

Агро магазин с обект на разработка Склад

Име: Иван Иванов Петров
Факултетен номер: ЗМИ0700035

Съдържание

- Описание на предметната област и заданието
 - Описание на предметната област и на заданието
 - Дефиниция на схемата
 - Примерно съдържание
- Версия "Описание на DOM съответстващ на схемата"
 - Екземпляри на документа
 - Конструирание и валидиране на всеки екземпляр чрез DOM
- Версия "Набор от XSLT трансформации"
 - Трансформации XML към XML
 - Трансформации XML към Текст
- Заключения
 - Подобрения

1. Версия "XML схема"

Описание на предметната област и на заданието

Целта на проекта е създаване на система за управление на складовото приложение на агро магазин, специализиран в продажбата на селскостопански стоки и оборудване. Основни компоненти на складовата система включват управление на наличните продукти, доставки и контакти с доставчици и собственика. Системата трябва да позволи съхраняване на детайлна информация за всеки склад, като адреси, продукти и детайли за доставка и да осигурява ефективно проследяване на движението на стоки.

Всеки склад съдържа различни продукти, като семена, инструменти, оборудване с допълнителна информация като наличности, срок на годност и характеристики. Доставките включват връзка с доставчици и информация за състоянието на доставката (напр. изпълнена, отказана), както и цена.

Дефиниция на схемата

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="warehouses">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="owner" type="Vendor"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```

        <xsd:element name="warehouse" minOccurs="1" maxOccurs="unbounded"
type="Warehouse" />
    </xsd:sequence>
</xsd:complexType>
</xsd:element>

<xsd:complexType name="Warehouse">
    <xsd:sequence>
        <xsd:element name="products">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="product" type="Product" minOccurs="1"
maxOccurs="unbounded" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="contact" type="Contact" />
        <xsd:element name="deliveries">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="delivery" type="Delivery" minOccurs="0"
maxOccurs="unbounded" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Delivery">
    <xsd:sequence>
        <xsd:element name="products">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="product" type="Product" minOccurs="1"
maxOccurs="unbounded" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="vendor" type="Vendor" />
    </xsd:sequence>
    <xsd:attribute name="status" default="pending">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="canceled" />
                <xsd:enumeration value="returned" />
                <xsd:enumeration value="pending" />
                <xsd:enumeration value="finished" />
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name="cost" type="xsd:double" />
</xsd:complexType>

```

```

<xsd:complexType name="Vendor">
  <xsd:sequence>
    <xsd:element name="name" type="NonEmptyString" />
    <xsd:element name="contact" type="Contact" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Contact">
  <xsd:sequence>
    <xsd:element name="location" minOccurs="1" maxOccurs="1"
type="Location" />
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="email" type="Email" />
        <xsd:element name="phone" type="Phone" minOccurs="0" />
      </xsd:sequence>
      <xsd:sequence>
        <xsd:element name="email" type="Email" minOccurs="0" />
        <xsd:element name="phone" type="Phone" />
      </xsd:sequence>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Location">
  <xsd:sequence>
    <xsd:element name="country" type="xsd:string" />
    <xsd:element name="city" type="xsd:string" />
    <xsd:element name="address" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Product">
  <xsd:sequence>
    <xsd:element name="info" type="ContentInfo" />
    <xsd:element name="notes" type="NonEmptyString" minOccurs="0"
maxOccurs="1">
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="name" type="NonEmptyString" use="required">
      </xsd:attribute>
    <xsd:attribute name="brand" type="NonEmptyString" use="required" />
  </xsd:complexType>

<xsd:complexType name="ContentInfo">
  <xsd:sequence>
    <xsd:element name="fragile" type="xsd:boolean" minOccurs="0"
maxOccurs="1"
      default="false" />
    <xsd:element name="price">
      <xsd:complexType>
        <xsd:simpleContent>
          <xsd:extension base="xsd:double">
            <xsd:attribute name="currency" default="BGN">

```

```

        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="BGN"/>
                <xsd:enumeration value="EUR"/>
                <xsd:enumeration value="USD"/>
                <xsd:enumeration value="RON"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name="quantity" type="xsd:int" />
<xsd:element name="weight">
    <xsd:complexType>
        <xsd:simpleContent>
            <xsd:extension base="xsd:double">
                <xsd:attribute name="unit" use="optional" default="g">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="g" />
                            <xsd:enumeration value="kg" />
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
</xsd:element>
<xsd:element name="expiry_date" type="xsd:date" />
<xsd:element name="type">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="tool" />
            <xsd:enumeration value="seeds" />
            <xsd:enumeration value="clothing" />
            <xsd:enumeration value="equipment" />
            <xsd:enumeration value="structure" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="Email">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="254" />
        <xsd:pattern
            value="[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}" />
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="Phone">

```

