

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»
Институт компьютерных наук и технологий

Отчет о прохождении учебной практики

Любченкова Анна Алексеевна
(Ф.И.О. обучающегося)

2 курс, гр.23531/5
(номер курса обучения и учебной группы)

09.03.01 «Информатика и вычислительная техника »
(Направление подготовки (код и наименование))

Место прохождения практики: кафедра КСПТ ИКНТ ФГАОУ ВО
«СПбПУ»

(указывается наименование профильной организации или наименование
структурного подразделения)

Сроки практики: с 24 июня по 20 июля 2019 г.

Руководитель практики от ФГАОУ ВО «СПбПУ»:

Петров М.А., ассистент каф. КСПТ ИКНТ
(Ф.И.О., уч.степень, должность)

Оценка (зачет) : _____

Руководитель практики
от ФГАОУ ВО «СПбПУ»::

Петров М.А.

Обучающийся:

Любченкова А.А.

Дата:

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»
Институт компьютерных наук и технологий

**ИНДИВИДУАЛЬНЫЙ ПЛАН (ЗАДАНИЕ И ГРАФИК)
ПРОВЕДЕНИЯ ПРАКТИКИ**

Ф.И.О. обучающегося Любченкова Анна Алексеевна

Направление подготовки (код/наименование): 09.03.01, «Информатика и
вычислительная техника»

Профиль (код/наименование): 09.03.01

Вид практики: учебная

Тип практики: ознакомительная

Место прохождения практики: кафедра КСПТ ИКНТ ФГАОУ ВО «СПбПУ»

Руководитель практики от ФГАОУ ВО «СПбПУ»:

Петров М.А., ассистент каф. КСПТ ИКНТ
(Ф.И.О., уч.степень, должность)

Рабочий график проведения практики

Сроки практики: с **24.06.2019** г. по **20.07.2019** г.

№ п/п	Этапы (периоды) практики	Вид работ	Сроки прохождения этапа (периода) практики
1	Организационный этап	Установочная лекция для разъяснения целей, задач, содержания и порядка прохождения практики, выдача сопроводительных документов по практике	24.06.2019 г.
2	Основной этап	Содержание практики: 1. Изучение базы данных Microsoft SQL и анализ работы графического интерфейса на языке C# 2. Обеспечение работоспособности и функциональности базы данных. Создание общего интерфейса программы, с помощью графического интерфейса на языке C#.	25.06.2019 - 16.07.2019 г.
3	Заключительный этап	Планируемые результаты прохождения практики: (Проект, реализующий функционирование магазина косметики с помощью базы данных и графического интерфейса пользователя) Подготовка отчета Промежуточная аттестация (зачет)	17.07.- 19.07.2019 г. 19.07 - 20.07.2019 г.

Обучающийся

Любченкова А.А.

Руководитель практики
от ФГАОУ ВО «СПбПУ»

Петров М.А.

Содержание

Введение	4
Цель проекта	4
Задачи	4
Анализ предметной области.....	4
Функции групп пользователей	4
Хранимые данные.....	5
Проектирование схемы данных	6
Реализация базы данных в среде SQL Server	6
Скрипт создания базы.....	6
Создание таблиц	8
Разработка представлений и хранимых процедур	12
Представления	12
Процедуры	14
Разработка клиентского приложения.....	17
Форма «Лог-ин».....	17
Форма «Новый клиент».....	17
Форма «Anna’s cosmetics»	18
<i>Вкладка «Товары».....</i>	<i>18</i>
<i>Вкладка «Заказы»</i>	<i>19</i>
<i>Вкладка «Списки»</i>	<i>21</i>
Тестирование.....	21
Создание нового заказа	21
Обработка заказа.....	22
Продажа заказа	24
Удаление клиента	25
Заключение.....	26
Список литературы	27

Введение

Данный документ является описанием проекта, сделанного мной и находящегося в облачном хранилище в открытом доступе по адресу:

https://github.com/Anna-Lyubchenkova/practice_project

Цель проекта

Целью проекта является изучение и освоение баз данных с помощью Microsoft SQL Management Studio и создание графического интерфейса для связи с базой данных.

Задачи

Разработать и провести тестирование базы данных «косметического магазина», а также многопользовательского приложения для взаимодействия с базой. Базу создать в среде MSSQL, имея следующие группы пользователей:

№ пп	Наименование пользователя
1	Продавец (Иванов)
2	Покупатель (Зарецкая)

Анализ предметной области

Функции групп пользователей

№ пп	Выполняемая функция	Входные данные	Выходные данные	Функции, которые должны быть реализованы в ИС
<i>Первый пользователь (продавец)</i>				
1	Информация о товаре	Наименование товара	Наличие товара, цена товара	Выдача формы информации о товаре в виде списка
2	Продажа товаров	Товары, кол-во	Сумма покупки	Выдать накладной лист, вычесть из единиц товара в магазине кол-во проданного товара, поставить статус ready. Если товара недостаточно, то увеличить число единиц товара в Reserved. На товары в заказе, готовые – ставим ready, на ожидаемые – wait. Статус всего заказа ставится пометкой «исполнение». (new->wait->ready)
<i>Второй пользователь (покупатель)</i>				
1	Наличие	Наименование	Наличие	Выдать запрос с краткой инфо о

	товара в салоне	товара	товара	товаре + наличие
2	Просмотр заказов	Идентификатор пользователя	Список заказов	Выдача списка заказов пользователя с общей суммой каждого заказа и статусом заказа

Хранимые данные

№	Хранимые данные	Пользователи, которым разрешен доступ	Ограничения по типу и значению
1	Наименование товара, цена товара	Все	Наименование – строка до 20 символов, цена – в рублях или валюте
2	Наличие товара в салоне	Продавец, Покупатель	Возможность продажи витринного товара
3	Список сотрудников	Администратор	Имя, Фамилия, должность, зарплата и т.д.
4	Данные формы заказа, время, кол-во товаров и т.д.	Продавец	Кол-во единицы каждого товара в магазине, резерве и цена за единицу
5	Список клиентов	Продавец, Администратор	Имя, Фамилия – строки. Телефон или email – обязательно иметь одно поле контактов.

Проектирование схемы данных

Реализованные таблицы:

- Basket (корзина, временная таблица)
- ClientOrder
- Clients
- ItemListToOrder
- Items
- Post
- Staff

