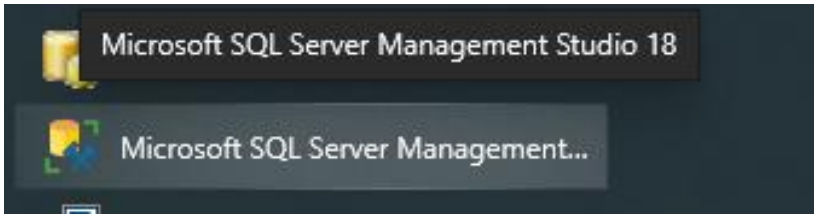
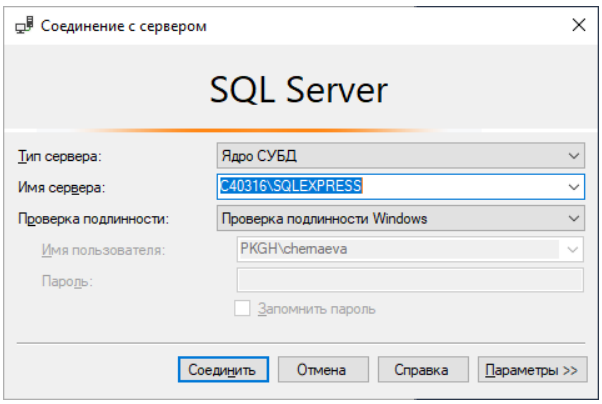
Школа Great Britain

При наличии подготовленных данных хранящих в скрипте, их можно импортировать в базу данных. Для этого выполните следующие действия:

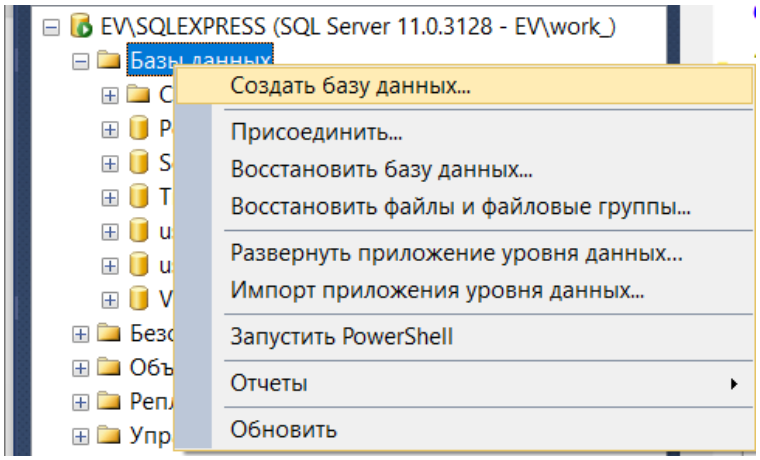
1. Запустите Microsoft SQL Server Management Studio