```

        <xsd:restriction base="xsd:string">
            <xsd:pattern
                value="(\+\d{1,3}\s?)?\(?\d{1,3}\)[\s\/-]? \d{2,4}[\s\/-]? \d{2,6}"
            />
        </xsd:restriction>
    </xsd:simpleType>

    <xsd:simpleType name="NonEmptyString">
        <xsd:restriction base="xsd:string">
            <xsd:minLength value="1" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>

```

Примерно съдържание

```

<?xml version="1.0" encoding="UTF-8"?>
<warehouses xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="warehouses.xsd">
    <owner>
        <name>Bulgarian Agro Supply</name>
        <contact>
            <location>
                <country>Bulgaria</country>
                <city>Sofia</city>
                <address>15 Tsar Simeon Street</address>
            </location>
            <email>contact@bulgarianagrosupply.bg</email>
            <phone>+359 (888)/123/456</phone>
        </contact>
    </owner>

    <warehouse>
        <products>
            <product name="Garden Shovel" brand="GardenMaster">
                <info>
                    <fragile>false</fragile>
                    <price currency="BGN">12.50</price>
                    <quantity>100</quantity>
                    <weight unit="kg">1.2</weight>
                    <expiry_date>2025-12-31</expiry_date>
                    <type>tool</type>
                </info>
                <notes>Heavy-duty garden shovel for soil work</notes>
            </product>
            <product name="Tomato Seeds" brand="SeedCo">
                <info>
                    <fragile>false</fragile>
                    <price currency="BGN">3.20</price>
                    <quantity>500</quantity>
                    <weight unit="g">0.05</weight>
                </info>
            </product>
        </products>
    </warehouse>
</warehouses>

```

```
        <expiry_date>2026-04-15</expiry_date>
        <type>seeds</type>
    </info>
    <notes>High yield variety</notes>
</product>
<product name="Safety Gloves" brand="SafeHands">
    <info>
        <fragile>false</fragile>
        <price currency="BGN">8.00</price>
        <quantity>200</quantity>
        <weight unit="g">0.15</weight>
        <expiry_date>2025-06-01</expiry_date>
        <type>clothing</type>
    </info>
</product>
</products>
<contact>
    <location>
        <country>Greece</country>
        <city>Thessaloniki</city>
        <address>45 Greenhouse Rd</address>
    </location>
    <email>thess@bulgarianagrosupply.gr</email>
</contact>
<deliveries>
    <delivery status="pending" cost="200.50">
        <products>
            <product name="Garden Shovel" brand="GardenMaster">
                <info>
                    <fragile>false</fragile>
                    <price currency="EUR">12.50</price>
                    <quantity>20</quantity>
                    <weight unit="kg">1.2</weight>
                    <expiry_date>2025-12-31</expiry_date>
                    <type>tool</type>
                </info>
            </product>
        </products>
    </delivery>
</deliveries>
</warehouse>

<warehouse>
```

```
<products>
  <product name="Irrigation Pipes" brand="WaterFlow">
    <info>
      <fragile>false</fragile>
      <price currency="BGN">35.00</price>
      <quantity>60</quantity>
      <weight unit="kg">3.5</weight>
      <expiry_date>2026-07-15</expiry_date>
      <type>equipment</type>
    </info>
  </product>
  <product name="Greenhouse Cover" brand="EcoCover">
    <info>
      <fragile>true</fragile>
      <price currency="EUR">75.00</price>
      <quantity>40</quantity>
      <weight unit="kg">5.0</weight>
      <expiry_date>2027-03-01</expiry_date>
      <type>structure</type>
    </info>
  </product>
  <product name="Garden Fork" brand="GreenTools">
    <info>
      <fragile>false</fragile>
      <price currency="BGN">15.00</price>
      <quantity>150</quantity>
      <weight unit="kg">1.5</weight>
      <expiry_date>2025-09-15</expiry_date>
      <type>tool</type>
    </info>
  </product>
</products>
<contact>
  <location>
    <country>Bulgaria</country>
    <city>Plovdiv</city>
    <address>33 Warehouse Street</address>
  </location>
  <email>plovdiv@bulgarianagrosupply.bg</email>
  <phone>+359 888 654321</phone>
</contact>
<deliveries>
  <delivery status="finished" cost="150.75">
    <products>
      <product name="Irrigation Pipes" brand="WaterFlow">
        <info>
          <fragile>false</fragile>
          <price currency="BGN">35.00</price>
          <quantity>10</quantity>
          <weight unit="kg">3.5</weight>
          <expiry_date>2026-07-15</expiry_date>
          <type>equipment</type>
        </info>
      </product>
```