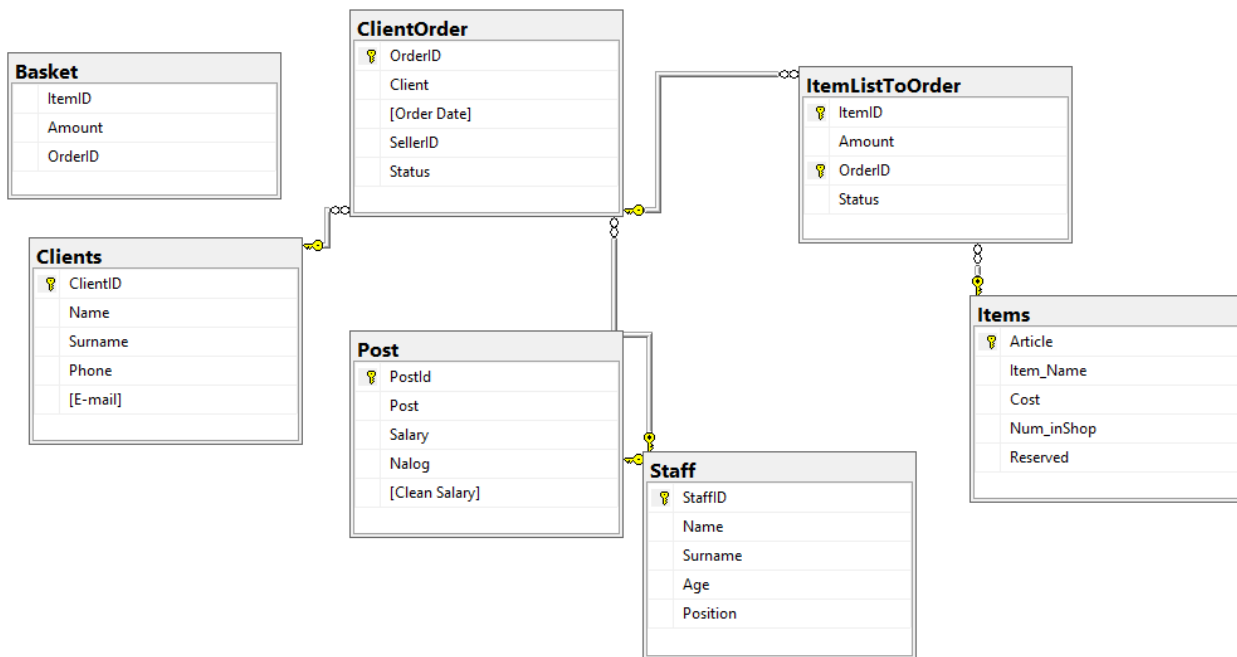


Рисунок 1 Диаграмма базы данных

Реализация базы данных в среде SQL Server

Скрипт создания базы

```
USE [master]
GO
CREATE DATABASE [practice_cosmetics_shop]
GO
ALTER DATABASE [practice_cosmetics_shop] SET COMPATIBILITY_LEVEL = 110
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [practice_cosmetics_shop].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [practice_cosmetics_shop] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET ANSI_NULLS OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET ANSI_PADDING OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET ARITHABORT OFF
GO
```

```
ALTER DATABASE [practice_cosmetics_shop] SET AUTO_CLOSE ON
GO
ALTER DATABASE [practice_cosmetics_shop] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [practice_cosmetics_shop] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [practice_cosmetics_shop] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET DISABLE_BROKER
GO
ALTER DATABASE [practice_cosmetics_shop] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [practice_cosmetics_shop] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET RECOVERY SIMPLE
GO
ALTER DATABASE [practice_cosmetics_shop] SET MULTI_USER
GO
ALTER DATABASE [practice_cosmetics_shop] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [practice_cosmetics_shop] SET DB_CHAINING OFF
GO
ALTER DATABASE [practice_cosmetics_shop] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [practice_cosmetics_shop] SET TARGET_RECOVERY_TIME = 0 SECONDS
GO
ALTER DATABASE [practice_cosmetics_shop] SET DELAYED_DURABILITY = DISABLED
GO
ALTER DATABASE [practice_cosmetics_shop] SET QUERY_STORE = OFF
```

Создание таблиц

1. Basket (корзина, временная таблица)

```
USE [practice_cosmetics_shop]
GO

CREATE TABLE [dbo].[Basket](
    [ItemID] [int] NOT NULL,
    [Amount] [int] NOT NULL,
    [OrderID] [int] NOT NULL
) ON [PRIMARY]
GO
```

2. ClientOrder

```
USE [practice_cosmetics_shop]
GO

CREATE TABLE [dbo].[ClientOrder](
    [OrderID] [int] IDENTITY(1,1) NOT NULL,
    [Client] [int] NOT NULL,
    [Order Date] [date] NOT NULL,
    [SellerID] [int] NOT NULL,
    [Status] [varchar](5) NULL,
    CONSTRAINT [PK_ClientOrder] PRIMARY KEY CLUSTERED
(
    [OrderID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[ClientOrder] ADD CONSTRAINT [DF_ClientOrder_Status] DEFAULT
(N'New') FOR [Status]
GO

ALTER TABLE [dbo].[ClientOrder] WITH CHECK ADD CONSTRAINT [FK_ClientOrder_Clients]
FOREIGN KEY([Client])
REFERENCES [dbo].[Clients] ([ClientID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO

ALTER TABLE [dbo].[ClientOrder] CHECK CONSTRAINT [FK_ClientOrder_Clients]
GO

ALTER TABLE [dbo].[ClientOrder] WITH CHECK ADD CONSTRAINT [FK_ClientOrder_Staff]
FOREIGN KEY([SellerID])
REFERENCES [dbo].[Staff] ([StaffID])
GO

ALTER TABLE [dbo].[ClientOrder] CHECK CONSTRAINT [FK_ClientOrder_Staff]
GO
```

3. Clients

```
USE [practice_cosmetics_shop]
GO

CREATE TABLE [dbo].[Clients](
    [ClientID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [varchar](20) NOT NULL,
    [Surname] [varchar](25) NOT NULL,
    [Phone] [varchar](10) NULL,
    [E-mail] [varchar](25) NULL,
```



```

CONSTRAINT [PK_Clients] PRIMARY KEY CLUSTERED
(
    [ClientID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Clients] ADD CONSTRAINT [DF_Clients_Phone] DEFAULT ('-') FOR
[Phone]
GO

ALTER TABLE [dbo].[Clients] ADD CONSTRAINT [DF_Clients_E-mail] DEFAULT ('-') FOR [E-
mail]
GO

```

4. ItemListToOrder

```

USE [practice_cosmetics_shop]
GO

CREATE TABLE [dbo].[ItemListToOrder](
    [ItemID] [int] NOT NULL,
    [Amount] [int] NOT NULL,
    [OrderID] [int] NOT NULL,
    [Status] [varchar](5) NULL,
    CONSTRAINT [PK_ItemListToOrder] PRIMARY KEY CLUSTERED
(
    [ItemID] ASC,
    [OrderID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[ItemListToOrder] ADD CONSTRAINT [DF_ItemListToOrder_Status]
DEFAULT (N'New') FOR [Status]
GO

ALTER TABLE [dbo].[ItemListToOrder] WITH CHECK ADD CONSTRAINT
[FK_ItemListToOrder_ClientOrder] FOREIGN KEY([OrderID])
REFERENCES [dbo].[ClientOrder] ([OrderID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO

ALTER TABLE [dbo].[ItemListToOrder] CHECK CONSTRAINT [FK_ItemListToOrder_ClientOrder]
GO

ALTER TABLE [dbo].[ItemListToOrder] WITH CHECK ADD CONSTRAINT
[FK_ItemListToOrder_Items] FOREIGN KEY([ItemID])
REFERENCES [dbo].[Items] ([Article])
GO

ALTER TABLE [dbo].[ItemListToOrder] CHECK CONSTRAINT [FK_ItemListToOrder_Items]
GO

```

5. Post

```

USE [practice_cosmetics_shop]
GO

CREATE TABLE [dbo].[Post](
    [PostId] [int] IDENTITY(1,1) NOT NULL,
    [Post] [nchar](15) NOT NULL,
    [Salary] [money] NULL,

```

```

        [Nalog] [int] NULL,
        [Clean Salary] AS ([Salary]-([Salary]*[Nalog])/(100)),
        CONSTRAINT [PK_Post] PRIMARY KEY CLUSTERED
    (
        [PostId] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]
GO

```

6. Items

```

USE [practice_cosmetics_shop]
GO

CREATE TABLE [dbo].[Items](
    [Article] [int] IDENTITY(1,1) NOT NULL,
    [Item_Name] [nvarchar](20) NOT NULL,
    [Cost] [money] NOT NULL,
    [Num_inShop] [int] NOT NULL,
    [Reserved] [int] NOT NULL,
    CONSTRAINT [PK_Items] PRIMARY KEY CLUSTERED
    (
        [Article] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Items] ADD CONSTRAINT [DF_Items_Num_inShop] DEFAULT ((0)) FOR
[Num_inShop]
GO

ALTER TABLE [dbo].[Items] ADD CONSTRAINT [DF_Items_Reserved] DEFAULT ((0)) FOR
[Reserved]
GO

ALTER TABLE [dbo].[Items] WITH CHECK ADD CONSTRAINT [inShop] CHECK
(([Num_inShop]>=(0)))
GO

ALTER TABLE [dbo].[Items] CHECK CONSTRAINT [inShop]
GO

EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Не может быть <0 в
магазине' , @level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'Items',
@level2type=N'CONSTRAINT',@level2name=N'inShop'
GO

```

7. Staff

```

USE [practice_cosmetics_shop]
GO

CREATE TABLE [dbo].[Staff](
    [StaffID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [varchar](25) NOT NULL,
    [Surname] [varchar](20) NOT NULL,
    [Age] [int] NOT NULL,
    [Position] [int] NOT NULL,
    CONSTRAINT [PK_Staff] PRIMARY KEY CLUSTERED
    (
        [StaffID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

```

```
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Staff] WITH CHECK ADD CONSTRAINT [FK_Staff_Post] FOREIGN
KEY([Position])
REFERENCES [dbo].[Post] ([PostId])
GO

ALTER TABLE [dbo].[Staff] CHECK CONSTRAINT [FK_Staff_Post]
GO
```

Разработка представлений и хранимых процедур

Ниже перечислены основные представления и функции со скриптами создания и задачами.