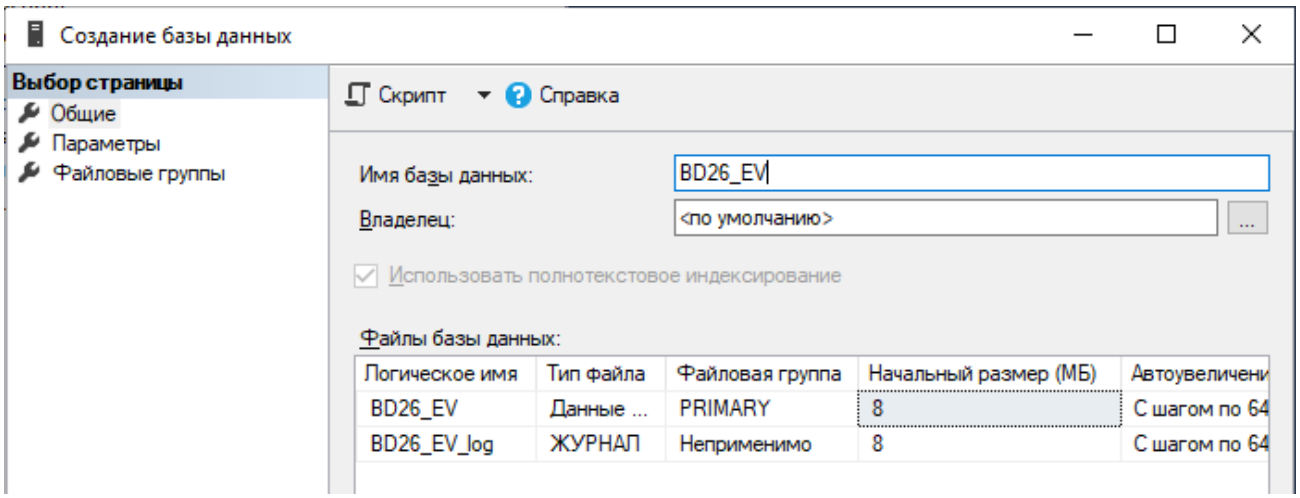


В появившемся окне подтвердите соединение с локальным сервером (Нажмите **Соединить**). Обратите внимание на имя сервера: оно включает имя компьютера и название сервера

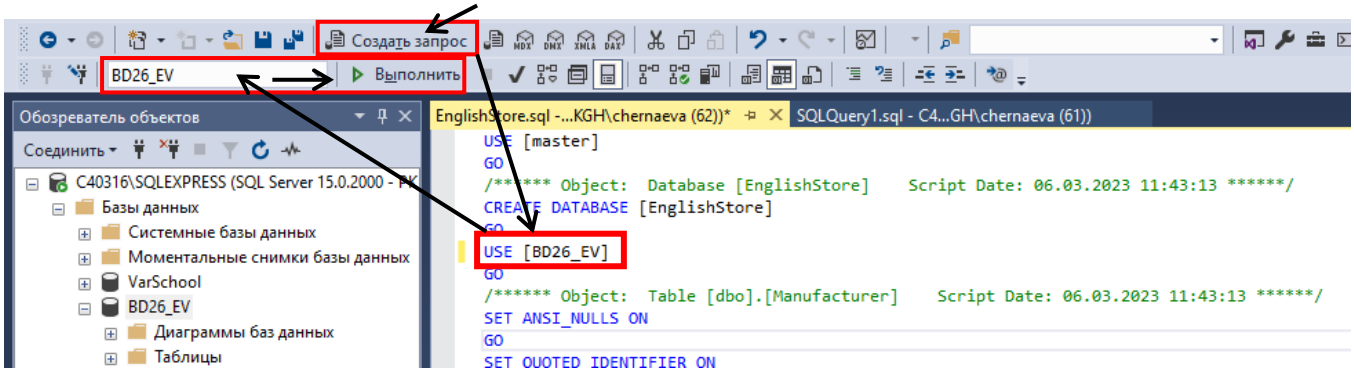


2. Для импорта данных из скрипта необходимо создать БД: ПКМ по **Базы данных – Создать базу данных,** задать имя БД – **BD_Familiy** (фамилию указать свою)



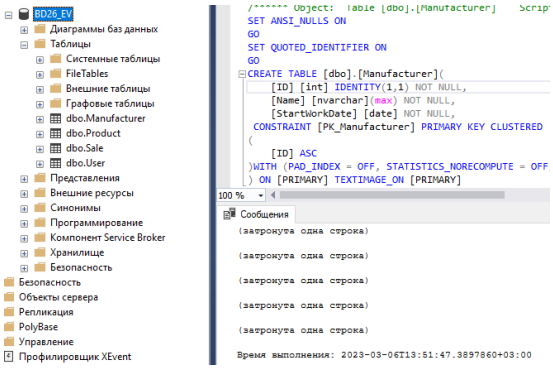


Далее нажать **Создать запрос** – перетащить файл со скриптом в созданный новый запрос, во второй строке **USE** заменить название БД на ваше имя файла для БД



Затем в списке БД выбрать название своей БД И нажать кнопку **!Выполнить**

Получим статистику выполнения запроса и при успешном выполнении таблицы нашей БД с данными в некоторых таблицах



