**Фрагменты кода для создания заказа технологией Entity Framework**

using System.IO;

namespace Shoes

{

public partial class ViewOrder : Form

{

List<ProductOrder> orderList; //Переданный через конструктор заказ

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

int uniqueCode; //Уникальный код заказа

DateTime dateNow; //Дата заказа - текущая

double totalSumma; //Общая сумма заказа

double totalSale; //Общая скидка заказа

DateTime dateDelivery; //Дата доставки товара

int orderId; //Номер заказа

/// Параметризованный конструктор

public ViewOrder(List<ProductOrder> order) //Переданный заказ с товарами

{

InitializeComponent();

this.orderList = order; //Пересохраняем в глобальной переменной

}

/// Метод отображения информации о товарах заказа

private void ShowOrder()

{

dataGridViewOrder.RowCount = orderList.Count; //Количество позиций в заказе

int i = 0;

int countProducts = 0;

totalSumma = 0; //Общая сумма заказа

totalSale = 0; //Общая скидка заказа

double cost, sale;

foreach (var item in orderList) //По всем товарам заказа

{

//Получить артикль из списка заказа

dataGridViewOrder.Rows[i].Cells[0].Value = item.productArticleNumber;

//Получить всю информацию о товаре по его артиклю – ключевому полю

**Entities.Product product = Helper.DB.Product.Find(item.productArticleNumber);**

//Вывести и провести расчеты для этого товара

dataGridViewOrder.Rows[i].Cells[1].Value = product.ProductName;

dataGridViewOrder.Rows[i].Cells[2].Value = item.productCount;

//Общие расчеты за заказ

countProducts += item.productCount;

cost = Convert.ToDouble(product.ProductCost) \* item.productCount;

sale = Convert.ToDouble(product.ProductSale) / 100.00 \* cost;

totalSumma += cost;

totalSale += sale;

dataGridViewOrder.Rows[i].Height = 30;

i++;

}

textBoxDescOrder.Text = "Позиций в заказе: " + i + Environment.NewLine;

textBoxDescOrder.Text += "Всего товаров: " + countProducts + Environment.NewLine;

textBoxDescOrder.Text += "Общая сумма за весь товар: " + totalSumma +

Environment.NewLine;

textBoxDescOrder.Text += "Общая сумма скидки: " + totalSale + Environment.NewLine;

textBoxDescOrder.Text += "Общая сумма за весь товар со скидкой: " +

(totalSumma - totalSale);

}

/// Начальные настройки

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

{

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

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

//Настройка списка пунктов выдачи товаров

**comboBoxPickupPoint.DataSource = Helper.DB.Point.ToList();**

comboBoxPickupPoint.DisplayMember = "PointAddress";

comboBoxPickupPoint.ValueMember = "PointId";

//Вывод ФИО для клиента

if (Helper.RoleID == (int)Roles.Клиент)

{

labelFIO.Text = Helper.UserFullName;

}

else

{

labelFIO.Text = "";

}

ShowOrder();

buttonCreateOrder.Enabled = true;

buttonCreateCoupon.Enabled = false;

}

/// Щелчок по ячейкам таблицы

private void dataGridViewOrder\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

//Получить данные о товаре в заказе

int num = e.RowIndex; //Номер строки в заказе- строка в таблице

string productArticle = dataGridViewOrder.Rows[num].Cells[0].Value.ToString(); //Артикль

//Нажали кнопку напротив товара - обнулить количество данного товара в списке

if (e.ColumnIndex == 3) //Кнопка «Удалить»

{

orderList.RemoveAt(num); //Удалить товар из заказа кнопкой

ShowOrder();

}

else

{

//Столбец названий товаров - отобразить информацию о товаре

if (e.ColumnIndex == 1)

{

//Получить информацию о товаре по его артиклю – ключевому полю

**Entities.Product product = Helper.DB.Product.Find(productArticle);**

Bitmap bit;

string photo = product.ProductPhoto;

if (String.IsNullOrEmpty(photo)) //Проверка пустоты поля с фото

{

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

}

else //Поле с картинкой не пустое

{

string s = pathExe + @"\Товар\_import\" + photo; //Полный путь к картинке

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

}

}

pictureBoxPhoto.Image = bit;