Представления

1) Для отображения пользователю

<pre>/* ПРЕДСТАВЛЕНИЕ КЛИЕНТОВ, КОТОРЫЕ СДЕЛАЛИ ХОТЯ БЫ ОДИН ЗАКАЗ */ USE [practice_cosmetics_shop] GO CREATE VIEW [dbo].[Active_clients] AS SELECT dbo.Clients.Name, dbo.Clients.Surname, COUNT(dbo.ClientOrder.OrderID) AS 'Orders' FROM dbo.Clients INNER JOIN dbo.ClientOrder ON dbo.Clients.ClientID = dbo.ClientOrder.Client GROUP BY dbo.Clients.Name, dbo.Clients.Surname GO</pre>
<pre>/* ПРЕДСТАВЛЕНИЕ ВСЕХ КЛИЕНТОВ */ USE [practice_cosmetics_shop] GO Create View [dbo].[Clients_view] As Select * FROM Clients; GO</pre>
<pre>/* ПРЕДСТАВЛЕНИЕ СПИСКА ВСЕХ ЗАКАЗОВ. ДЛЯ КАТЕГОРИЙ - АНАЛОГИЧНО */ USE [practice_cosmetics_shop] GO Create View [dbo].[Clients_Orders_all] As Select OrderID, ClientID, Clients.Surname, Clients.Name, [Order Date], Staff.Surname as 'Staff', Status FROM ClientOrder JOIN Clients ON Clients.ClientID = ClientOrder.Client JOIN Staff ON ClientOrder.SellerID = Staff.StaffID Where ClientOrder.Status = ANY (Select Status FROM ClientOrder); GO</pre>
<pre>/* ПРЕДСТАВЛЕНИЕ СПИСКА ТОВАРОВ*/ USE [practice_cosmetics_shop] GO CREATE VIEW [dbo].[Product_List] AS SELECT Article, Item_Name, Cost AS [cost, RUB], Num_inShop FROM dbo.Items GO</pre>
<pre>/* ПРЕДСТАВЛЕНИЕ СПИСКА НОВЫХ ЗАКАЗОВ */ USE [practice_cosmetics_shop] GO Create View [dbo].[Clients_Orders_new] As Select OrderID, ClientID, Clients.Surname, Clients.Name, [Order Date], Staff.Surname as 'Staff', Status FROM ClientOrder JOIN Clients ON Clients.ClientID = ClientOrder.Client</pre>

<pre> JOIN Staff ON ClientOrder.SellerID = Staff.StaffID Where ClientOrder.Status = 'New'; GO </pre>
<pre> /* ПРЕДСТАВЛЕНИЕ СПИСКА ГОТОВЫХ К ВЫДАЧЕ ЗАКАЗОВ */ USE [practice_cosmetics_shop] GO Create View [dbo].[Clients_Orders_ready] As Select OrderID, ClientID, Clients.Surname, Clients.Name, [Order Date], Staff.Surname as 'Staff', Status FROM ClientOrder JOIN Clients ON Clients.ClientID = ClientOrder.Client JOIN Staff ON ClientOrder.SellerID = Staff.StaffID Where ClientOrder.Status = 'Ready'; GO </pre>
<pre> /* ПРЕДСТАВЛЕНИЕ СПИСКА ЗАКАЗОВ НА ИСПОЛНЕНИИ */ USE [practice_cosmetics_shop] GO Create View [dbo].[Clients_Orders_wait] As Select OrderID, ClientID, Clients.Surname, Clients.Name, [Order Date], Staff.Surname as 'Staff', Status FROM ClientOrder JOIN Clients ON Clients.ClientID = ClientOrder.Client JOIN Staff ON ClientOrder.SellerID = Staff.StaffID Where ClientOrder.Status = 'Wait'; GO </pre>
<pre> /* ПРЕДСТАВЛЕНИЕ СПИСКА ЗАВЕРШЕННЫХ ЗАКАЗОВ */ USE [practice_cosmetics_shop] GO Create View [dbo].[Clients_Orders_sold] As Select OrderID, ClientID, Clients.Surname, Clients.Name, [Order Date], Staff.Surname as 'Staff', Status FROM ClientOrder JOIN Clients ON Clients.ClientID = ClientOrder.Client JOIN Staff ON ClientOrder.SellerID = Staff.StaffID Where ClientOrder.Status = 'Sold'; GO </pre>
<pre> /* ПРЕДСТАВЛЕНИЕ СОТРУДНИКОВ */ USE [practice_cosmetics_shop] GO Create View [dbo].[Staff_view] As Select StaffID, Surname, Name, Age, Position, Post, Salary FROM Staff,Post Where Staff.Position = Post.PostId; GO </pre>

2) Для системных задач и обработки

<pre> /*ПРЕДСТАВЛЕНИЕ ДЛЯ ОБРАБОТКИ ЗАКАЗОВ В АВТОМАТИЧЕСКОМ РЕЖИМЕ СИСТЕМОЙ*/ USE [practice_cosmetics_shop] GO </pre>

```

Create View [dbo].[Ordered_Items]
As
    Select OrderID, ItemID, Amount, Num_inShop, Reserved,
ItemLstToOrder.Status
    From Items JOIN ItemLstToOrder ON Items.Article =
ItemLstToOrder.ItemID
GO

```

Процедуры

```

USE [practice_cosmetics_shop]
GO

CREATE Procedure [dbo].[Product_Info] (@product_n as int)
As
    /*запрос на получение информации по товару от продавца. наименование -
Входное данное*/
    Select Item_Name, Cost as 'cost, RUB', Num_inShop, Reserved
    FROM Items
    Where Article=@product_n;

```

```

USE [practice_cosmetics_shop]
GO

CREATE Procedure [dbo].[Product_Info_Client] (@product_n as nchar(20))
As
    /*запрос на информацию о товаре от покупателя. Входное данное -
наименование*/
    Select Item_Name, Cost as 'cost, RUB', Num_inShop
    FROM Items
    Where Item_name=@product_n;

```

```

/*перевести все созданные и готовые к продаже товары к состоянию Sold*/

USE [practice_cosmetics_shop]
GO

CREATE Procedure [dbo].[Seller_sold_order] (@order_num as int)
AS
    UPDATE Ordered_Items SET Status = 'Sold'
    Where Ordered_Items.OrderID = @order_num
    UPDATE ClientOrder SET Status = 'Sold'
    Where ClientOrder.OrderID = @order_num

```

```

USE [practice_cosmetics_shop]
GO

CREATE Procedure [dbo].[Product_in_Range] (@a_range as int, @b_range as int)
AS
    /*запрос о выдаче списка товаров, лежащих в диапазоне цен, можно добавить
товары
    только по акции через параметр. Входное данное - границы диапазона.*/
    Select Item_Name, Cost , Num_inShop as 'Available'
    FROM Items
    Where CONVERT(int,Cost) Between @a_range And @b_range
    Order By Cost,Item_Name;

```

```

/*добавить в магазин товар, который был зарезервирован*/
USE [practice_cosmetics_shop]
GO

CREATE Procedure [dbo].[Seller_upd_order] (@order_num as int)
AS
    UPDATE ClientOrder SET Status = 'Wait'
    Where ClientOrder.OrderID = @order_num
    UPDATE Ordered_Items SET Reserved = Reserved + Amount
    Where Amount > Num_inShop AND Status = 'New' AND OrderID =

```

```

@order_num
        UPDATE Ordered_Items SET Status = 'Wait'
        Where Amount > Num_inShop AND Status IN ('New','Wait') AND OrderID
= @order_num
        UPDATE Ordered_Items Set Status = 'PreR'
        Where Amount <= Num_inShop AND Status IN ('New','Wait') AND OrderID
= @order_num
        UPDATE Ordered_Items SET Num_inShop = Num_inShop - Amount
        Where Status = 'PreR' AND OrderID = @order_num
        UPDATE Ordered_Items SET Status = 'Ready'
        Where Status = 'PreR' AND OrderID = @order_num