```
</products>
<vendor>
  <name>Water Supplies Ltd</name>
  <contact>
    <location>
      <country>Romania</country>
      <city>Bucharest</city>
      <address>101 Water Rd</address>
    </location>
    <email>orders@watersupplies.ro</email>
  </contact>
</vendor>
</delivery>
</deliveries>
</warehouse>

<warehouse>
  <products>
    <product name="Plant Fertilizer" brand="EcoGro">
      <info>
        <fragile>false</fragile>
        <price currency="BGN">25.00</price>
        <quantity>80</quantity>
        <weight unit="kg">1.0</weight>
        <expiry_date>2026-10-01</expiry_date>
        <type>tool</type>
      </info>
    </product>
    <product name="Pruning Shears" brand="SharpCut">
      <info>
        <fragile>true</fragile>
        <price currency="BGN">18.00</price>
        <quantity>50</quantity>
        <weight unit="g">150</weight>
        <expiry_date>2025-08-20</expiry_date>
        <type>tool</type>
      </info>
    </product>
    <product name="Pesticide Spray" brand="SafeGrow">
      <info>
        <fragile>false</fragile>
        <price currency="BGN">30.00</price>
        <quantity>30</quantity>
        <weight unit="kg">0.8</weight>
        <expiry_date>2027-12-01</expiry_date>
        <type>tool</type>
      </info>
    </product>
  </products>
  <contact>
    <location>
      <country>North Macedonia</country>
      <city>Skopje</city>
      <address>22 Agriculture Blvd</address>
```



```

        </location>
        <email>skopje@bulgarianagrosupply.mk</email>
        <phone>+389 70 123456</phone>
    </contact>
    <deliveries>
        <delivery status="pending" cost="110.00">
            <products>
                <product name="Pesticide Spray" brand="SafeGrow">
                    <info>
                        <fragile>false</fragile>
                        <price currency="BGN">30.00</price>
                        <quantity>5</quantity>
                        <weight unit="kg">0.8</weight>
                        <expiry_date>2027-12-01</expiry_date>
                        <type>tool</type>
                    </info>
                </product>
            </products>
            <vendor>
                <name>Safe Chemicals Ltd</name>
                <contact>
                    <location>
                        <country>Serbia</country>
                        <city>Belgrade</city>
                        <address>50 Chemical Street</address>
                    </location>
                    <email>orders@safechemicals.rs</email>
                </contact>
            </vendor>
        </delivery>
    </deliveries>
</warehouse>
</warehouses>

```

2. Версия "Описание на DOM съответстващ на схемата"

Екземпляри на документа

```

<?xml version="1.0" encoding="UTF-8"?>
<warehouses xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="warehouses.xsd">
    <owner>
        <name>Агро магазин Владимир Огнянов</name>
        <contact>
            <location>
                <country>Словения</country>
                <city>Марибор</city>
                <address>ул. Главен площад 4</address>
            </location>
            <email>agro_magazin_vladimir_ognyanov1009@outlook.com</email>
        </contact>
    </owner>
</warehouses>

```

