-- <=> <!ELEMENT preparation (step*)>-
->
          <xs:complexType>
               <xs:sequence>
                     <xs:element name="step" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/> <!--<!ELEMENT step (#PCDATA)>-->
               </xs:sequence>
          </xs:complexType>
      </xs:element>
      <xs:element name="nutrition"> <!--<!ELEMENT nutrition EMPTY> => празен,
има само атрибути-->
          <xs:complexType>
              <xs:attributeGroup ref="nutritionAttributes"/>
          </xs:complexType>
      </xs:element>
      <xs:attributeGroup name="IngredientAttributes">
      <!-- <!ATTLIST ingredient name CDATA #REQUIRED
                     amount CDATA #IMPLIED
                     unit CDATA #IMPLIED> -->
          <xs:attribute name="name" type="xs:string" use="required"/>
          <xs:attribute name="amount" type="xs:string"/>
          <xs:attribute name="unit" type="xs:string"/>
      </xs:attributeGroup>
      <xs:attributeGroup name="nutritionAttributes">
          <!-- <!ATTLIST nutrition protein CDATA #REQUIRED
                    carbohydrates CDATA #REQUIRED
                    fat CDATA #REQUIRED
                    calories CDATA #REQUIRED
                    alcohol CDATA #IMPLIED> -->
          <xs:attribute name="protein" type="xs:string" use="required"/>
          <xs:attribute name="carbohydrates" type="xs:string" use="required"/>
          <xs:attribute name="fat" type="xs:string" use="required"/>
          <xs:attribute name="calories" type="xs:string" use="required"/>
          <xs:attribute name="alcohol" type="xs:string"/>
      </xs:attributeGroup>
</xs:schema>
```

2)

Файл task2_xsd.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
     <xs:element name="bank">
          <xs:complexType>
               <xs:all> <!-- 1.Редът на срещане на под-елементите на bank
(accounts, customers и customer_accounts) няма значение => използваме xs:all -
->
                    <xs:element name="accounts"> <!-- 6. Дефинирайте елемента
accounts като комплексен локален тип -->>
                         <xs:complexType>
                              <xs:all> <!-- 1.Редът на срещане на под-
елементите на accounts (saving_accounts и checking_accounts) няма значение =>
използваме xs:all -->
                                   <xs:element ref="saving_accounts"/>
                                   <xs:element ref="checking_accounts"/>
                              </xs:all>
                         </xs:complexType>
                    </xs:element>
                    <xs:element ref="customers"/>
                    <xs:element ref="customer_accounts"/>
               </xs:all>
          </xs:complexType>
     </xs:element>
     <xs:element name="saving_accounts">
          <xs:complexType>
               <xs:sequence>
                     <xs:element ref="saving_account" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
          </xs:complexType>
     </xs:element>
      <xs:element name="saving_account"> <!--6. Дефинирайте елемента