```

```

USE [practice_cosmetics_shop]
GO

CREATE Procedure [dbo].[Check_Order] (@orderid as int, @sort as varchar(10))
AS
        /*Запрос на вывод товаров в заказе по номеру заказа. Входное данное -
номер заказа*/
        if (@sort = 'Item_Name')
        Select Item_Name, Amount,
        Items.Cost * Amount as 'Summa', ItemListToOrder.Status
        FROM ClientOrder Join ItemListToOrder on ClientOrder.OrderID
= ItemListToOrder.OrderID
        JOIN Items ON Items.Article = ItemListToOrder.ItemID
        Where ClientOrder.OrderID = @orderid
        Order By Item_Name;

        if (@sort = 'Summa')
        Select Item_Name, Amount,
        Items.Cost * Amount as 'Summa', ItemListToOrder.Status
        FROM ClientOrder Join ItemListToOrder on ClientOrder.OrderID
= ItemListToOrder.OrderID
        JOIN Items ON Items.Article = ItemListToOrder.ItemID
        Where ClientOrder.OrderID = @orderid
        Order By 'Summa';

        if (@sort = 'Status')
        Select Item_Name, Amount,
        Items.Cost * Amount as 'Summa', ItemListToOrder.Status
        FROM ClientOrder Join ItemListToOrder on ClientOrder.OrderID
= ItemListToOrder.OrderID
        JOIN Items ON Items.Article = ItemListToOrder.ItemID
        Where ClientOrder.OrderID = @orderid
        Order By ItemListToOrder.Status;

```

```

USE [practice_cosmetics_shop]
GO

CREATE Procedure [dbo].[Reserved_Items]
As
        /*Запрос на вывод товаров, которые в данный момент в резерве*/
        Select Item_Name, Items.Reserved FROM Items
        Where Reserved > 0;

```

```

USE [practice_cosmetics_shop]
GO

CREATE Procedure [dbo].[View_Orders] (@client as int, @mode as int = 0)
AS
        /*Запрос на вывод заказов по данному клиенту, комбинации по фильтру в кач.
переменной
        Входное данное - номер клиента*/
        If (@mode = 0) /*вывести все заказы*/
        Select OrderID, Clients.Surname, Clients.Name, [Order Date],
Staff.Surname as 'Staff', Status
        FROM ClientOrder JOIN Clients ON Clients.ClientID =
ClientOrder.Client
        JOIN Staff ON ClientOrder.SellerID = Staff.StaffID
        Where Client = @client AND ClientOrder.Status = ANY

```

```

(Select Status FROM ClientOrder);
    IF (@mode = 1) /*только созданные заказы*/
        Select OrderID, Clients.Surname, Clients.Name, [Order Date],
        Staff.Surname as 'Staff', Status
        FROM ClientOrder JOIN Clients ON Clients.ClientID =
ClientOrder.Client
        JOIN Staff ON ClientOrder.SellerID = Staff.StaffID
        Where Client = @client AND ClientOrder.Status = 'New'
    IF (@mode = 2) /*только заказы в резерве.При том полностью reserved и частично
reserved заказы - WAIT*/
        Select OrderID, Clients.Surname, Clients.Name, [Order Date],
        Staff.Surname as 'Staff', Status
        FROM ClientOrder,Clients, Staff
        Where Clients.ClientID = ClientOrder.Client AND
        ClientOrder.SellerID = Staff.StaffID AND Client =
@client AND
        ClientOrder.Status = 'Wait'
    IF (@mode = 4) /*только завершённые заказы*/
        Select OrderID, Clients.Surname, Clients.Name, [Order Date],
        Staff.Surname as 'Staff', Status
        FROM ClientOrder,Clients, Staff
        Where Clients.ClientID = ClientOrder.Client AND
        ClientOrder.SellerID = Staff.StaffID AND Client =
@client AND
        ClientOrder.Status = 'Sold'
    IF (@mode = 3) /*резерв и только созданные заказы */
        Select OrderID, Clients.Surname, Clients.Name, [Order Date],
        Staff.Surname as 'Staff', Status
        FROM ClientOrder JOIN Clients ON Clients.ClientID =
ClientOrder.Client
        JOIN Staff ON ClientOrder.SellerID = Staff.StaffID
        Where Client = @client
    EXCEPT
        Select OrderID, Clients.Surname, Clients.Name, [Order Date],
        Staff.Surname as 'Staff', Status
        FROM ClientOrder,Clients, Staff
        Where Clients.ClientID = ClientOrder.Client AND
        ClientOrder.SellerID = Staff.StaffID AND
        ClientOrder.Status IN ('Sold');

```


Разработка клиентского приложения

Клиентское приложение написано на языке C# и позволяет осуществлять многопользовательский доступ к базе данных.

Форма «Лог-ин»

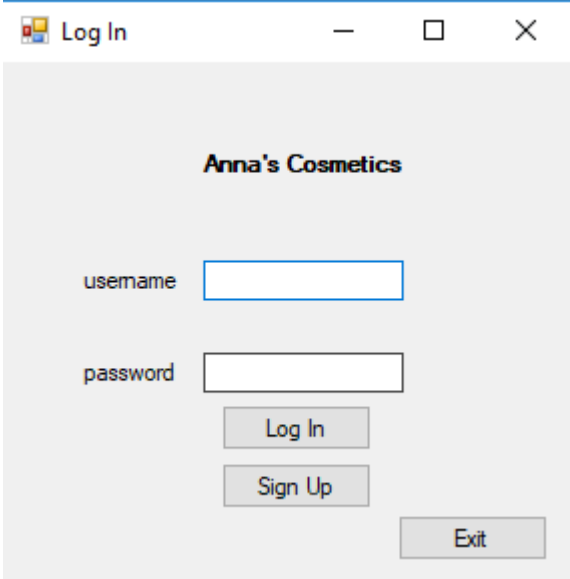


Рис. 1

Осуществляется проверка наличия пользователя в базе данных на основании сравнения логина и фамилии в таблицах. Сотрудник обязан добавлять к своей фамилии строку «_staff» для входа в систему.

Форма «Новый клиент»

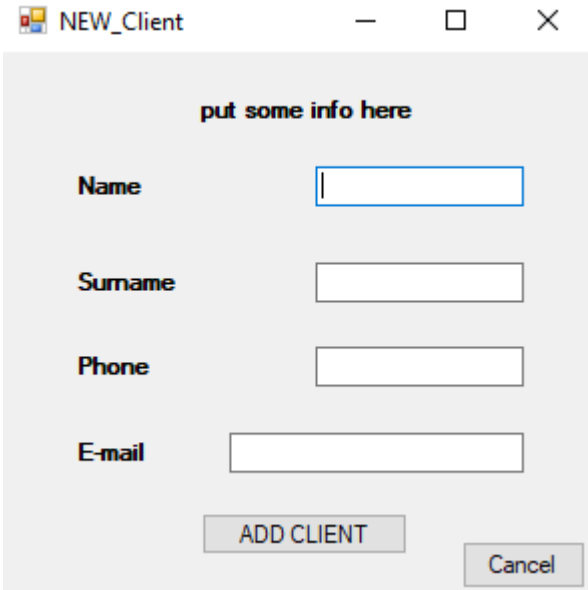


Рис. 2

Осуществляется добавление нового клиента в базу данных.

Вносить клиентов в базу может как администратор, так и сам клиент. Управлять же списком сотрудников разрешено только администратору/продавцу (которым может являться директор магазина).

Для внесения клиента происходит переход в спец. форму. Нельзя не

указывать хотя бы одного контакта!

Форма «Anna's cosmetics»

Вкладка «Товары»

	Article	Item_Name	cost, RUB	Num_inShop
▶	1	Патчи Ruby Rose	1200,0000	5
	3	Тонер Holika Holika	1025,0000	3
	4	Пенка SNP	1550,0000	2
	9	Сыворотка SNP	1700,0000	9
	10	Набор 3 товара SNP	2500,0000	4
	11	Тинт Elizavecca	250,0000	3
	12	Хайлайтер MAC	450,0000	8
	13	Пудра Evelin	600,0000	7
	14	Патчи Golden	950,0000	3
	15	Тени NYX	300,0000	2

Рис. 3 Список товаров

	Item_Name	Cost	Available
▶	Тени NYX	300,0000	2
	Хайлайтер MAC	450,0000	8
	Пудра Evelin	600,0000	7
	Патчи Golden	950,0000	3
*			

Рис. 4 Поиск в диапазоне (отображается у покупателя)