Примечание: *Для просмотра содержимого таблиц: выделить имя нужной таблицы – ПКМ – Изменить первые 200 строк. Для просмотра структуры таблицы (имена полей и их типы) достаточно просто развернуть таблицу, нажав + радом с ее названием*

Создание приложения

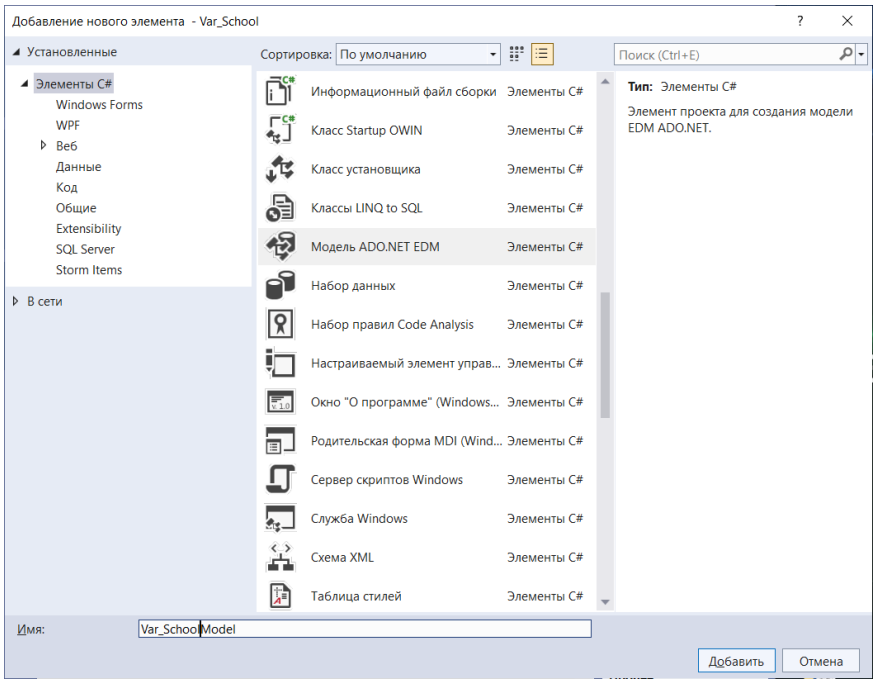
1. Создайте проект **WinForm** с именем *EngSchool_ Familiy*
2. Добавьте к проекту папки (**ПКМ** по имени проекта – **Добавить – Создать папку…**) **Models** (для классов описания таблиц БД), **Forms** (для всех форм)

Создание модели БД (Entity Framework)

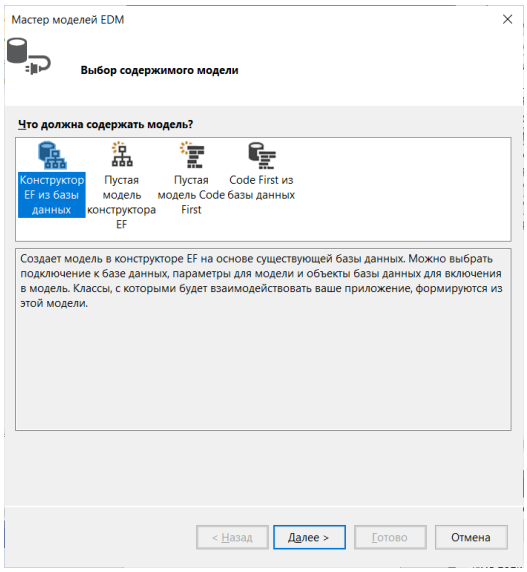
Предварительно у вас должна быть создана БД

1. Выделите папку **Models**, щелкните правой кнопкой мыши по папке, выберите **Добавить\Создать Элемент**.

2. Слева выберите **Данные\ Модель ADO.NET EDM**. Задайте имя модели, например, *EngSchool_ FamiliyModel* и нажмите **Добавить**.