saving_account като комплексен глобален тип-->
          <xs:complexType>
               <xs:sequence>
                     <xs:element name="balance" type="xs:integer"/>
               </xs:sequence>
```



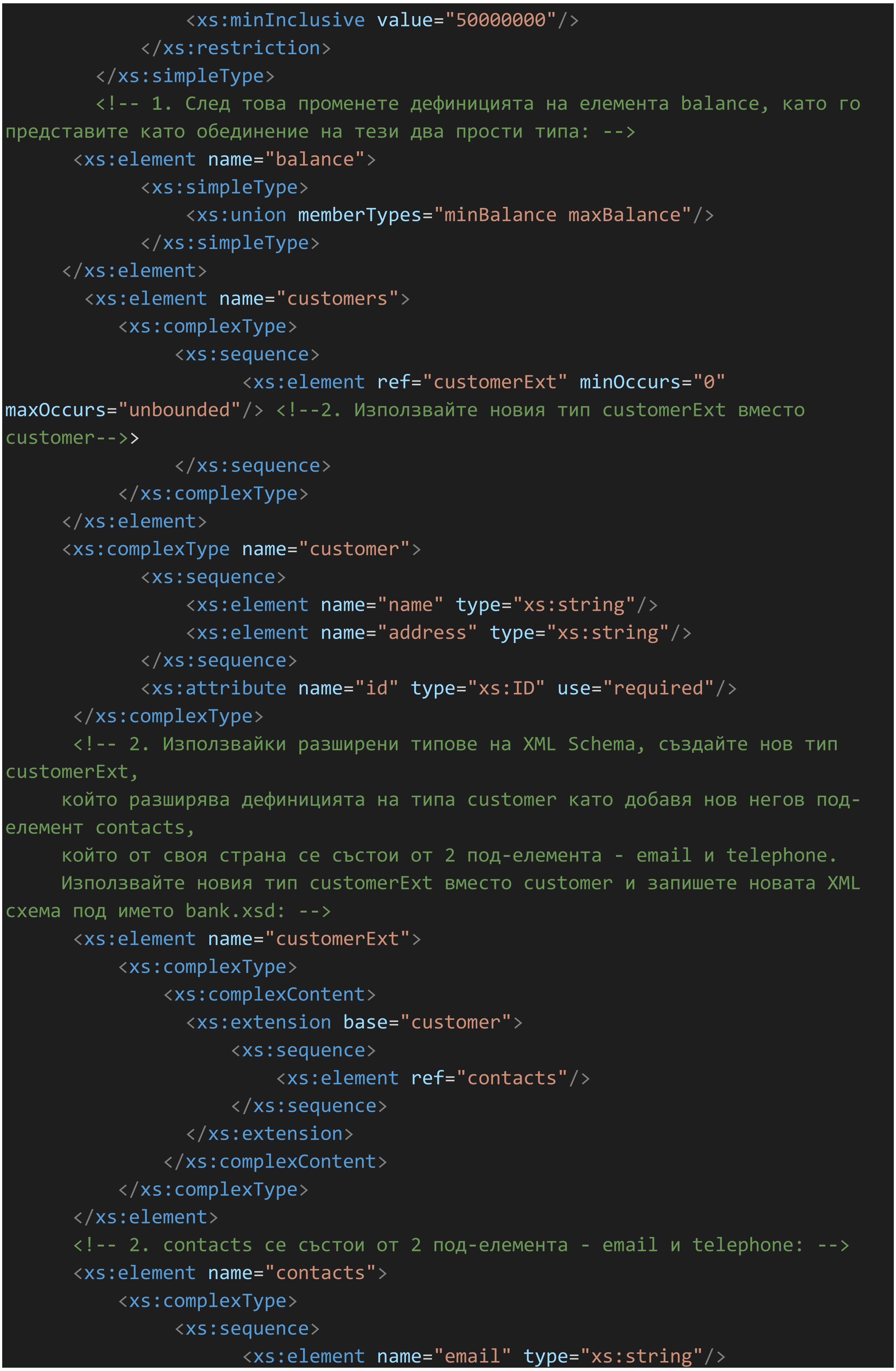
```
               <xs:attribute name="id" type="xs:ID" use="required"/> <!--
2.Всяка сметка има уникален идентификатор => xs:ID-->
               <xs:attribute name="interest" type="xs:decimal"/>
          </xs:complexType>
      </xs:element>
     <xs:element name="checking_accounts">
          <xs:complexType>
               <xs:sequence>
                     <xs:element ref="checking_account" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
          </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"/> <!--
2.Всяка сметка има уникален идентификатор => xs:ID-->
          </xs:complexType>
      </xs:element>
      <xs:element name="balance">  <!-- 4.Балансът на сметката не може да бъде
по-малък от -5000 : -->
          <xs:simpleType>
               <xs:restriction base="xs:integer">
                     <xs:minInclusive value="-5000"/>
               </xs:restriction>
          </xs:simpleType>
     </xs:element>
       <xs:element name="customers"> <!--6.Дефинирайте елемента customers като
комплексен глобален тип-->
          <xs:complexType>
               <xs:sequence>
                     <xs:element ref="customer" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
          </xs:complexType>
     </xs:element>
     <xs:element name="customer">
          <xs:complexType>
               <xs:sequence>
                     <xs:element name="name" type="xs:string"/>
                     <xs:element name="address" type="xs:string"/>
               </xs:sequence>
               <xs:attribute name="id" type="xs:ID" use="required"/> <!--
3.Всеки клиент има уникален идентификатор => xs:ID -->
          </xs:complexType>
      </xs:element>
```
]

```
      <xs:element name="customer_accounts">
          <xs:complexType>
               <xs:sequence>
                     <xs:element ref="customer_account" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
          </xs:complexType>
      </xs:element>
       <xs:element name="customer_account">
          <xs:complexType>
                <xs:attribute name="c_id" type="xs:IDREF" use="required"/> <!-
-5. Атрибутът c_id реферира към съответния клиент => използваме xs:IDREF-->
                <xs:attribute name="ac_id" type="xs:IDREF" use="required"/>
<!--5. Атрибутът ac_id реферира към съответната му сметка => използваме
xs:IDREF-->
          </xs:complexType>
      </xs:element>
</xs:schema>
```

3)

bank_xsd.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
     <xs:element name="bank">
          <annotation>
               <info>A bank contains accounts, customers and accounts of
customers</info> <!--4.Добавете няколко анотации в по-горе създадената XML
схема (bank.xsd) към избрани от вас комплексните типове, описващи тяхното
предназначение-->