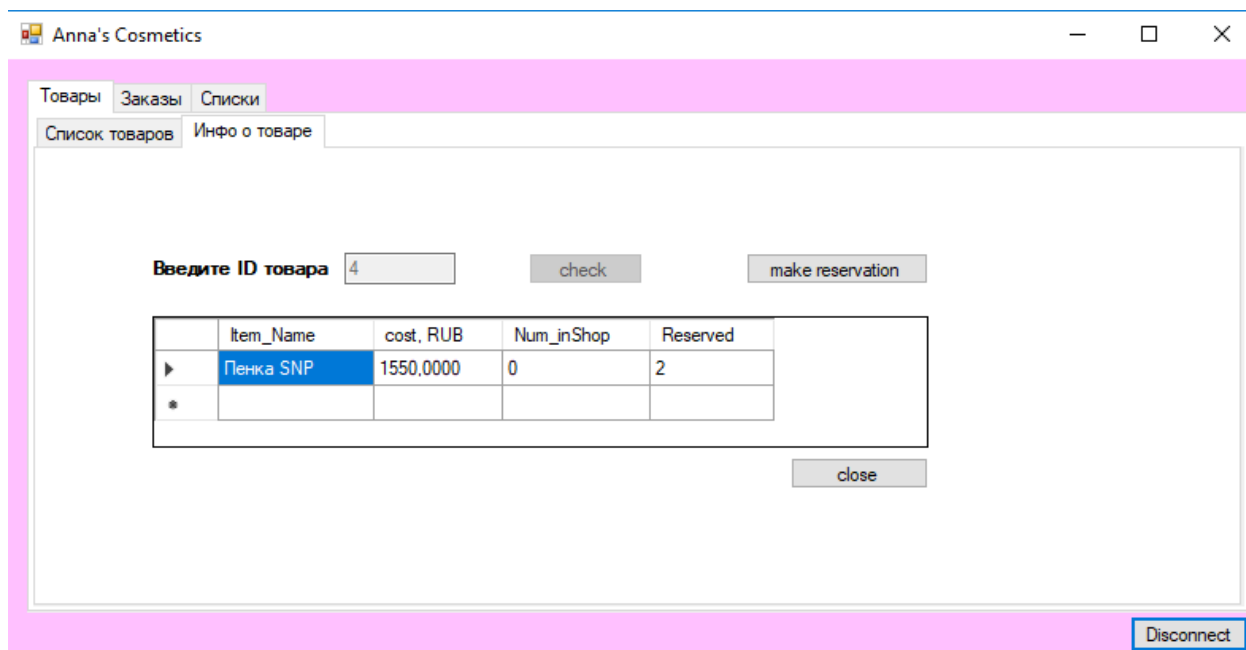


Рис. 5 Информация о товаре(отображается у продавца)

Вкладка «Заказы»

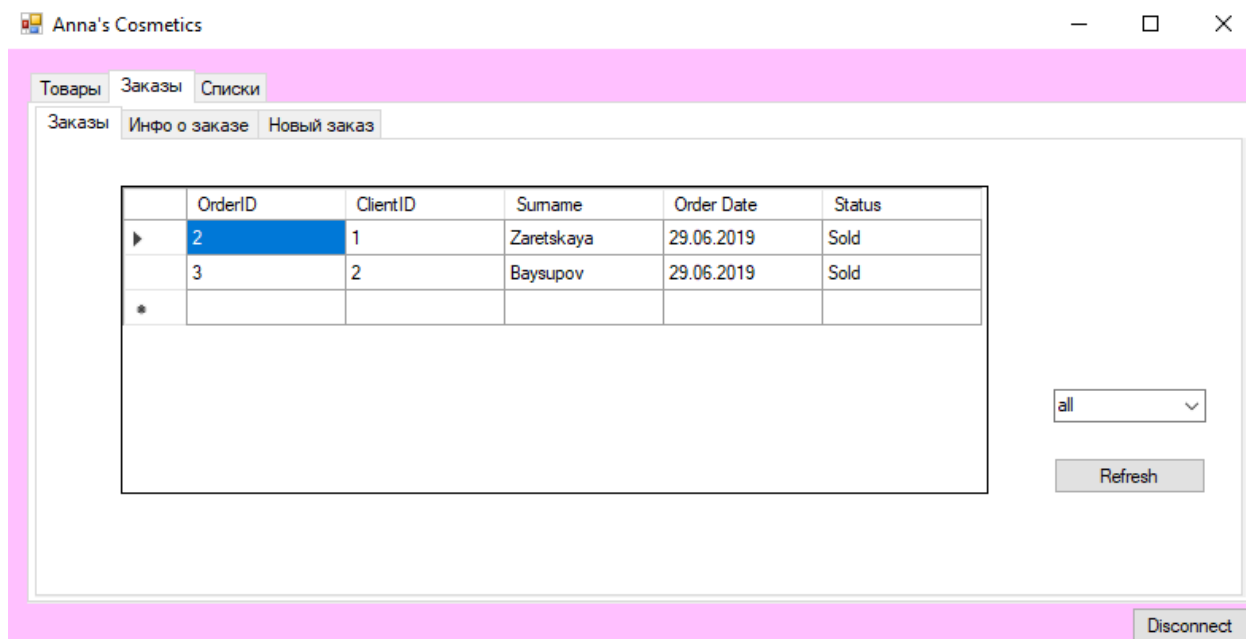


Рис. 6 Список заказов

Anna's Cosmetics

Товары Заказы Списки

Заказы Инфо о заказе Новый заказ

update state

Отсортировать по Item_Name ▾

Введите ID заказа 7

CHECK

Отменить заказ DO!

	Item_Name	Amount	Summa	Status
▶	Пенка SNP	1	1550.0000	Sold
*				

Close Order

Disconnect

Рис. 7 Информация о заказе

Anna's Cosmetics

Товары Заказы Списки

Заказы Инфо о заказе Новый заказ

Введите id клиента

1 DO

ID товара

Количество

Добавить в корзину

Корзина

	Item_Name	Amount
▶	Пенка SNP	1
*		

Завершить заказ

Очистить корзину

Отменить заказ

Disconnect

Рис. 8 Новый заказ

Вкладка «Списки»

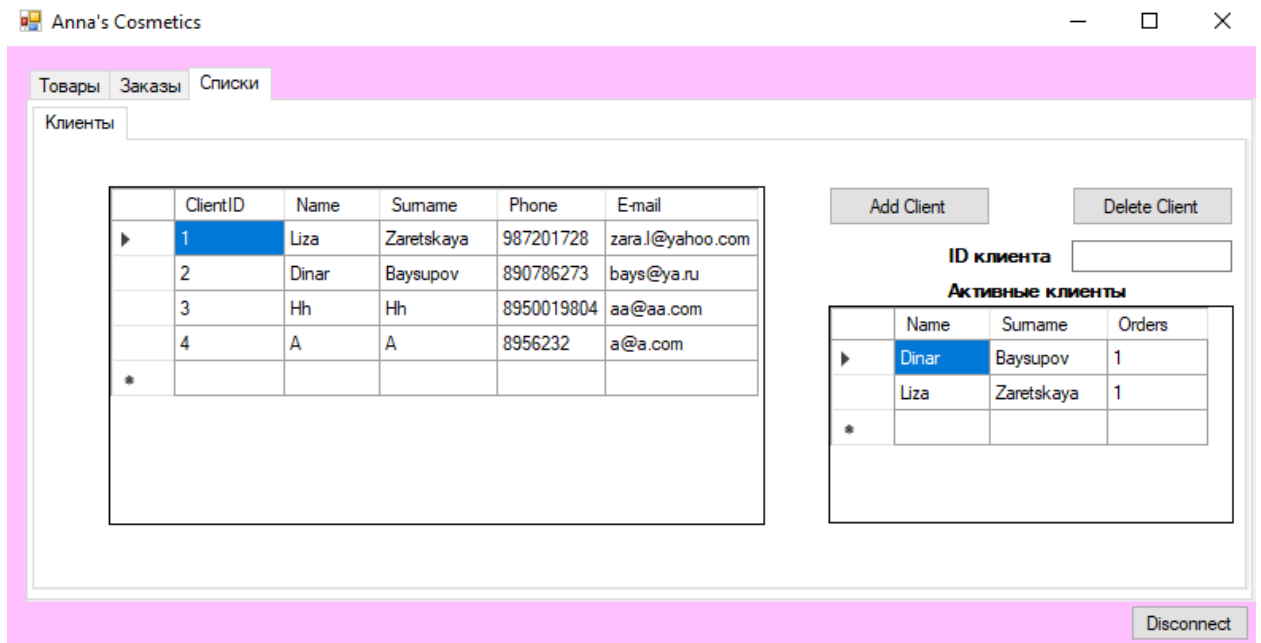


Рис. 9 Списки клиентов

Тестирование

Создание нового заказа

Для начала создания заказа необходимо ввести id клиента. Далее добавляются товары в корзину с id товара и кол-вом единиц. Корзину можно очистить. Также можно отменить создание заказа. Для создания заказа следует нажать кнопку «Завершить заказ».