```
</owner>
<warehouse>
  <products>
    <product name="трактор" brand="Машиникс">
      <info>
        <fragile>false</fragile>
        <price currency="USD">150000</price>
        <quantity>24</quantity>
        <weight unit="kg">1500</weight>
        <expiry_date>2032-05-25</expiry_date>
        <type>equipment</type>
      </info>
      <notes>идеален за средни площи</notes>
    </product>
    <product name="лопата" brand="АгроИнструмент">
      <info>
        <fragile>false</fragile>
        <price currency="BGN">25.5</price>
        <quantity>12</quantity>
        <weight unit="kg">2.3</weight>
        <expiry_date>2028-01-15</expiry_date>
        <type>tool</type>
      </info>
      <notes>подходяща за градинска работа</notes>
    </product>
  </products>
  <contact>
    <location>
      <country>Сърбия</country>
      <city>Белград</city>
      <address>бул. Краля Александра 25</address>
    </location>
    <email>warehouse_mariya_todorova1239@yahoo.com</email>
    <phone>+808 422-857-552257</phone>
  </contact>
  <deliveries>
    <delivery status="canceled" cost="7813594.5">
      <products>
        <product name="форма за разсад" brand="СеменАгро">
          <info>
            <fragile>true</fragile>
            <price currency="EUR">50</price>
            <quantity>14</quantity>
            <weight unit="kg">2.5</weight>
            <expiry_date>2026-04-18</expiry_date>
            <type>equipment</type>
          </info>
          <notes>за точни и равномерни посеви</notes>
        </product>
        <product name="трактор" brand="Машиникс">
          <info>
            <fragile>false</fragile>
            <price currency="USD">150000</price>
            <quantity>28</quantity>
```

```
        <weight unit="kg">1500</weight>
        <expiry_date>2032-05-25</expiry_date>
        <type>equipment</type>
    </info>
    <notes>идеален за средни площи</notes>
</product>
<product name="защитни очила" brand="ФермерГард">
    <info>
        <fragile>true</fragile>
        <price currency="BGN">10</price>
        <quantity>9</quantity>
        <weight unit="g">250</weight>
        <expiry_date>2032-05-20</expiry_date>
        <type>clothing</type>
    </info>
    <notes>защита от при пръскане</notes>
</product>
<product name="семена пепино пъпеш" brand="Градинар">
    <info>
        <fragile>false</fragile>
        <price currency="RON">12.99</price>
        <quantity>4</quantity>
        <weight unit="g">0.9</weight>
        <expiry_date>2030-04-10</expiry_date>
        <type>seeds</type>
    </info>
    <notes>медено сладък сорт</notes>
</product>
</products>
<vendor>
    <name>Тодор Борисов</name>
    <contact>
        <location>
            <country>Турция</country>
            <city>Истанбул</city>
            <address>пл. Таксим 1</address>
        </location>
        <email>todor_borisov1727@outlook.com</email>
        <phone>+34 501-8411-571</phone>
    </contact>
</vendor>
</delivery>
<delivery status="canceled" cost="673.28">
    <products>
        <product name="семена от домати" brand="СемеАгро">
            <info>
                <fragile>false</fragile>
                <price currency="EUR">8.5</price>
                <quantity>8</quantity>
                <weight unit="g">3</weight>
                <expiry_date>2025-10-10</expiry_date>
                <type>seeds</type>
            </info>
            <notes>високодобивни сортове</notes>
```

```

        </product>
        <product name="градски ботуши" brand="ТехноОбувки">
            <info>
                <fragile>false</fragile>
                <price currency="BGN">45</price>
                <quantity>12</quantity>
                <weight unit="kg">1.5</weight>
                <expiry_date>2032-02-20</expiry_date>
                <type>clothing</type>
            </info>
            <notes>удобни и издръжливи при всякакви условия</notes>
        </product>
    </products>
    <vendor>
        <name>Тодор Цветков</name>
        <contact>
            <location>
                <country>Румъния</country>
                <city>Букурещ</city>
                <address>ул. Липскани 32</address>
            </location>
            <email>todor_tsvetkov1308@outlook.com</email>
            <phone>+240 066-102-55702</phone>
        </contact>
    </vendor>
</delivery>
</deliveries>
</warehouse>
</warehouses>

```

Конструиране и валидиране на всеки екземпляр чрез DOM

```

//@ts-check

import {
    generateEmail,
    generateFullName,
    generatePhone,
} from "../data-generator";
import { contactLocations, deliveryStatuses, products } from "../types";
import { precision, random } from "../utility";

const namespace = "http://www.w3.org/2001/XMLSchema-instance";
let deliveriesCount = 3;
let productsCount = 3;
let warehousesCount = 3;

