Учреждение образования БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет непрерывного и дистанционного обучения

Специальность «Информационные системы и технологии в бизнес менеджменте»

Дисциплина: «Средства и технологии анализа и разработки информационных систем»

Отчет по контрольной работе №2

Тема : Разработка автоматизированной системы поддержки управления продажами в электронной коммерции

Выполнил: Галкин И.В.

Группа: 894351

Вариант: 21



Проверила: Унучек Т.М.

Минск 2021

**СОДЕРЖАНИЕ**

[**1. Краткие теоретические сведения** 3](#_Toc73237433)

[**2. Результаты моделирования** 4](#_Toc73237434)

[**Выводы** 5](#_Toc73237435)

[**Список использованной литературы** 14](#_Toc73237436)

# **1. Краткие теоретические сведения**

# **Краткие теоретические сведения**

XML — это текстовый формат, удобный для хранения и передачи структурированных данных. Но работать с ним напрямую сложно в любом языке программирования. Чтобы можно было легко считывать из XML-документа данные или модифицировать его, нужно вначале его преобразовать в более подходящую форму. На данный момент для решения этой задачи наиболее широко используется программная модель DOM, основанная на том, что любой корректный документ может быть представлен в виде дерева объектов. Применяется DOM и Flash-плеером. DOM (Document Object Model — объектная модель документа) была разработана комитетом W3C и реализована в большинстве языков и программных сред. Ее основной конкурент — основанная на событиях модель SAX — распространена куда меньше. Причиной столь значительной популярности DOM является ее простота и интуитивность.

Как уже говорилось, основной идеей DOM является то, что для того, чтобы описать XML- документ, нужно построить дерево объектов. Каждый объект изображает один узел (проще говоря, тег). У каждого объекта есть массив ссылок на объекты, соответствующие дочерним узлам описываемого им узла дерева XML. Они расположены в том же порядке, в котором прописаны теги, на основании которых они были заданы. У каждого объекта узла также есть свойство, ссылающееся на объект родительского узла. В особом объекте, связанном с объектом узла, в форме свойств хранятся его атрибуты.

# **2. Руководство пользователя со скриншотами**

Функциональность приложения:

1. Аутентификация
2. Администрирование
3. Оформление заказов

После запуска приложения, пользователю предлагается войти в систему одним из трех способов:

1. Войти как пользователь
2. Войти как администратор
3. Зарегистрироваться

Администратор может:

1. Просматривать пользователей системы
2. Просматривать заказы определенного пользователя
3. Просматривать заказы назначенные на него
4. Изменять статус заказов

Пользователь может:

1. Просмотреть каталог товаров
2. Добавить желаемые товары в корзину
3. Оформить заказ
4. Выбрать адрес доставки из существующих, либо добавить новый
5. Подтвердить заказ
6. В случае выхода из каталога с заполненной корзиной, у пользователя остается возможность восстановить выбранные ранее товары
7. Посмотреть оформленные на него заказы

Основное отличие данной версии приложения от предыдущей – добавления поддержки хранения данных на XML файле. При запуске приложения XML файл с данными проверяется по шаблону XSD файла, и, в случае несоответствия схемы, приложение прекращает работу.

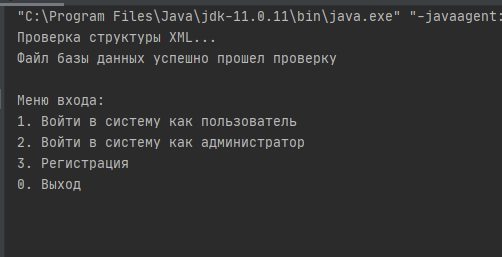


Рисунок 2.1 – Сообщение о успешном прохождении валидации XML файла

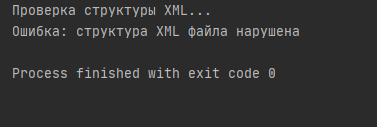


Рисунок 2.2 – Сообщение о непрохождении валидации XML файла

# 

Так как в предыдущей версии приложения все данные хранились в памяти приложения, а для текущей версии необходимо добавить поддержку XML хранилища, был реализован архитектурный паттерн Unit of Work. Благодаря этому паттерну приложение поддерживает 2 вида хранения данных – в памяти и в XML. Основной плюс этого паттерна в том, что клиентский код не зависит от того, какой вид хранилища используется в приложении.