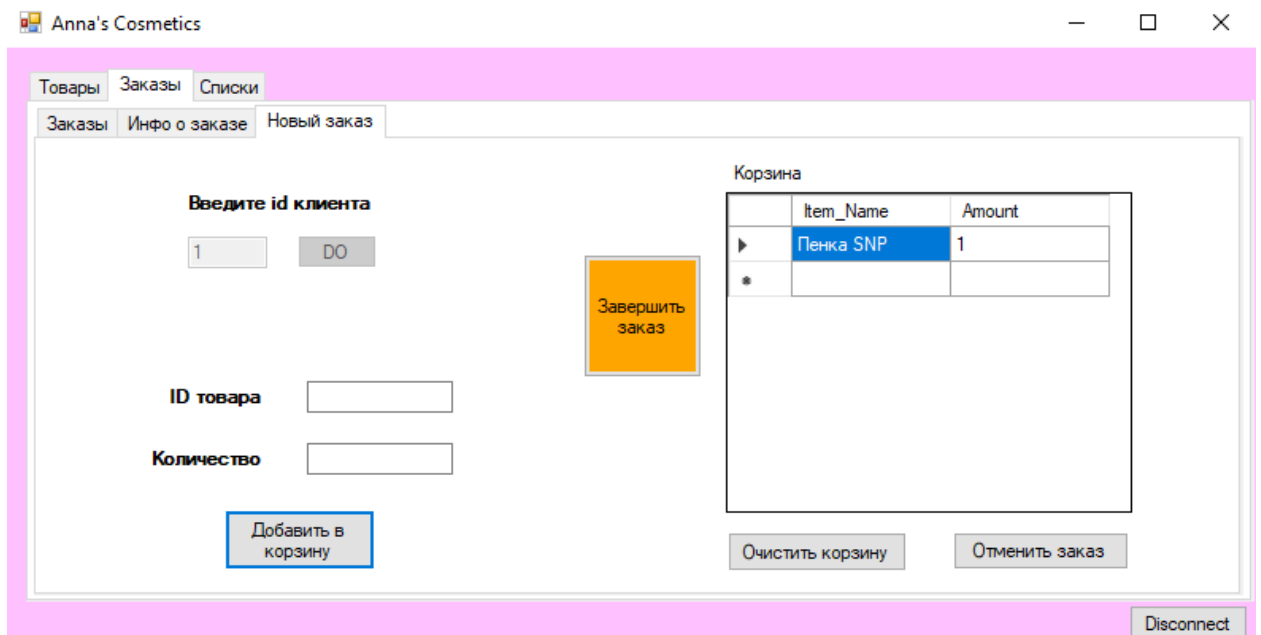


Рис. 2 Создание заказа

Обработка заказа

После создания заказа, продавец начнет его обработку. Товары, которых достаточно для продажи, будут помечены как “Ready”. Товары, которых недостаточно в магазине, можно будет добавить во вкладке товары. Процесс добавления в магазин помечается “Wait”. Обработка осуществляется с помощью кнопки “Update state”. При этом заказ можно отменить до того, как его статус будет “Ready” (Рис. 11-15)