/**
 *
 * @param {number} deliveryCnt
 * @param {number} productsCnt

```

```

* @param {number} warehousesCnt
* @returns
*/
export function generateDoc(deliveryCnt, productsCnt, warehousesCnt) {
  deliveriesCount = deliveryCnt;
  productsCount = productsCnt;
  warehousesCount = warehousesCnt;

  const documentImplementation = document.implementation;
  const xmlDoc = documentImplementation.createDocument(null, null);

  const xml = xmlDoc.createProcessingInstruction(
    "xml",
    'version="1.0" encoding="UTF-8"'
  );
  xmlDoc.appendChild(xml);

  let warehousesEl = xmlDoc.createElement("warehouses");
  warehousesEl.setAttribute("xmlns:xsi", namespace);
  warehousesEl.setAttribute(
    "xsi:noNamespaceSchemaLocation",
    "warehouses.xsd"
  );

  let ownerEl = xmlDoc.createElement("owner");
  let fullName = `Арго магазин ${generateFullName()}`;
  let nameEl = xmlDoc.createElement("name");
  nameEl.textContent = fullName;

  ownerEl.appendChild(nameEl);
  ownerEl.appendChild(generateContact(xmlDoc, fullName));
  warehousesEl.appendChild(ownerEl);

  for (let ind = 0; ind < random(warehousesCount) + 1; ind++) {
    warehousesEl.appendChild(generateWarehouse(xmlDoc));
  }

  xmlDoc.appendChild(warehousesEl);
  return xmlDoc;
}
/**
 *
 * @param {XMLDocument} xmlDoc
 * @returns {Element} Random warehouse
 */
function generateWarehouse(xmlDoc) {
  let warehouseEl = xmlDoc.createElement("warehouse");
  let productsEl = xmlDoc.createElement("products");
  let setOfProducts = new Set();

  for (let ind = 0; ind < random(warehousesCount) + 1; ind++) {
    let randProduct;
    while (true) {
      randProduct = products[random(products.length)];
    }
  }
}

```

```

        if (!setOfProducts.has(randProduct)) {
            setOfProducts.add(randProduct);
            break;
        }
    }

    productsEl.appendChild(generateProduct(xmlDoc, randProduct).element);
}

let deliveriesEl = xmlDoc.createElement("deliveries");
for (let ind = 0; ind < random(deliveriesCount) + 1; ind++) {
    deliveriesEl.appendChild(generateDelivery(xmlDoc));
}

warehouseEl.appendChild(productsEl);
warehouseEl.appendChild(
    generateContact(xmlDoc, `Warehouse ${generateFullName()}`)
);
warehouseEl.appendChild(deliveriesEl);

return warehouseEl;
}

/**
 *
 * @param {XMLDocument} xmlDoc
 * @returns {Element} Random delivery
 */
function generateDelivery(xmlDoc) {
    let deliveryEl = xmlDoc.createElement("delivery");
    let setOfProducts = new Set();
    let totalCost = 0;
    let productsEl = xmlDoc.createElement("products");

    for (let ind = 0; ind < random(productsCount) + 1; ind++) {
        let randProduct;
        while (true) {
            randProduct = products[random(products.length)];
            if (!setOfProducts.has(randProduct)) {
                setOfProducts.add(randProduct);
                break;
            }
        }
        let result = generateProduct(xmlDoc, randProduct);
        totalCost += result.cost;
        productsEl.appendChild(result.element);
    }
    console.log(totalCost);
    let vendorEl = xmlDoc.createElement("vendor");
    let fullName = generateFullName();
    let nameEl = xmlDoc.createElement("name");
    nameEl.textContent = fullName;

    vendorEl.appendChild(nameEl);

```

```

        vendorEl.appendChild(generateContact(xmlDoc, fullName));

        deliveryEl.setAttribute(
            "status",
            deliveryStatuses[random(deliveryStatuses.length)]
        );

        deliveryEl.setAttribute("cost", precision(totalCost, 2).toString());

        deliveryEl.appendChild(productsEl);
        deliveryEl.appendChild(vendorEl);
        return deliveryEl;
    }

/**
 *
 * @param {XMLDocument} xmlDoc
 * @param {import("./types").Product} product
 * @returns {{element:Element, cost:number}}
 */
function generateProduct(xmlDoc, product) {
    let productEl = xmlDoc.createElement("product");
    productEl.setAttribute("name", product.name);
    productEl.setAttribute("brand", product.brand);

    let notesEl = xmlDoc.createElement("notes");
    notesEl.textContent = product.notes;

    let contentInfoEl = xmlDoc.createElement("info");
    let fragileEl = xmlDoc.createElement("fragile");
    fragileEl.textContent = product.contentInfo.isFragile.toString();
    contentInfoEl.appendChild(fragileEl);