3. Выберите **Конструктор EF** из базы данных и нажмите **Далее**.



4. Нажмите **Создать соединение… (подключение)**

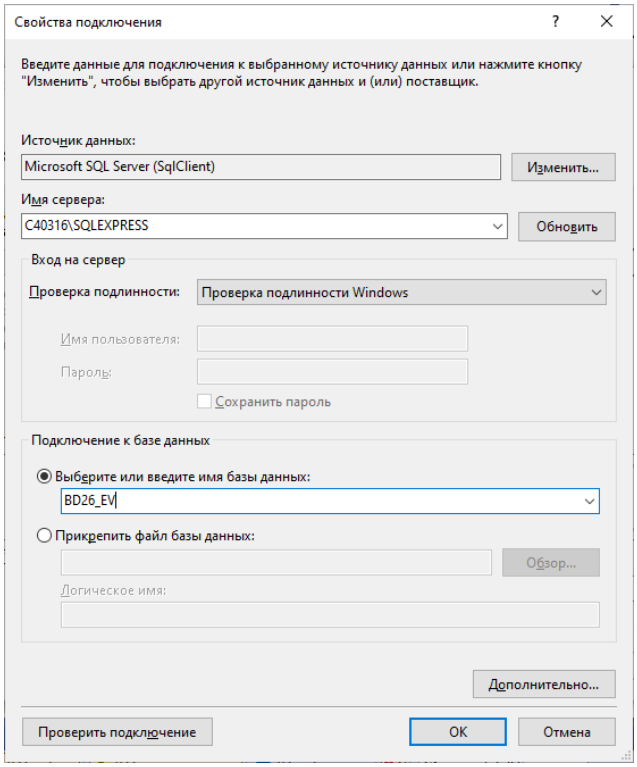
5. Локальное подключение

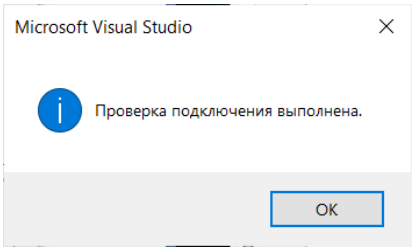
В качестве источника данных укажите **Microsoft SQL Server(SqlClient),**

имя сервера (можно выбрать из выпадающего списка после его нахождения), например, **.\SQLEXPRESS,**

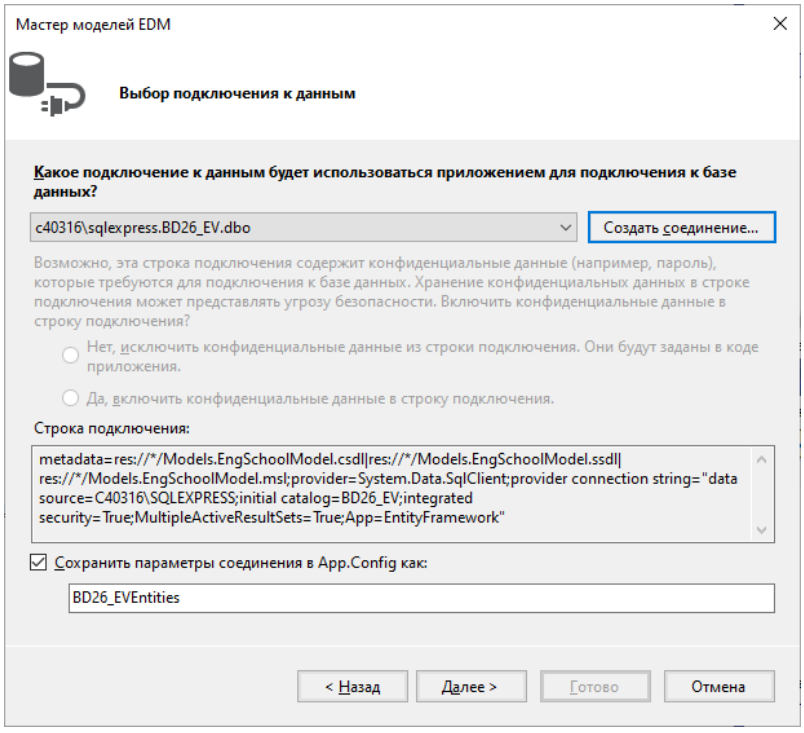
база данных: имя вашей бд, например *EngSchool_ Familiy*.