          </annotation>
          <xs:complexType>
               <xs:all>
                    <xs:element ref="accounts"/>
                    <xs:element ref="customers"/>
                    <xs:element ref="customer_accounts"/>
               </xs:all>
          </xs:complexType>
     </xs:element>
     <xs:element name="accounts">
           <annotation>
               <info>An accounts contains saving accounts and checking
accounts</info> <!--4.Добавете няколко анотации в по-горе създадената XML
схема (bank.xsd) към избрани от вас комплексните типове, описващи тяхното
предназначение-->>
          </annotation>
          <xs:complexType>
                <xs:all>
                    <xs:element ref="saving_accounts"/>
                    <xs:element ref="checking_accounts"/>
               </xs:all>
          </xs:complexType>
     </xs:element>
     <xs:element name="saving_accounts">
          <xs:complexType>
               <xs:sequence>
                     <xs:element ref="saving_account" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
          </xs:complexType>
     </xs:element>
      <xs:element name="saving_account">
          <xs:complexType>
               <xs:sequence>
                     <xs:element name="balance" type="xs:integer"/>
               </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_accounts">
          <xs:complexType>
               <xs:sequence>
                     <xs:element ref="checking_account" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
          </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>
      <!-- 1. Дефинирайте два прости типа, задаващи горна и долна граница на
стойността на елемента balance: -->
        <xs:simpleType name="minBalance">
            <xs:restriction base="xs:integer">
                <xs:minInclusive value="-5000"/>
            </xs:restriction>
        </xs:simpleType>
      <xs:simpleType name="maxBalance">
            <xs:restriction base="xs:integer">
```



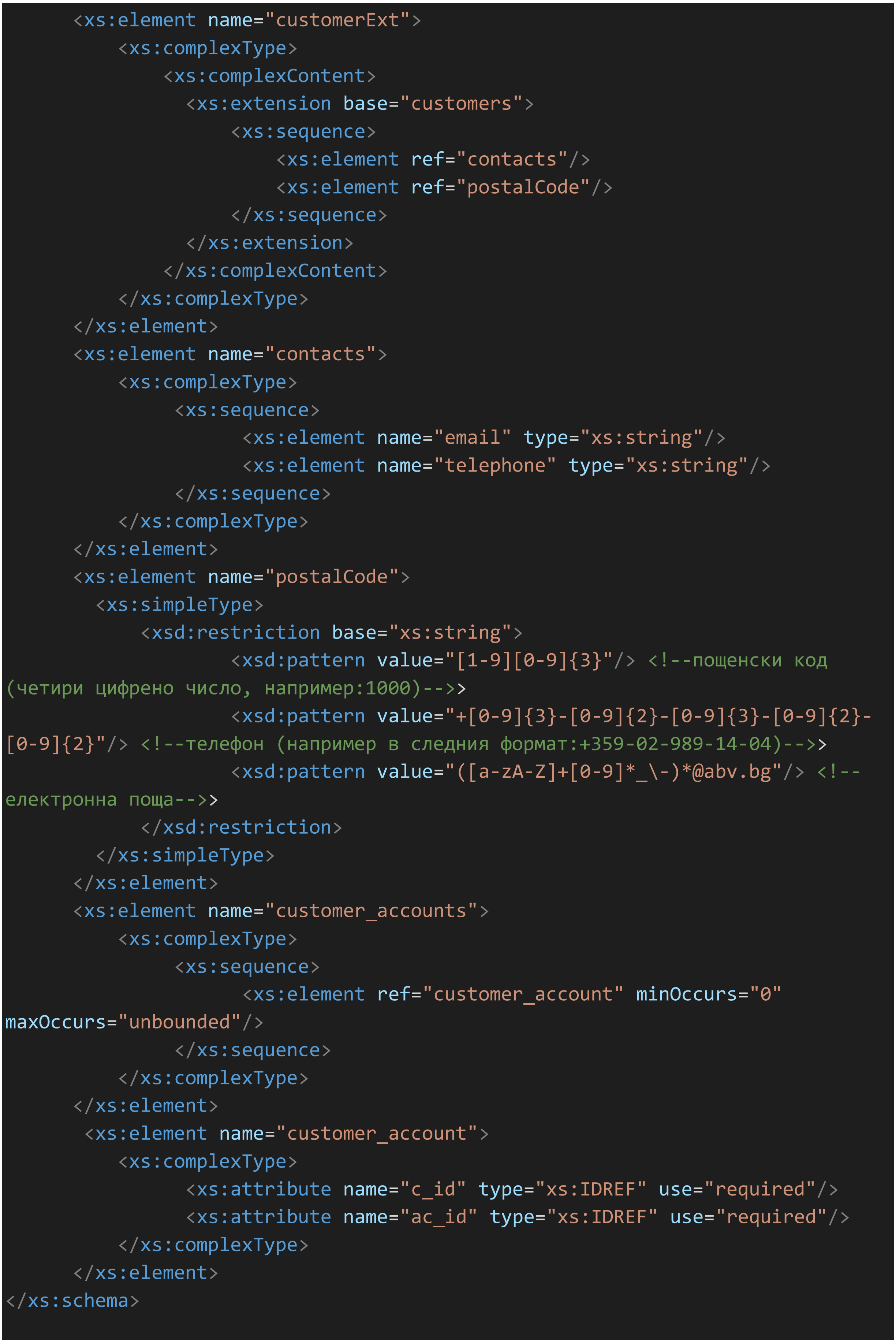
```
                <xs:minInclusive value="50000000"/>
            </xs:restriction>
        </xs:simpleType>
        <!-- 1. След това променете дефиницията на елемента balance, като го
представите като обединение на тези два прости типа: -->
      <xs:element name="balance">
            <xs:simpleType>
                <xs:union memberTypes="minBalance maxBalance"/>
            </xs:simpleType>
     </xs:element>
       <xs:element name="customers">
          <xs:complexType>
               <xs:sequence>
                     <xs:element ref="customerExt" minOccurs="0"
maxOccurs="unbounded"/> <!--2. Използвайте новия тип customerExt вместо
customer-->>
               </xs:sequence>
          </xs:complexType>
     </xs:element>
     <xs:complexType name="customer">
            <xs:sequence>
                <xs:element name="name" type="xs:string"/>
                <xs:element name="address" type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:ID" use="required"/>