    let priceEl = xmlDoc.createElement("price");
    priceEl.textContent = product.contentInfo.price.toString();
    priceEl.setAttribute("currency", product.contentInfo.currency);
    contentInfoEl.appendChild(priceEl);

    let quantityEl = xmlDoc.createElement("quantity");
    const quantity = random(30) + 1;
    quantityEl.textContent = quantity.toString();
    contentInfoEl.appendChild(quantityEl);

    let weightEl = xmlDoc.createElement("weight");
    weightEl.textContent = product.contentInfo.weight.toString();
    weightEl.setAttribute("unit", product.contentInfo.unit);
    contentInfoEl.appendChild(weightEl);

    let expireDateEl = xmlDoc.createElement("expiry_date");
    expireDateEl.textContent = product.contentInfo.expiryDate;
    contentInfoEl.appendChild(expireDateEl);

    let typeEl = xmlDoc.createElement("type");
    typeEl.textContent = product.contentInfo.type;

```

```

        contentInfoEl.appendChild(typeEl);

        productEl.appendChild(contentInfoEl);
        productEl.appendChild(notesEl);

        let currencyCoeff = 1;
        if (product.contentInfo.currency === "EUR") {
            currencyCoeff = 1.96;
        } else if (product.contentInfo.currency === "USD") {
            currencyCoeff = 1.86;
        } else if (product.contentInfo.currency === "RON") {
            currencyCoeff = 2.55;
        }
        return {
            element: productEl,
            cost: quantity * product.contentInfo.price * currencyCoeff,
        };
    }

    /**
     *
     * @param {XMLDocument} xmlDoc
     * @param {string} fullname
     * @returns {Element}
     */
    function generateContact(xmlDoc, fullname = generateFullName()) {
        let contactEl = xmlDoc.createElement("contact");
        let locationEl = xmlDoc.createElement("location");

        let randomLocation = contactLocations[random(contactLocations.length)];
        let countryEl = xmlDoc.createElement("country");
        countryEl.textContent = randomLocation.country;
        let cityEl = xmlDoc.createElement("city");
        cityEl.textContent = randomLocation.city;
        let addressEl = xmlDoc.createElement("address");
        addressEl.textContent = randomLocation.address;

        locationEl.appendChild(countryEl);
        locationEl.appendChild(cityEl);
        locationEl.appendChild(addressEl);

        contactEl.appendChild(locationEl);

        let choice = random(3);
        if (choice < 1) {
            let emailEl = xmlDoc.createElement("email");
            emailEl.textContent = generateEmail(fullname);
            contactEl.appendChild(emailEl);
        } else if (choice === 1) {
            let phoneEl = xmlDoc.createElement("phone");
            phoneEl.textContent = generatePhone();
            contactEl.appendChild(phoneEl);
        } else {
            let emailEl = xmlDoc.createElement("email");

```



```

        emailEl.textContent = generateEmail(fullname);
        let phoneEl = xmlDoc.createElement("phone");
        phoneEl.textContent = generatePhone();
        contactEl.appendChild(emailEl);
        contactEl.appendChild(phoneEl);
    }

    return contactEl;
}

```

3. Версия 'Набор от XSLT трансформации'

Трансформации XML към XML

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" encoding="UTF-8" indent="yes"/>

  <xsl:template match="/">
    <warehouses>
      <xsl:apply-templates select="warehouses"/>
    </warehouses>
  </xsl:template>

  <xsl:template match="warehouses">
    <xsl:element name="owner">
      <xsl:apply-templates select="owner"/>
    </xsl:element>
    <xsl:for-each select="warehouse">
      <xsl:element name="warehouse">
        <contact>
          <xsl:apply-templates select="contact"/>
        </contact>
        <xsl:if test="deliveries">
          <deliveries>
            <xsl:apply-templates select="deliveries/delivery"/>
          </deliveries>
        </xsl:if>
      </xsl:element>
    </xsl:for-each>
  </xsl:template>

  <xsl:template match="owner|vendor">
    <xsl:element name="name">
      <xsl:value-of select="name"/>
    </xsl:element>
    <xsl:element name="contact">
      <xsl:apply-templates select="contact"/>
    </xsl:element>
  </xsl:template>