Нажмите **ОК**.



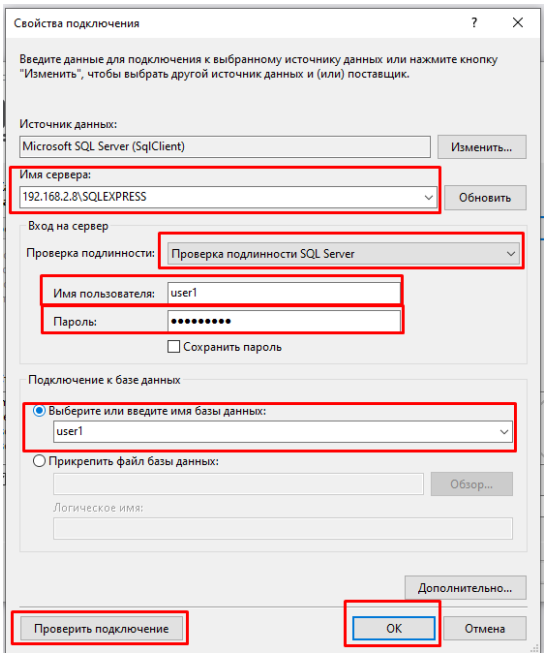


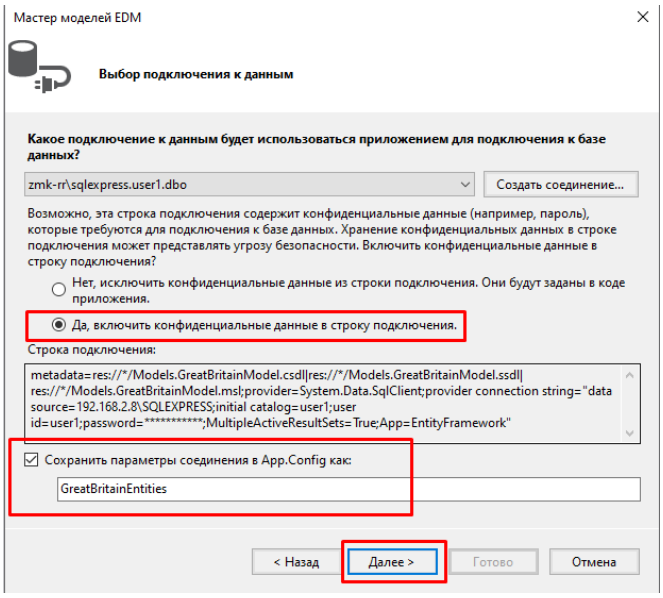
6. Проверьте подключение (обратите внимание на строку подключения, для дальнейшей работы ее можно скопировать в текстовый файл) и настройки. Поставьте флажок возле пункта **Сохранить параметры соединения** в файл **App.config**.



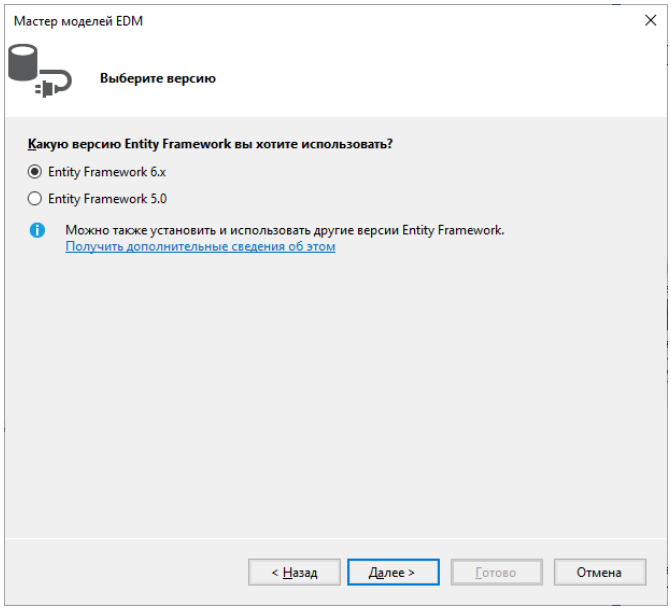
Примечание: *При подключении к удаленному ПК:*