Для того, чтобы переключить тип хранилища, необходимо поменять класс, реализующий этот паттерн. На текущий момент это делается через код, но планируется добавить поддержку конфигурационного файла.

В каждой из реализаций Unit of Work есть элементы – репозитории. Репозитории работают с конкретными сущностями и инкапсулируют все операции, проводимые над ними. Для каждой сущности есть 2 реализации репозитория – версия для xml хранилища и версия для хранения в памяти.

Такая архитектура приложения позволяет добавлять сколько угодно типов хранилищ данных. Существует интерфейс IDb, который указывает какие репозитории должны присутствовать в каждом хранилище. А в интерфейсе IRepository прописаны основные методы для работы с сущностью. Но, так как, зачастую, каждая сущность имеет свою логику работы с данными, помимо стандартных операций, репозитории для конкретных сущностей поддерживают свои интерфейсы, например, IAdminRepository.

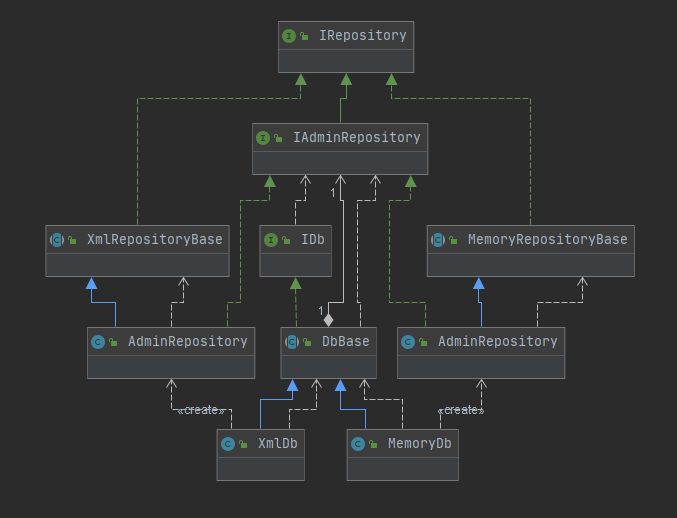


Рисунок 2.3 – Классовая диаграмма структуры хранилища

Результаты работы программы:

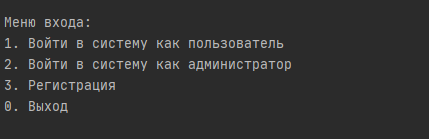


Рисунок 2.1 – Выбор способа входа в систему

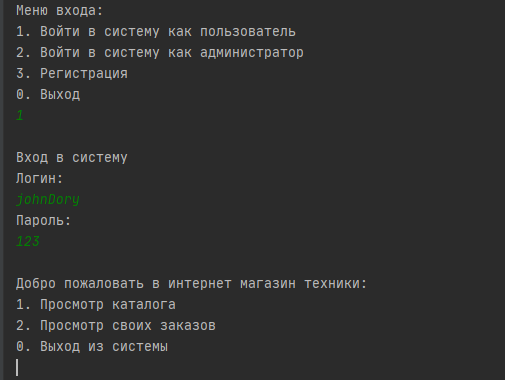


Рисунок 2.2 – Вход в систему в роли пользователя

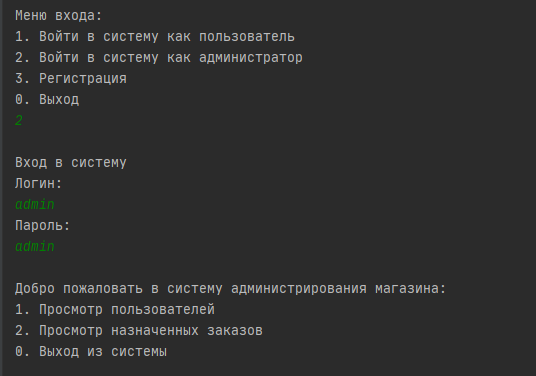


Рисунок 2.3 – Вход в систему в роли администратор

# 

Рисунок 2.4 – Добавление товара в корзину

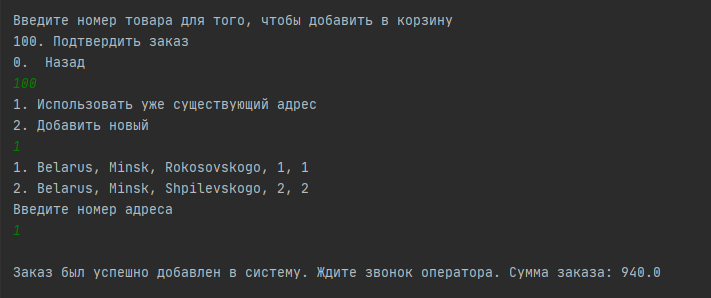


Рисунок 2.5 – Оформление заказа

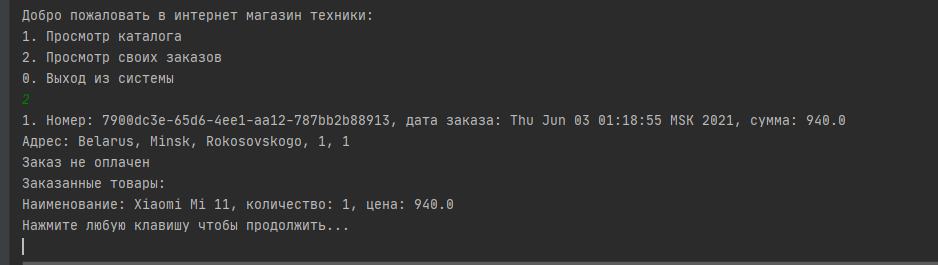


Рисунок 2.6 – Просмотр добавленного заказа

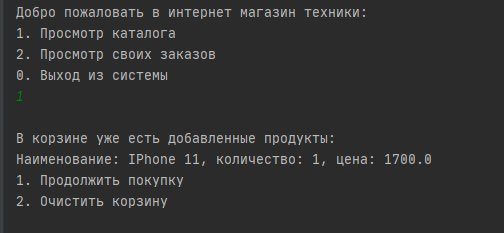


Рисунок 2.7 – Предложение восстановить добавленные заранее в корзину товары

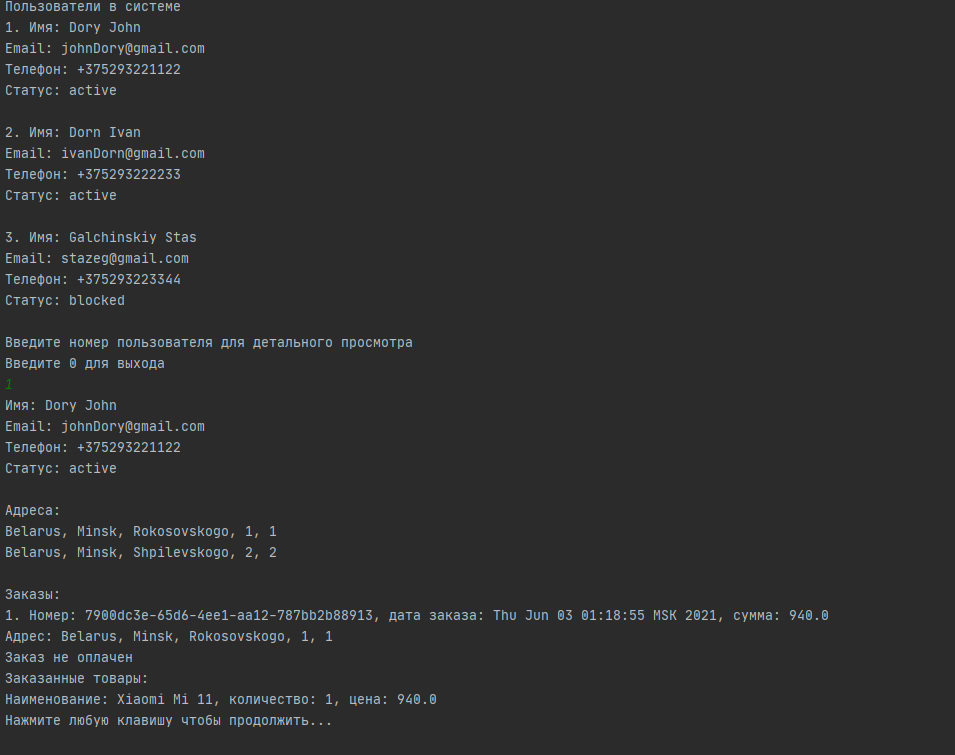


Рисунок 2.8 – Просмотр пользователей в системе и детальный просмотр конкретного

# 

Рисунок 2.9 – Просмотр назначенных на администратора заказов и изменение статуса конкретного

В результате проведенной операции, появилась информация об оплате заказа. Позже, можно будет добавить указывание конкретной информации об оплате.