      </xs:complexType>
      <!-- 2. 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: -->
      <xs:element name="customerExt">
          <xs:complexType>
              <xs:complexContent>
                <xs:extension base="customer">
                    <xs:sequence>
                        <xs:element ref="contacts"/>
                    </xs:sequence>
                </xs:extension>
              </xs:complexContent>
          </xs:complexType>
      </xs:element>
      <!-- 2. contacts се състои от 2 под-елемента - email и telephone: -->
      <xs:element name="contacts">
          <xs:complexType>
               <xs:sequence>
                     <xs:element name="email" type="xs:string"/>
                     <xs:element name="telephone" type="xs:string"/>
               </xs:sequence>
          </xs:complexType>
      </xs:element>
      <xs:element name="customer_accounts">
          <xs:complexType>
               <xs:sequence>
                     <xs:element ref="customer_account" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
          </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:schema>
```

bankExt_xsd.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<!-- 3. 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 -->
    <xs:redefine schemaLocation="bank_xsd">  <!-- предефинира типа balance и
customerExt от външната за нея XML схема - bank_xsd.xml-->
        <xs:complexType name="customerExt">
            <xs:complexContent>
                <xs:extension base="customerExt"> <!-- новият тип customerExt
съдържа допълнително нов под-елемент image-->
                    <xs:sequence>
                        <xs:element name="image" type="image" />
                    </xs:sequence>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <xs:simpleType name="minBalance">
            <xs:restriction base="minBalance"> <!-- новият тип balance не
трябва да бъде по-малък от -5200-->
                <xs:minInclusive value="-5200" />
            </xs:restriction>
        </xs:simpleType>
    </xs:redefine>
    <xs:complexType name="image"> <!-- елемент от тип base64Binary-->
        <xs:simpleContent>
            <xs:extension base="xs:base64Binary">
                <xs:attribute name="src" type="xs:string" use="required"/> <!-
- с атрибут src от тип string-->
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
```

4)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
     <xs:element name="bank">
          <xs:complexType>
               <xs:all>
                    <xs:element ref="accounts"/>
                    <xs:element ref="customers"/>
                    <xs:element ref="customer_accounts"/>
               </xs:all>
          </xs:complexType>
     </xs:element>
     <xs:element name="accounts">
          <xs:complexType>
                <xs:all>
                    <xs:element ref="saving_accounts"/>
                    <xs:element ref="checking_accounts"/>
               </xs:all>
          </xs:complexType>
     </xs:element>
