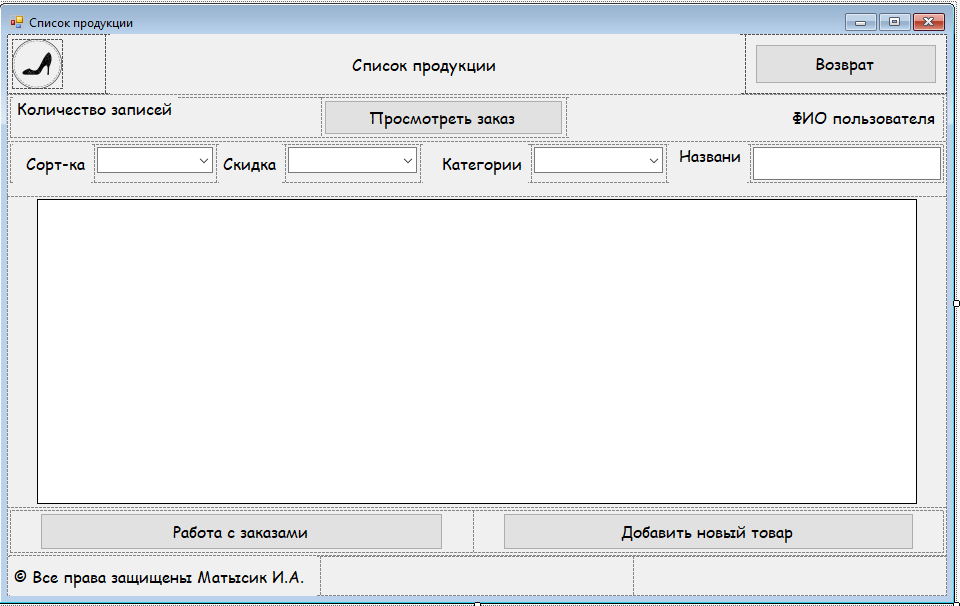
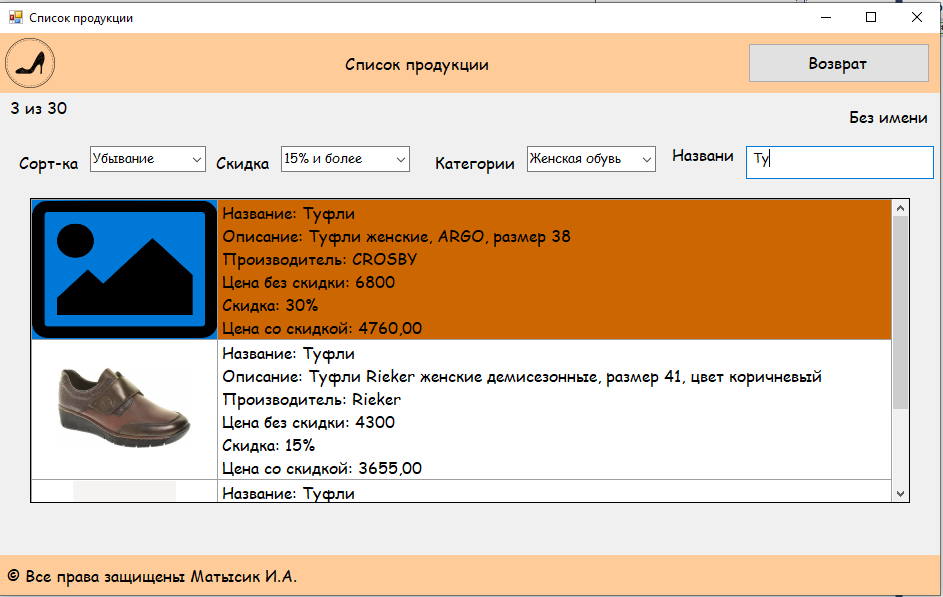
**Реализация просмотра товаров с применением платформы Entity Framework**



Форма в режиме Конструктора



Форма в процессе просмотра каталога товаров с элементами фильтрации

**Фрагменты кода**

namespace Shoes.View

{

public partial class ProductList : Form

{

//Глобальные величины

string pathExe = Application.StartupPath; //Путь к exe без имени файла

int filterDiscount, filterCategory; //Фильтр по скидке и категории

string sort; //Направление сортировки

int countAll, countFilter; //Количество всех записей и после фильтрации

string search; //Строка поиска

//Массив диапазонов скидок

double[,] arrayDiscount = new double[,] { { 0, 100 }, { 0, 10 }, { 11, 14 }, { 15, 100 } };

/// Метод отображения результатов запросов под условия пользователя

private void ShowInfo()

{

countAll = Helper.DB.Product.Count(); //Общее количество товаров в таблице товаров БД

//\*\*\*\*\*Создание сложного запроса из отдельных элементов

var products = Helper.DB.Product.ToList(); //Вся информация о товарах

//Фильтрация по скидке

products = products.Where(x => (x.ProductMaxSale >= arrayDiscount[filterDiscount, 0] &&

x.ProductMaxSale <= arrayDiscount[filterDiscount, 1])).ToList();

//Фильтрация по категории

if (filterCategory != 0) //Не «Все категории»

{

products = products.Where(x => x.CategoryId == filterCategory).ToList();