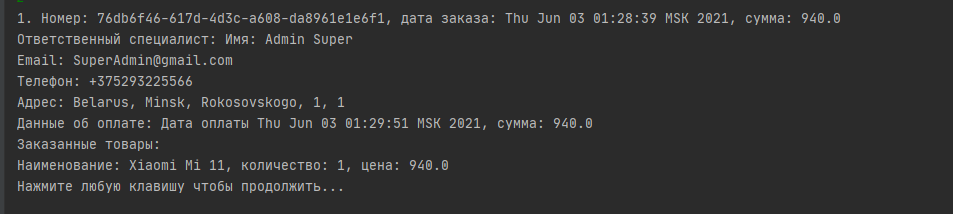


Рисунок 2.10 – Проверка результатов

# **Выводы**

В данной контрольной работе было реализовано хранилище на основе XML файла. Помимо этого, был реализован паттерн репозиторий для всех сущностей приложения.

# **Листинг кода**

# **db.xsd**

<?xml version="1.0" encoding="utf-8"?>

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="db">

<xs:complexType>

<xs:sequence>

<xs:element name="admins">

<xs:complexType>

<xs:sequence>

<xs:element name="admin">

<xs:complexType>

<xs:sequence>

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

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

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

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

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

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

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

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="customers">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="customer">

<xs:complexType>

<xs:sequence>

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

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

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

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

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

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

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

<xs:element name="customerStatus" type="xs:integer" />

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="addresses">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="address">

<xs:complexType>

<xs:sequence>

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

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

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

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

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

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

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

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="shoppingCarts">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="shoppingCart">

