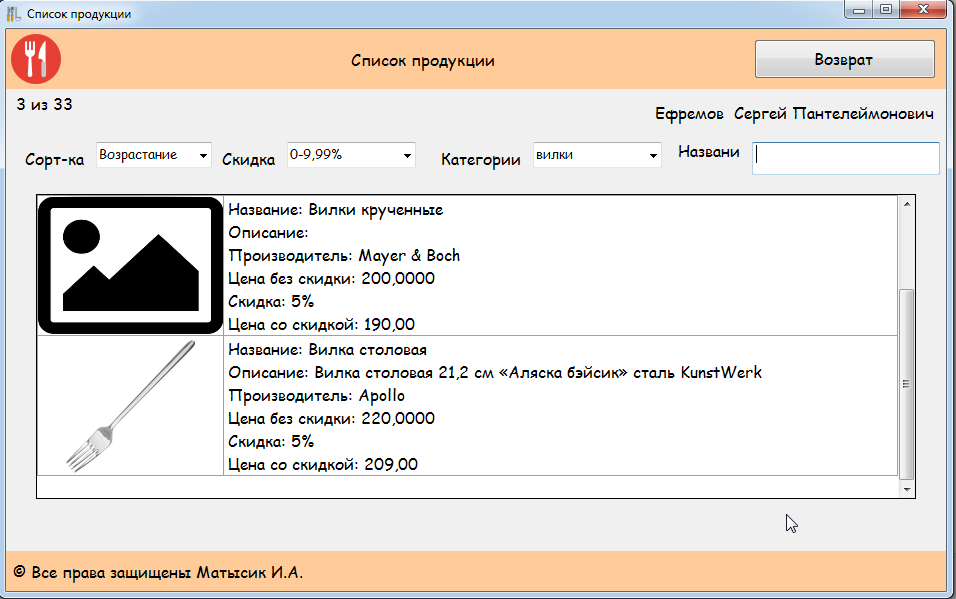
**Фрагмент кода для формы просмотра товаров с применением фильтрации**



Момент работы приложения при выборе скидки и категории товаров

namespace WinFormCook

{

/// Форма просмотра товаров

public partial class ListProduct : Form

{

//Глобальные величины

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

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

string sort; //Направление сортировки

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

string search; //Строка поиска

//Массив скидок

string[,] arrayDiscount = new string[,] { {"0","100"},{"0","9.99" },{"10","14.99"}, {"15","100"}};

/// Метод отображения результатов запросов под условия пользователя

private void ShowInfo()

{

//Общее количество товаров в таблице товаров

string sqlCount = "SELECT COUNT (\*) FROM [Product]";

using (SqlCommand command = Helper.Connection.CreateCommand())

{

command.CommandText = sqlCount;

countAll = (int)command.ExecuteScalar();

}

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

//Все поля таблицы товаров и производителей

string sqlProducts = "SELECT \* FROM [Product], [Manufacturer]"+

" WHERE [Product].ProductManufacturerID = [Manufacturer].ManufacturerID";

//Фильтрация по скидке

string filterDiscountString = " AND [Product].ProductDiscountMax BETWEEN " +

arrayDiscount[filterDiscount, 0] + " AND " + arrayDiscount[filterDiscount, 1];

sqlProducts += filterDiscountString;

//Фильтрация по категории

if (filterCategory != 0)

{

string filterCategoryString = " AND [Product].ProductCategoryID=" + filterCategory;

sqlProducts += filterCategoryString;

}

//Поиск по названию

string searchString = " AND [Product].ProductName LIKE '%" + search + "%'";

sqlProducts += searchString;

//Сортировка

string sortString = " ORDER BY [Product].ProductCost " + sort;

sqlProducts += sortString;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Выполнение сформированного запроса и отображение его результатов

using (SqlCommand command = Helper.Connection.CreateCommand())

{

command.CommandText = sqlProducts;

countFilter = 0;

dataGridViewProducts.Rows.Clear();

SqlDataReader reader = command.ExecuteReader();

while(reader.Read())

{

dataGridViewProducts.Rows.Add();

dataGridViewProducts.Rows[countFilter].Cells[0].Value =

reader["ProductArticleNumber"].ToString();

//Работа с фото

Bitmap bit;

//проверка пустоты поля с фото

if (String.IsNullOrEmpty(reader["ProductPhoto"].ToString()))

{

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

}

else //Поле с картинкой не пустое

{

//Полный путь к картинке

string s = pathExe + @"\Photos\" + reader["ProductPhoto"].ToString();

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[countFilter].Cells[1].Value = bit;

//Работа с описанием товара - строка из нескольких строк

string dics = "";

dics += "Название: " + reader["ProductName"].ToString() + Environment.NewLine;

dics += "Описание: " + reader["ProductDescription"].ToString() +

Environment.NewLine;

dics += "Производитель: " + reader["ManufacturerName"].ToString() +

Environment.NewLine;

dics += "Цена без скидки: " + reader["ProductCost"].ToString() +

Environment.NewLine;

//Расчет стоимости товара со скидкой

if (reader.GetByte(reader.GetOrdinal("ProductDiscountMax")) > 0)

{

double discount =(double)( reader.GetDecimal(reader.GetOrdinal("ProductCost")))

\* reader.GetByte(reader.GetOrdinal("ProductDiscountMax")) / 100.0;

double costWithDiscount =

(double)(reader.GetDecimal(reader.GetOrdinal("ProductCost"))) - discount;

dics += "Скидка: " + reader["ProductDiscountMax"].ToString() + "%" +

Environment.NewLine;

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

}

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

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

dataGridViewProducts.Rows[countFilter].Height = 140;

//Задание цвета для строк при заданных условиях

if (reader.GetByte(reader.GetOrdinal("ProductDiscountMax")) > 15)

{

dataGridViewProducts.Rows[countFilter].DefaultCellStyle.BackColor =

Color.FromArgb(204, 102, 0);

}

else

{

dataGridViewProducts.Rows[countFilter].DefaultCellStyle.BackColor =

Color.FromArgb(255, 255, 255);

}

countFilter++; //Число записей, попавших под фильтр

} //Завершение цикла while

reader.Close();

} //Закрытие using

//Вывод количества отфильтрованных записей и общего количества товаров

labelCount.Text = countFilter + " из " + countAll;

}

/// Начальные настройки

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

{

//Фрагмент для фильтрации и сортировки

filterDiscount = 0;

filterCategory = 0;

sort = "ASC";

ShowInfo();

}

/// Изменение направления сортировки

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

{

sort = "ASC";

if (comboBoxSort.SelectedIndex==1)

sort = "DESC";

ShowInfo();

}

/// Выбор фильтрации по скидке

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

{

filterDiscount = comboBoxDiscount.SelectedIndex;

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();

}

}

}