}

//Поиск по названию

if (!String.IsNullOrEmpty(search))

{

products = products.Where(x => x.ProductName.Contains(search)).ToList();

}

//Сортировка

if (sort == "ASC")

{

products = products.OrderBy(x => x.ProductCost).ToList();

}

else

{

products = products.OrderByDescending(x => x.ProductCost).ToList();

}

//Количество оставшихся записей после выполнения фильтрации

countFilter = products.Count();

//\*\*\*Отображение результатов запроса в настроенной сетке

//Начальные настройки сетки

dataGridViewProducts.Rows.Clear(); //Очистить строки

int row = 0; //Номер текущей строки

//\*\*\*\*\*\*Перебор по очереди всех товаров из списка товаров

foreach (var item in products)

{

dataGridViewProducts.Rows.Add(); //Добавить новую пустую строку в сетку

//В первую колонку занести артикул товара - невидимый

dataGridViewProducts.Rows[row].Cells[0].Value = products[row].ProductArticle;

//Работа с фото

Bitmap bit;

if (String.IsNullOrEmpty(products[row].ProductPhoto)) /Проверка пустоты поля с фото

{

bit = Properties.Resources.picture; //Картинку из ресурсов

}

else //Поле с картинкой не пустое

{ //Полный путь к картинке

string s = pathExe + @"\Товар\_import\" + products[row].ProductPhoto;

if (!File.Exists(s)) //Проверка существования файла с фото

{

bit = Properties.Resources.picture; //Не нашли файл - загрузка из ресурса

}

else

{

bit = new Bitmap(s); //Загрузка файла

}

}

dataGridViewProducts.Rows[row].Cells[1].Value = bit; //Загрузить картинку

//Работа с описанием товара - строка из нескольких полей

string dics = "Название: " + products[row].ProductName + Environment.NewLine;

dics += "Описание: " + products[row].ProductDescription + Environment.NewLine;

dics += "Производитель: " + products[row].Manufacturer.ManufacturerName +

Environment.NewLine;

dics += "Цена без скидки: " + products[row].ProductCost + Environment.NewLine;

//Расчет стоимости товара со скидкой

if (products[row].ProductMaxSale > 0) //Есть скидка

{

double discount = (double)products[row].ProductCost \*

products[row].ProductMaxSale / 100.0;

double costWithDiscount = (double)products[row].ProductCost - discount;

dics += "Скидка: " + products[row].ProductMaxSale + "%" + Environment.NewLine;

dics += "Цена со скидкой: " + costWithDiscount.ToString("F2");

}

dataGridViewProducts.Rows[row].Cells[2].Value = dics; //Отобразить поле описания

//\*\*\*\*\*\*\*\*\*\*\*Подготовить таблицу для отображения

dataGridViewProducts.Rows[row].Height = 140; //Высота строк таблицы

//Задание цвета для строк при заданных условиях

if (products[row].ProductMaxSale > 15) //Если скидка больше 15%

{

dataGridViewProducts.Rows[row].DefaultCellStyle.BackColor = Color.FromArgb(204, 102, 0);

}

else

{

dataGridViewProducts.Rows[row].DefaultCellStyle.BackColor = Color.FromArgb(255, 255, 255);

}

row++;

}

labelCount.Text = countFilter + " из " + countAll; //Вывод количества

}

/// Начальные настройки

private void ProductList\_Load(object sender, EventArgs e)

{

tableLayoutPanelTop.BackColor = Color.FromArgb(255, 204, 153);

tableLayoutPanelBottom.BackColor = Color.FromArgb(255, 204, 153);

dataGridViewProducts.BackgroundColor = Color.FromArgb(255, 255, 255);

//Формирование списка категорий из таблицы БД + строка "Все категории"

var category = Helper.DB.Category.Select(x => x.CategoryName).ToList(); //Все категории

category.Insert(0, "Все категории"); //Добавить первой «Все категории»

comboBoxCategory.DataSource = category; //Занести построенный список

//Настроить видимость первой строки списка

comboBoxCategory.Text = comboBoxCategory.Items[0].ToString();

comboBoxDiscount.Text = comboBoxDiscount.Items[0].ToString();

comboBoxSort.Text = comboBoxSort.Items[0].ToString();

labelFIO.Text = Helper.UserFullName; //Отобразить ФИО клиента

//Начальные данные для фильтрации

filterDiscount = 0;

filterCategory = 0;

sort = "ASC";

ShowInfo(); //Отобразить список товаров

}

/// Выбор фильтрации по скидке

private void comboBoxDiscount\_SelectedIndexChanged(object sender, EventArgs e)

{

filterDiscount = comboBoxDiscount.SelectedIndex;

ShowInfo();

}

/// Изменение направления сортировки

private void comboBoxSort\_SelectedIndexChanged(object sender, EventArgs e)

{

sort = "ASC";

if (comboBoxSort.SelectedIndex == 1)

sort = "DESC";

ShowInfo();

}

/// Фильтр по категории

private void comboBoxCategory\_SelectedIndexChanged(object sender, EventArgs e)

{

filterCategory = comboBoxCategory.SelectedIndex;

ShowInfo();

}

/// Ввод строки для поиска

private void textBoxSearch\_TextChanged(object sender, EventArgs e)

{

search = textBoxSearch.Text;

ShowInfo();

}

}

}