     <xs:element name="saving_accounts">
          <xs:complexType>
               <xs:sequence>
                     <xs:element ref="saving_account" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
          </xs:complexType>
     </xs:element>
      <xs:element name="saving_account">
          <xs:complexType>
               <xs:sequence>
                     <xs:element name="balance" type="xs:integer"/>
               </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_accounts">
          <xs:complexType>
               <xs:sequence>
                     <xs:element ref="checking_account" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
          </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:simpleType name="minBalance">
            <xs:restriction base="xs:integer">
                <xs:minInclusive value="-5000"/>
            </xs:restriction>
        </xs:simpleType>
      <xs:simpleType name="maxBalance">
            <xs:restriction base="xs:integer">
                <xs:minInclusive value="50000000"/>
            </xs:restriction>
        </xs:simpleType>
      <xs:element name="balance">
            <xs:simpleType>
                <xs:union memberTypes="minBalance maxBalance"/>
            </xs:simpleType>
     </xs:element>
       <xs:element name="customers">
          <xs:complexType>
               <xs:sequence>
                     <xs:element ref="customerExt" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
          </xs:complexType>
     </xs:element>
     <xs:complexType name="customer">
            <xs:sequence>
                <xs:element name="name" type="xs:string"/>
                <xs:element name="address" type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:ID" use="required"/>
      </xs:complexType>
```



```
      <xs:element name="customerExt">
          <xs:complexType>
              <xs:complexContent>
                <xs:extension base="customers">
                    <xs:sequence>
                        <xs:element ref="contacts"/>
                        <xs:element ref="postalCode"/>
                    </xs:sequence>
                </xs:extension>
              </xs:complexContent>
          </xs:complexType>
      </xs:element>
      <xs:element name="contacts">
          <xs:complexType>
               <xs:sequence>
                     <xs:element name="email" type="xs:string"/>
                     <xs:element name="telephone" type="xs:string"/>
               </xs:sequence>
          </xs:complexType>
      </xs:element>
      <xs:element name="postalCode">
        <xs:simpleType>
            <xsd:restriction base="xs:string">
                    <xsd:pattern value="[1-9][0-9]{3}"/> <!--пощенски код
(четири цифрено число, например:1000)-->>
                    <xsd:pattern value="+[0-9]{3}-[0-9]{2}-[0-9]{3}-[0-9]{2}-
[0-9]{2}"/> <!--телефон (например в следния формат:+359-02-989-14-04)-->>
                    <xsd:pattern value="([a-zA-Z]+[0-9]*_\-)*@abv.bg"/> <!--
електронна поща-->>