```

```

<xsl:template match="delivery">
  <delivery>
    <xsl:attribute name="status">
      <xsl:value-of select="@status"/>
    </xsl:attribute>
    <xsl:attribute name="cost">
      <xsl:value-of select="@cost"/>
    </xsl:attribute>
    <vendor>
      <xsl:apply-templates select="vendor"/>
    </vendor>
  </delivery>
</xsl:template>

<xsl:template match="contact">
  <location>
    <country>
      <xsl:value-of select="location/country"/>
    </country>
    <city>
      <xsl:value-of select="location/city"/>
    </city>
    <address>
      <xsl:value-of select="location/address"/>
    </address>
  </location>
  <xsl:if test="email">
    <email>
      <xsl:value-of select="email"/>
    </email>
  </xsl:if>

  <xsl:if test="phone">
    <phone>
      <xsl:value-of select="phone"/>
    </phone>
  </xsl:if>
</xsl:template>
</xsl:stylesheet>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" encoding="UTF-8" indent="yes"/>

  <xsl:template match="/">
    <warehouses xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="../public/warehouses.xsd">
      <xsl:apply-templates select="warehouses"/>
    </warehouses>
  </xsl:template>

```

```

<xsl:template match="warehouses">
  <xsl:element name="owner">
    <xsl:apply-templates select="owner"/>
  </xsl:element>
  <xsl:for-each select="warehouse[1]">
    <xsl:element name="warehouse">
      <products>
        <xsl:apply-templates select="products/product"/>
      </products>
      <contact>
        <xsl:apply-templates select="contact"/>
      </contact>
      <deliveries/>
    </xsl:element>
  </xsl:for-each>
</xsl:template>

<xsl:template match="owner|vendor">
  <xsl:element name="name">
    <xsl:value-of select="name"/>
  </xsl:element>
  <xsl:element name="contact">
    <xsl:apply-templates select="contact"/>
  </xsl:element>
</xsl:template>

<xsl:template match="product">
  <product>
    <xsl:attribute name="name">
      <xsl:value-of select="@name"/>
    </xsl:attribute>
    <xsl:attribute name="brand">
      <xsl:value-of select="@brand"/>
    </xsl:attribute>
    <info>
      <xsl:apply-templates select="info"/>
    </info>
  </product>
</xsl:template>

<xsl:template match="info">
  <price>
    <xsl:attribute name="currency">
      <xsl:value-of select="price/@currency"/>
    </xsl:attribute>
    <xsl:value-of select="price"/>
  </price>
  <quantity>
    <xsl:value-of select="quantity"/>
  </quantity>
  <weight>
    <xsl:attribute name="unit">
      <xsl:value-of select="weight/@unit"/>
    </xsl:attribute>
  </weight>

```

```

        </xsl:attribute>
        <xsl:value-of select="weight"/>
    </weight>
    <expiry_date>
        <xsl:value-of select="expiry_date"/>
    </expiry_date>
    <type>
        <xsl:value-of select="type"/>
    </type>
</xsl:template>

<xsl:template match="contact">
    <location>
        <country>
            <xsl:value-of select="location/country"/>
        </country>
        <city>
            <xsl:value-of select="location/city"/>
        </city>
        <address>
            <xsl:value-of select="location/address"/>
        </address>
    </location>
    <xsl:choose>
        <xsl:when test="email">
            <email>
                <xsl:value-of select="email"/>
            </email>
        </xsl:when>
        <xsl:otherwise>
            <phone>
                <xsl:value-of select="phone"/>
            </phone>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>
</xsl:stylesheet>

```

Трансформации XML към Текст

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
    <xsl:output method="text" encoding="UTF-8"/>

    <xsl:template match="/warehouses">
        Собственик:
        <xsl:value-of select="owner/name"/>
        <xsl:text>##10;Контакт:</xsl:text>
        <xsl:apply-templates select="owner/contact"/>
        <xsl:text>##10;=====##10;</xsl:text>
    </xsl:template>

```

```

    <xsl:for-each select="warehouse">
      <xsl:text>Склад&#10;</xsl:text>
      <xsl:apply-templates select="contact"/>
      <xsl:text>&#10;Продукти в склада:&#10;</xsl:text>
      <xsl:apply-templates select="products/product"/>
      <xsl:text>Доставки:&#10;</xsl:text>
      <xsl:apply-templates select="deliveries/delivery"/>
      <xsl:text>=====&#10;</xsl:text>
    </xsl:for-each>
  </xsl:template>