//Работа с описанием товара - строка из нескольких строк

string dics = "";

dics += "Название: " + product.ProductName + Environment.NewLine;

dics += "Описание: " + product.ProductDescription + Environment.NewLine;

dics += "Категория: " + product.Category.CategoryName + Environment.NewLine;

dics += "Производитель: " + product.Manufacturer.ManufacturerName +

Environment.NewLine;

dics += "Цена без скидки: " + product.ProductCost.ToString() + Environment.NewLine;

dics += "Скидка: " + product.ProductSale.ToString() + Environment.NewLine;

dics += "Поставщик: " + product.Provider.ProviderName;

textBoxDescProduct.Text = dics;

}

}

}

/// ввод нового значения в ячейку с количеством

private void dataGridViewOrder\_CellEndEdit(object sender, DataGridViewCellEventArgs e)

{

//Новое значение

int res = Convert.ToInt32(dataGridViewOrder.Rows[e.RowIndex].Cells[e.ColumnIndex].Value);

orderList[e.RowIndex].productCount = res; //Занесение в заказ

if (res == 0)

{

orderList.RemoveAt(e.RowIndex); //Удалить товар из заказа при нулевом количестве

}

ShowOrder();

}

/// Фиксация в БД заказа

private void buttonCreateOrder\_Click(object sender, EventArgs e)

{

dateNow = DateTime.Now;

//Просмотр наличия всех товаров в заказе на складе

int daysDelivery = 3; //Количество дней доставки

int countInStock; //Количество каждого товара на складе

foreach (var item in orderList) //По всем товарам заказа

{

//Найти информацию о товаре на складе по артиклю – ключевому полю

**Entities.Product product = Helper.DB.Product.Find(item.productArticleNumber);**

//Кол-во данного товара на складе

countInStock = product.ProductCount;

if (countInStock < 3)

{

daysDelivery = 6; //Количество дней доставки увеличивается

break; //Хотя бы для одного заказа

}

}

//Вычисление даты доставки

dateDelivery = dateNow.AddDays(daysDelivery);

int idPickupPoint = Convert.ToInt32(comboBoxPickupPoint.SelectedValue);

//Формирование уникального кода заказа

Entities.Order order;

do

{

uniqueCode = new Random().Next(100, 1000);

//Поиск уникального кода среди всех заказов

**order = Helper.DB.Order.Where(x => x.UniqueCode == uniqueCode).FirstOrDefault();**

}

while (order!=null); //Пока такой код есть

//Определение ФИО заказчика только для клиента

string userFullName = null;

if (Helper.RoleID == (int)Roles.Клиент)

{

userFullName = Helper.UserFullName;

}

//Добавление в основную таблицу заказов

//Формирование нового объекта - заказа

**Entities.Order newOrder = new Entities.Order();**

newOrder.OrderDate = dateNow;

newOrder.OrderDeliveryDate = dateDelivery;

newOrder.PointId = idPickupPoint;

newOrder.UserFullName = userFullName;

newOrder.UniqueCode = uniqueCode;

newOrder.StatusId = 2; //Статус «Новый»

try

{

**Helper.DB.Order.Add(newOrder); //Добавление в список**

**Helper.DB.SaveChanges(); //Обновление БД по списку**

//Получение номера последнего добавленного заказа

**orderId = Helper.DB.Order.ToList().Last().OrderId;**

//Добавление товаров созданного заказа в таблицу OrderProduct

foreach (var item in orderList) //По всем товарам заказа

{

//Создание нового объекта - товар в заказе

**Entities.OrderProduct orderProduct = new Entities.OrderProduct();**

orderProduct.OrderId = orderId;

orderProduct.ProductArticle = item.productArticleNumber;

orderProduct.ProductCount = item.productCount;

try

{

**Helper.DB.OrderProduct.Add(orderProduct);** //Добавление в список

**Helper.DB.SaveChanges();** //Обновление БД по списку

}

catch

{

MessageBox.Show("Не удалось оформить заказ");

this.Close();

return;

}

}

}

catch

{

MessageBox.Show("Не удалось оформить заказ");

this.Close();

return;

}

MessageBox.Show("Заказ удачно оформился" + Environment.NewLine +

"Код Вашего заказа: " + uniqueCode + Environment.NewLine +

"Дата получения заказа: " + dateDelivery.ToLongDateString());

buttonCreateOrder.Enabled = false;

buttonCreateCoupon.Enabled = true;

}