            </xsd:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="customer_accounts">
          <xs:complexType>
               <xs:sequence>
                     <xs:element ref="customer_account" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
          </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:schema>
```

5)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:redefine schemaLocation="bank_xsd">
        <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:simpleType name="minBalance">
            <xs:restriction base="minBalance">
                <xs:minInclusive value="-5200" />
            </xs:restriction>
        </xs:simpleType>
    </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"/> <!-- Редактирайте XML схемата от задача 3 (bankExt_xsd.xml), като добавите към елемента image атрибут type от тип нотация.-->>
                    <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>
```

6)

task6.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<seller xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="task6_3.xml">
```

```
    <client xmlns="http://example.org/client"
xmlns:client="http://example.org/client">
        <name/>
        <years/>
        <favourite_shop>
            <name/>
            <description/>
        </favourite_shop>
    </client>
</seller>
```

task6_3.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:client="http://example.org/client">
 <xsd:import namespace="http://example.org/client"
schemaLocation="task6_2.xml"/>
 <xsd:element name="seller" type="sellerType"/>
 <xsd:complexType name="sellerType">
  <xsd:sequence>
   <xsd:element ref="client:client" maxOccurs="unbounded"/>
  </xsd:sequence>
 </xsd:complexType>
</xsd:schema>
```

task6_2.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://example.org/client"
xmlns:client2="http://example.org/client"
xmlns:client="http://example.org/client" elementFormDefault="qualified">
 <xsd:import namespace="http://example.org/shop"
schemaLocation="task6_1.xml"/>
 <xsd:element name="client" type="client:clientType"/>
 <xsd:complexType name="clientType">
  <xsd:sequence>
   <xsd:element name="name" type="xsd:string"/>
   <xsd:element name="years" type="xsd:integer"/>
   <xsd:element name="favourite_shop" type="client:fav_shop"/>
  </xsd:sequence>
 </xsd:complexType>
 <xsd:complexType name="fav_shop">
  <xsd:sequence>
   <xsd:element ref="shop:shop" maxOccurs="unbounded"/>
  </xsd:sequence>
 </xsd:complexType>
```

```
</xsd:schema>
```

task6_1.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://example.org/shop" xmlns="http://example.org/shop"
xmlns:shop2=" http://example.org/shop2" elementFormDefault="qualified">
 <xsd:import namespace="http://example.org/shop2"
schemaLocation="task6_1.xml"/>
 <xsd:element name="shop" type="shopType"/>
 <xsd:complexType name="shopType">
  <xsd:sequence>
   <xsd:element name="name" type="xs:string"/>
   <xsd:element name="description" type="xs:string"/>
  </xsd:sequence>
 </xsd:complexType>
</xsd:schema>
```