  <xsl:template match="contact">
    <xsl:text>Местоположение: </xsl:text>
    <xsl:value-of select="location/country"/>
    <xsl:text>, </xsl:text>
    <xsl:value-of select="location/city"/>
    <xsl:text>, </xsl:text>
    <xsl:value-of select="location/address"/>
    <xsl:text>&#10;Имейл: </xsl:text>
    <xsl:value-of select="email"/>
    <xsl:text>&#10;Телефон: </xsl:text>
    <xsl:value-of select="phone"/>
  </xsl:template>

  <xsl:template match="product">
    <xsl:text>- Продукт: </xsl:text>
    <xsl:value-of select="@name"/>
    <xsl:text> (Марка: </xsl:text>
    <xsl:value-of select="@brand"/>
    <xsl:text>)&#10;</xsl:text>
    <xsl:text> Цена: </xsl:text>
    <xsl:value-of select="info/price"/>
    <xsl:text> </xsl:text>
    <xsl:value-of select="info/price/@currency"/>
    <xsl:text>, Количество: </xsl:text>
    <xsl:value-of select="info/quantity"/>
    <xsl:text>, Тип: </xsl:text>
    <xsl:value-of select="info/type"/>
    <xsl:text>&#10;</xsl:text>
  </xsl:template>

  <xsl:template match="delivery">
    <xsl:text>- Доставка (Статус: </xsl:text>
    <xsl:value-of select="@status"/>
    <xsl:text>, Цена: </xsl:text>
    <xsl:value-of select="@cost"/>
    <xsl:text>):&#10;</xsl:text>
    <xsl:text> Продукти:&#10;</xsl:text>
    <xsl:apply-templates select="products/product"/>
    <xsl:text> Доставчик: </xsl:text>
    <xsl:value-of select="vendor/name"/>
    <xsl:text>&#10;</xsl:text>
  </xsl:template>
</xsl:stylesheet>

```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="text" encoding="UTF-8"/>

  <xsl:template match="/warehouses">
    <xsl:text>===== ДОКЛАД ЗА СКЛАДОВЕТЕ =====&#10;&#10;</xsl:text>

    Собственик: <xsl:value-of select="owner/name"/>&#10;
    Контакти:
    <xsl:apply-templates select="owner/contact"/>
    <xsl:text>&#10;&#10;</xsl:text>

    <xsl:for-each select="warehouse">
      <xsl:text>--- СКЛАД ---&#10;</xsl:text>
      <xsl:apply-templates select="contact"/>
      <xsl:text>&#10;Продукти:&#10;</xsl:text>
      <xsl:apply-templates select="products/product"/>
      <xsl:text>Доставки:&#10;</xsl:text>
      <xsl:apply-templates select="deliveries/delivery"/>
      <xsl:text>=====&#10;&#10;</xsl:text>
    </xsl:for-each>
  </xsl:template>

  <xsl:template match="contact">
    <xsl:text>Локация: </xsl:text>
    <xsl:value-of select="location/country"/>
    <xsl:text>, </xsl:text>
    <xsl:value-of select="location/city"/>
    <xsl:text>, </xsl:text>
    <xsl:value-of select="location/address"/>
    <xsl:text>&#10;Имейл: </xsl:text>
    <xsl:value-of select="email"/>
    <xsl:text>&#10;Телефон: </xsl:text>
    <xsl:value-of select="phone"/>
  </xsl:template>

  <xsl:template match="product">
    <xsl:text>- </xsl:text>
    <xsl:value-of select="@name"/>
    <xsl:text> (Марка: </xsl:text>
    <xsl:value-of select="@brand"/>
    <xsl:text>, Цена: </xsl:text>
    <xsl:value-of select="info/price"/>
    <xsl:text> </xsl:text>
    <xsl:value-of select="info/price/@currency"/>
    <xsl:text>, Количество: </xsl:text>
    <xsl:value-of select="info/quantity"/>
    <xsl:text>)&#10;</xsl:text>
  </xsl:template>
```

```
<xsl:template match="delivery">
  <xsl:text>    - Доставка (Статус: </xsl:text>
  <xsl:value-of select="@status"/>
  <xsl:text>, Цена: </xsl:text>
  <xsl:value-of select="@cost"/>
  <xsl:text>)&#10;</xsl:text>
  <xsl:text>    Продукты:&#10;</xsl:text>
  <xsl:apply-templates select="products/product"/>
  <xsl:text>    Доставчик: </xsl:text>
  <xsl:value-of select="vendor/name"/>
  <xsl:text>&#10;</xsl:text>
</xsl:template>
</xsl:stylesheet>
```

4. Заключение

Подобрения