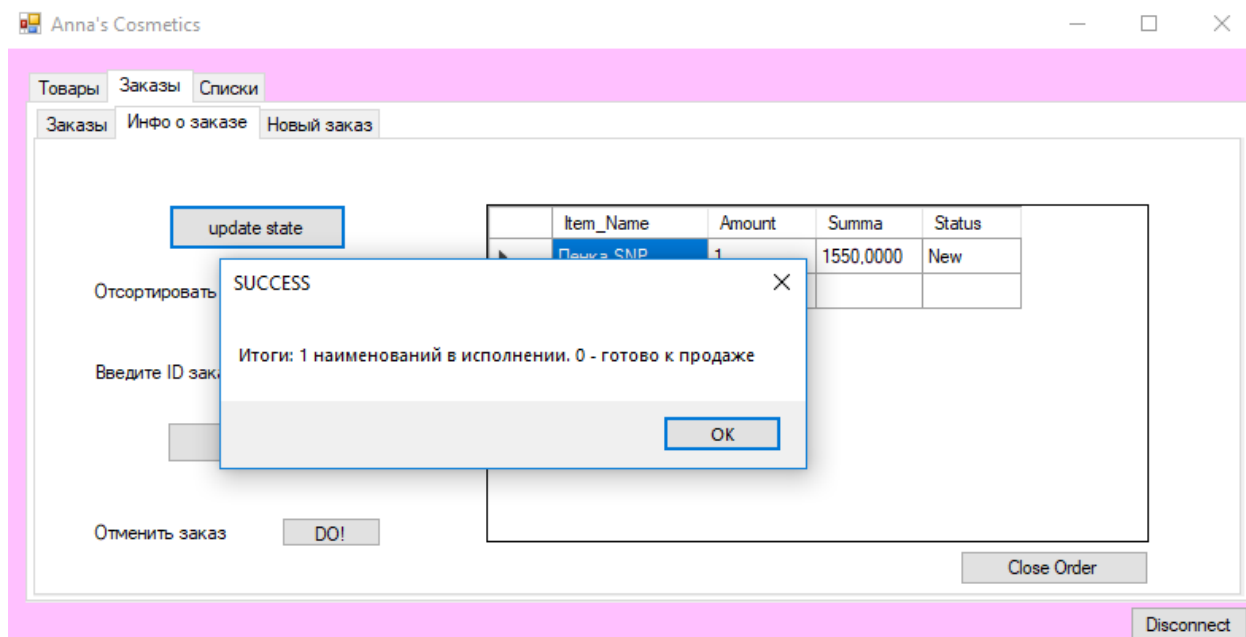


Рис. 11 Заказ в исполнении

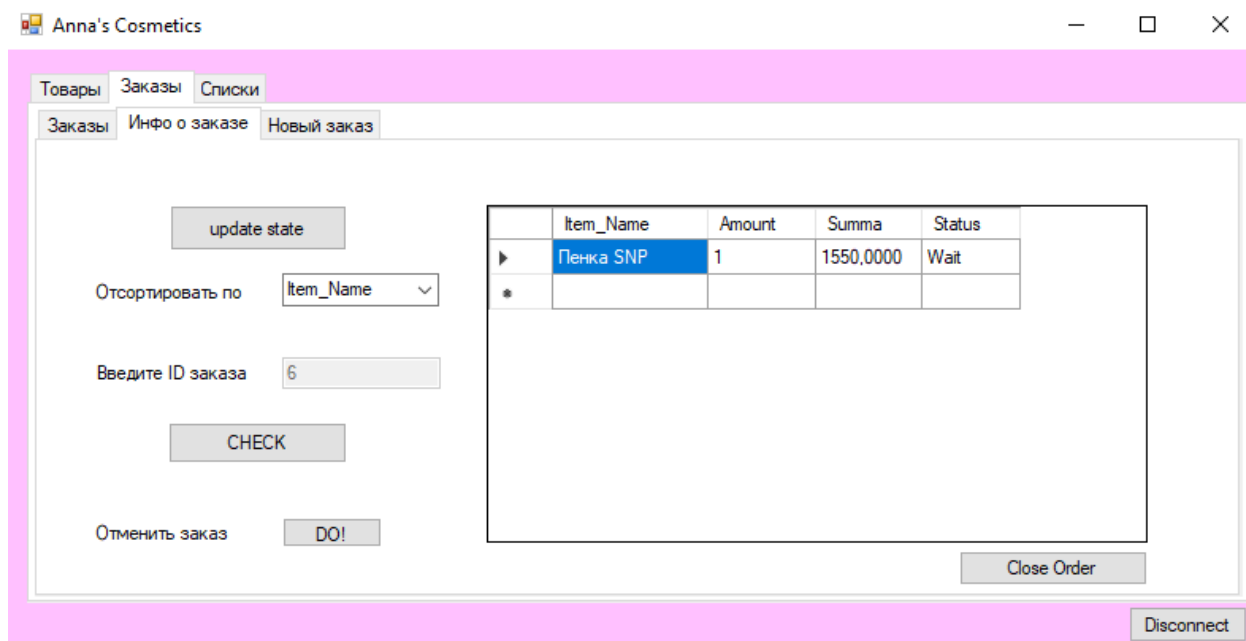


Рис. 12 Процесс добавления товара в магазин

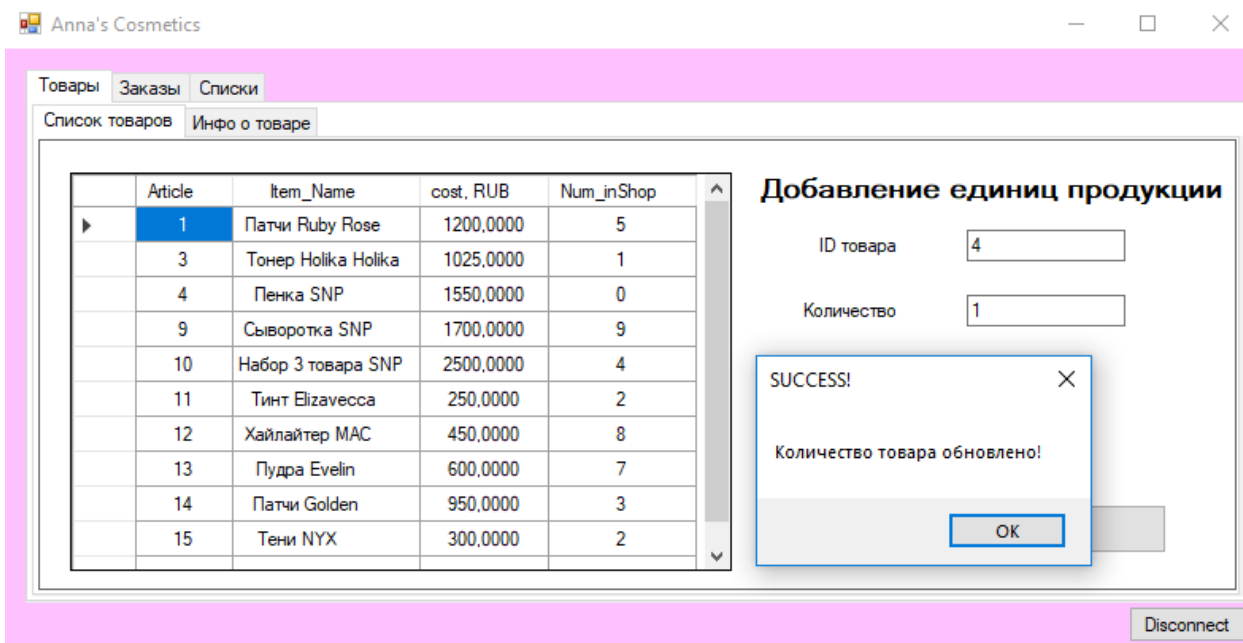


Рис. 13 Добавление недостающих позиций в магазин

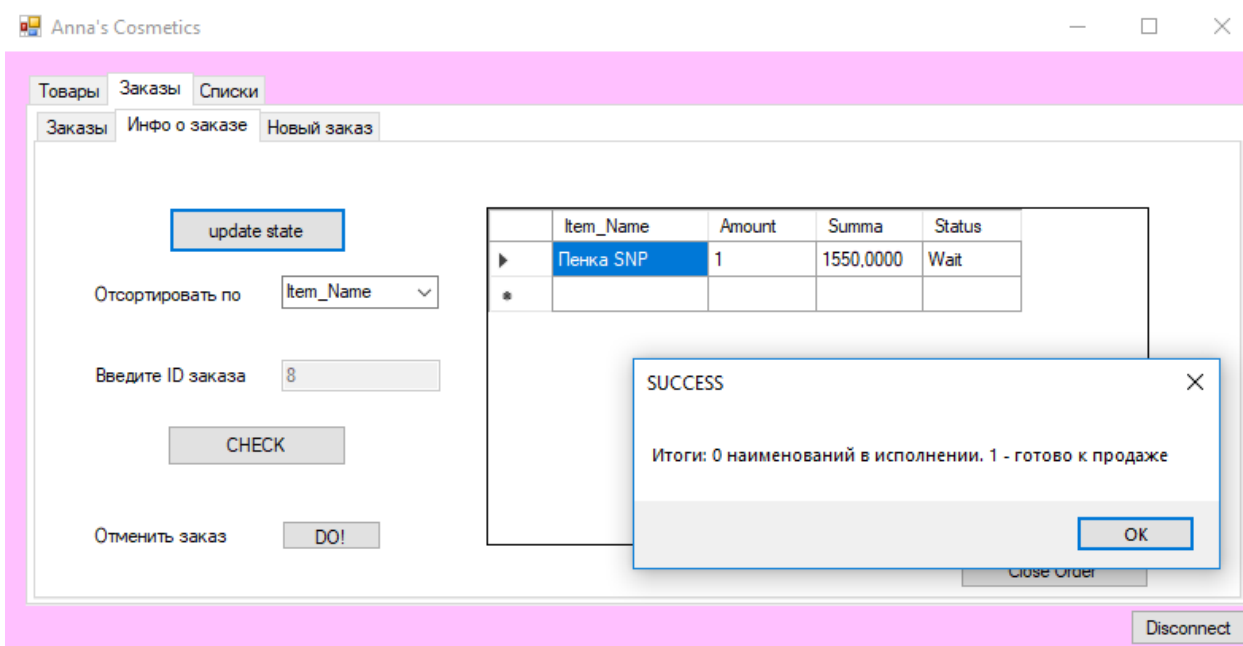


Рис. 14 Переход заказа в статус «Ready»

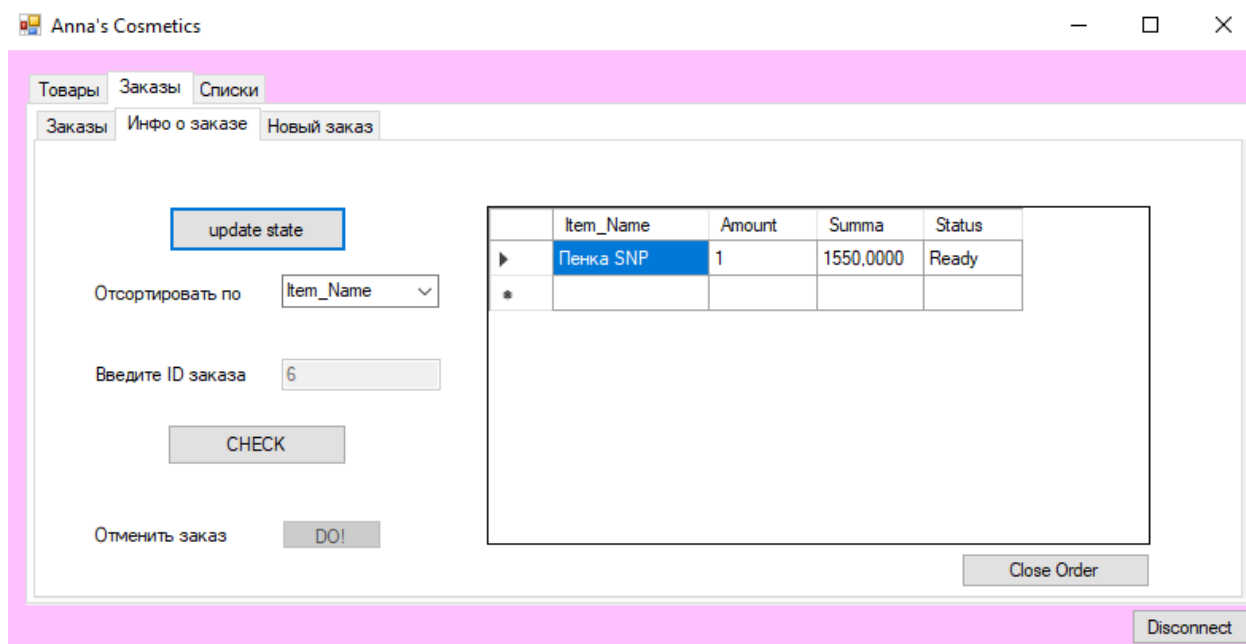


Рис. 15 Заказ готов к продаже

Продажа заказа

Наконец, когда весь заказ готов (сразу же или в момент добавления продавцом товара), заказ отдается клиенту и ставится статус “Sold”. (Рис.16-17)