<xs:complexType>

<xs:sequence>

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

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

<xs:element name="hasItems" type="xs:boolean" />

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="products">

<xs:complexType>

<xs:sequence>

<xs:element maxOccurs="unbounded" name="product">

<xs:complexType>

<xs:sequence>

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

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

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

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

<xs:element name="price" type="xs:integer" />

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="bills" >

<xs:complexType>

<xs:sequence>

<xs:element minOccurs="0" maxOccurs="unbounded" name="bill">

<xs:complexType>

<xs:sequence>

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

<xs:element name="issueDate" type="xs:date" />

<xs:element name="sum" type="xs:double" />

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

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="items" >

<xs:complexType>

<xs:sequence>

<xs:element minOccurs="0" maxOccurs="unbounded" name="item">

<xs:complexType>

<xs:sequence>

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

<xs:element name="quantity" type="xs:integer" />

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

<xs:element name="cartId" type="xs:string" minOccurs="0" />

<xs:element name="orderId" type="xs:string" minOccurs="0" />

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name="orders">

<xs:complexType>

<xs:sequence>

<xs:element minOccurs="0" maxOccurs="unbounded" name="order">

<xs:complexType>

<xs:sequence>

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

<xs:element name="orderDate" type="xs:date" />

<xs:element name="sum" type="xs:double" />

<xs:element name="billId" type="xs:string" minOccurs="0" />

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

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

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

<xs:element name="orderStatus" type="xs:integer" />

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>

# **db.xml**

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<db>

<admins>

<admin>

<id>761520e6-4b8a-40d4-9421-e1f1a0f81b2b</id>

<firstName>Super</firstName>

<lastName>Admin</lastName>

<email>SuperAdmin@gmail.com</email>

<login>admin</login>

<password>admin</password>

<phone>+375293225566</phone>

</admin>

</admins>

<customers>

<customer>

<id>f65fd5da-a250-494a-a31b-e04c0df0f880</id>

<firstName>John</firstName>

<lastName>Dory</lastName>

<email>johnDory@gmail.com</email>

<login>johnDory</login>

<password>123</password>

<phone>+375293221122</phone>

<customerStatus>0</customerStatus>

</customer>

<customer>

<id>a35ad6a2-7b14-40ed-8fc2-7a7e946b20fa</id>

<firstName>Ivan</firstName>

<lastName>Dorn</lastName>

<email>ivanDorn@gmail.com</email>

<login>ivanDorm</login>

<password>123</password>

<phone>+375293222233</phone>

<customerStatus>0</customerStatus>

</customer>

<customer>

<id>bdc14d54-8d63-4222-a932-e4f4726e6ce7</id>

<firstName>Stas</firstName>

<lastName>Galchinskiy</lastName>

<email>stazeg@gmail.com</email>

<login>stazeg</login>

<password>123</password>

<phone>+375293223344</phone>

<customerStatus>0</customerStatus>

</customer>

</customers>

<addresses>

<address>

<id>d450cd87-06df-4e06-865f-435a0b4ac1a8</id>

<userId>f65fd5da-a250-494a-a31b-e04c0df0f880</userId>

<apartmentNumber>1</apartmentNumber>

<city>Minsk</city>

<country>Belarus</country>

<houseNumber>1</houseNumber>

<street>Rokosovskogo</street>

</address>

<address>

<id>a2c7cf75-bb6c-4f91-b74f-f05ba8f28fe8</id>

<userId>f65fd5da-a250-494a-a31b-e04c0df0f880</userId>

<apartmentNumber>2</apartmentNumber>

<city>Minsk</city>

<country>Belarus</country>

<houseNumber>2</houseNumber>

<street>Shpilevskogo</street>

</address>

<address>

<id>32661eb7-0c65-402f-b870-d7bc77e80bc0</id>

<userId>a35ad6a2-7b14-40ed-8fc2-7a7e946b20fa</userId>

<apartmentNumber>3</apartmentNumber>

<city>Minsk</city>

<country>Belarus</country>

<houseNumber>3</houseNumber>

<street>Gashkevicha</street>

</address>

<address>

<id>2aa039fd-e586-4bca-969c-10092ca5982b</id>

<userId>bdc14d54-8d63-4222-a932-e4f4726e6ce7</userId>

<apartmentNumber>4</apartmentNumber>

<city>Minsk</city>

<country>Belarus</country>

<houseNumber>4</houseNumber>

<street>Bogdanovicha</street>

</address>

</addresses>

<shoppingCarts>

<shoppingCart>

<id>19a2a032-369a-410c-be19-c674c137858e</id>

<customerId>f65fd5da-a250-494a-a31b-e04c0df0f880</customerId>

<hasItems>true</hasItems>

</shoppingCart>

<shoppingCart>

<id>fbfdb984-292f-44fb-a5fb-96970c98047c</id>

<customerId>a35ad6a2-7b14-40ed-8fc2-7a7e946b20fa</customerId>

<hasItems>false</hasItems>

</shoppingCart>

<shoppingCart>

<id>bdc14d54-8d63-4222-a932-e4f4726e6ce7</id>

<customerId>bdc14d54-8d63-4222-a932-e4f4726e6ce7</customerId>

<hasItems>false</hasItems>

</shoppingCart>

</shoppingCarts>

<products>

<product>

<id>9f72fab3-e32c-49ae-a02a-4533c0655ce2</id>

<name>IPhone 11</name>

<description>Мобильный телефон от Apple</description>

<manufacturer>Apple</manufacturer>

<price>1700</price>

</product>

<product>

<id>ff201890-6fdf-4363-8c3f-ddff15e07375</id>

<name>Xiaomi Mi 11</name>

<description>Мобильный телефон от Xiaomi</description>

<manufacturer>Xiaomi</manufacturer>

<price>940</price>

</product>

<product>

<id>eb3d6676-32f5-42a4-a2f1-efbfe1caa0b3</id>

<name>Samsung Galaxy A52</name>

<description>Мобильный телефон от Samsung</description>

<manufacturer>Samsung</manufacturer>

<price>860</price>

</product>

<product>

<id>dff613fe-dd33-4fbf-834f-36dd3a0cb91a</id>

<name>Apple iPhone 12 Pro</name>

<description>Дорогой мобильный телефон от Apple</description>

<manufacturer>Apple</manufacturer>

<price>2880</price>

</product>

<product>

<id>523ad188-6c7b-45a6-b03f-67354469c748</id>

<name>Huawei P40</name>

<description>Мобильный телефон от Huawei</description>

<manufacturer>Huawei</manufacturer>

<price>600</price>

</product>

</products>

<bills />

<items>

<item>

<id>aaac936c-2170-4132-8013-878332cbf81a</id>

<quantity>2</quantity>

<productId>9f72fab3-e32c-49ae-a02a-4533c0655ce2</productId>

<cartId>19a2a032-369a-410c-be19-c674c137858e</cartId>

<orderId />

</item>

</items>

<orders />

</db>

# **Список использованной литературы**

[1] Паттерн репозиторий [Электронный ресурс]: – Режим доступа: https://habr.com/ru/post/248505/