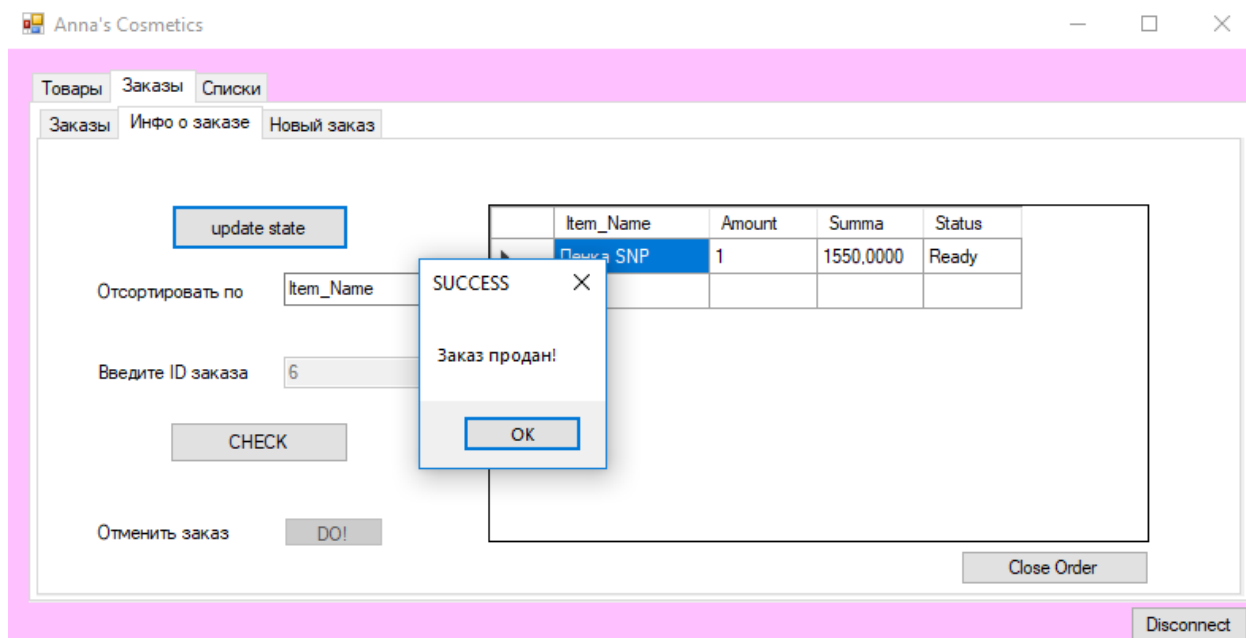


Рис. 16 Момент продажи заказа

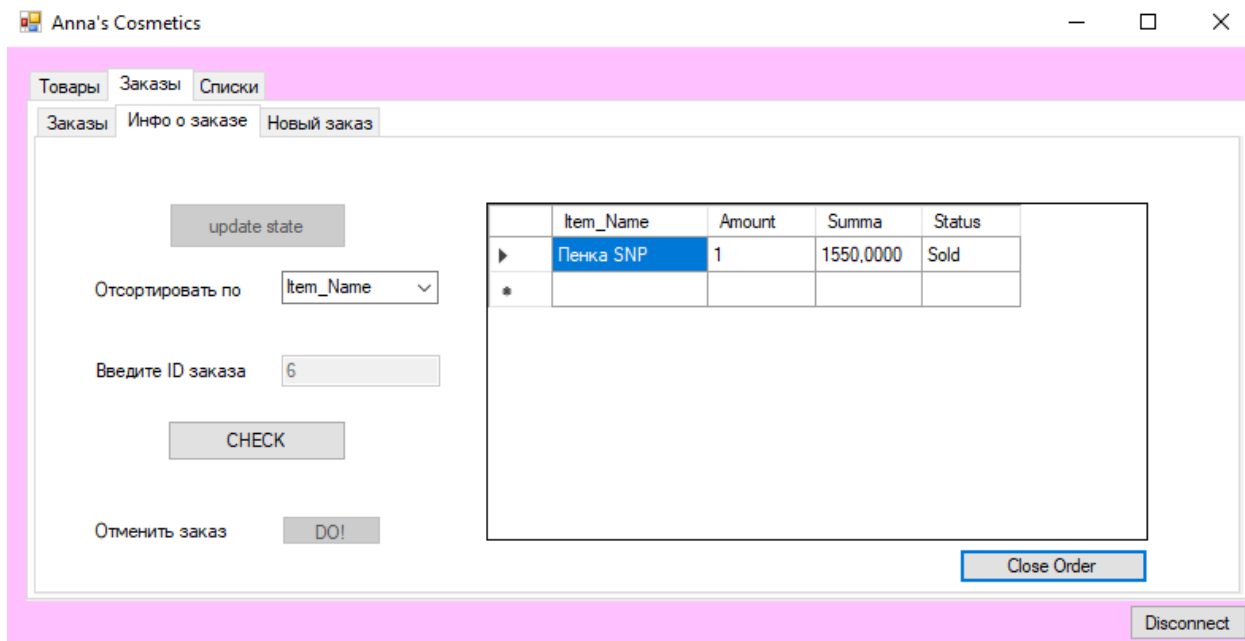


Рис. 17 Присвоение заказу статуса «Sold»

Удаление клиента

Удаление клиента происходит простым вводом его id.

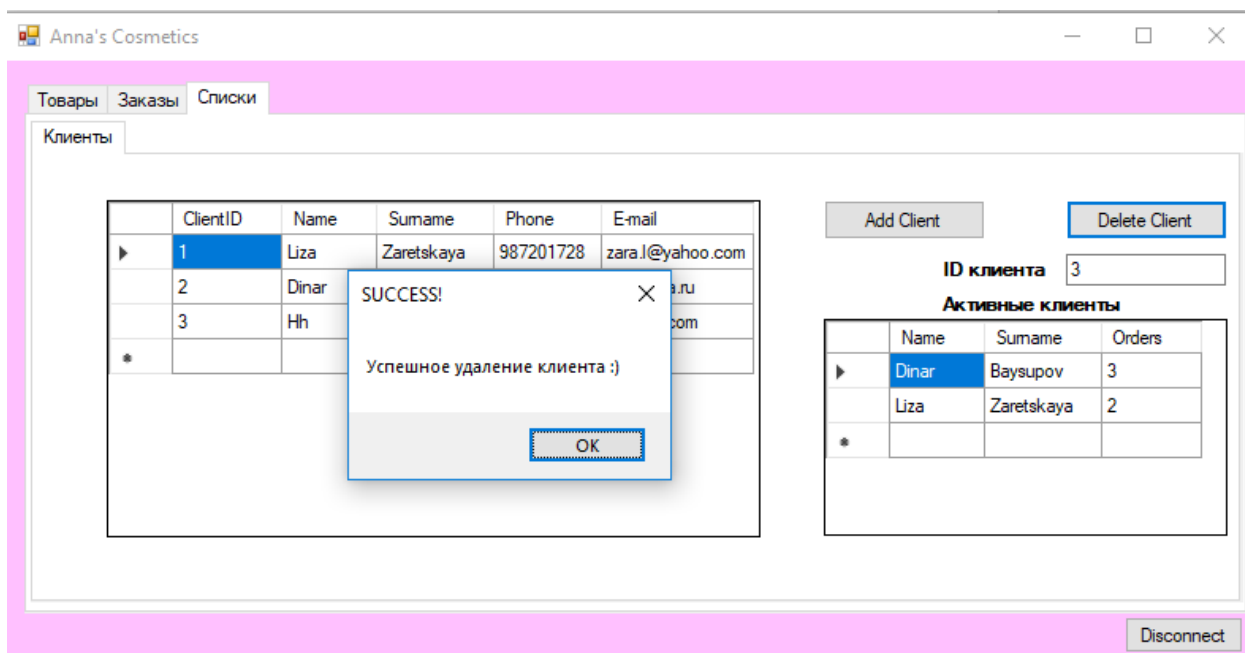


Рис. 18 Удаление клиента из базы данных

Заключение

В ходе работы над проектом, создания и разработки базы данных были изучены принципы построения многопользовательских баз данных, применены навыки использования языка SQL для написания запросов, создания представлений и таблиц, а так же создана графическая оболочка в виде клиентского приложения для взаимодействия с базой. Как непосредственно база данных, так и клиентское приложения являются многопользовательскими, с разделением прав доступа к различным таблицам и элементам базы данных. Также проведено тестирование проекта, после которого можно убедиться, что всё работает корректно.

Список литературы

Грабер, М. (2007). *Введение в SQL* (изд. Лори). Лори.

Дейт, К. Д. (2006). *Введение в системы баз данных*. Спб: Питер.

Марков, А. С., & Лисовский, К. Ю. (2008). *Базы данных: введение в теорию и методологию* (изд. Финансы и статистика). Финансы и статистика.

Стиллмен Э., Грин Дж. (2014). *Изучаем с#*. 3-е изд. Спб: Питер.