- ✓ *в качестве источника данных укажите Microsoft SQL Server(SqlClient),*
- ✓ *имя сервера: IP-адрес сервера\SQLEXPRESS,*
- ✓ *Проверка подлинности SQL Server*
- ✓ *Задайте ваши имя пользователя и пароль.*
- ✓ *Выберите вашу БД*

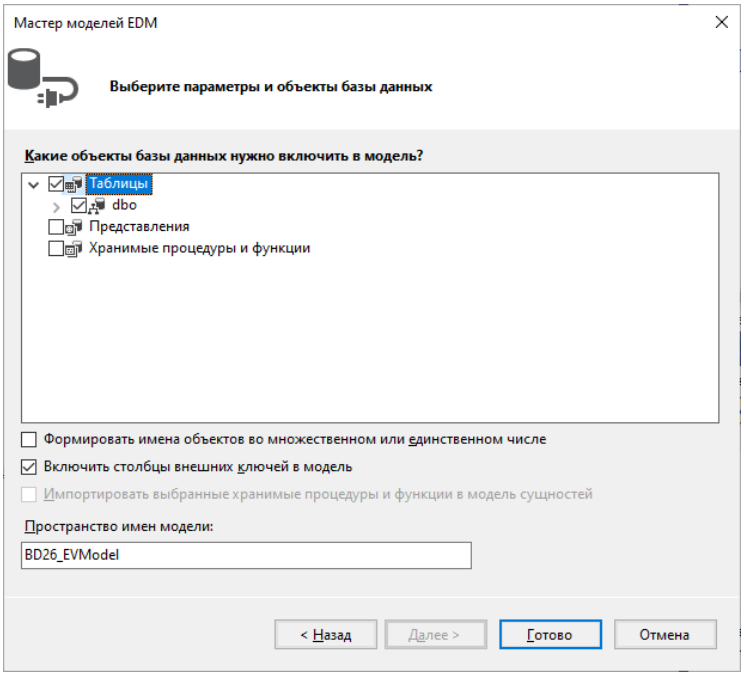




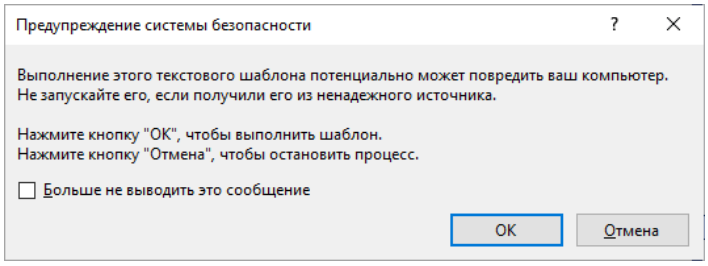
7. Укажите версию **Entity Framework** и нажмите **Далее**.

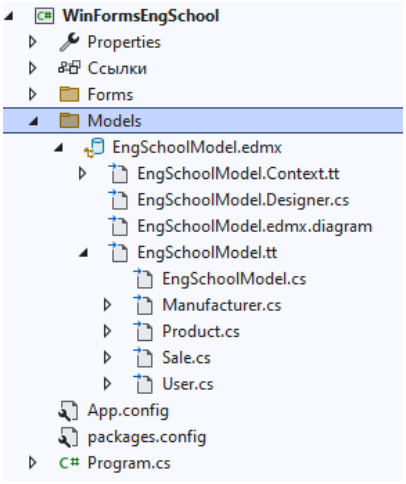


8. Выберите объекты – **Таблицы**, поставьте везде флажки и переименуйте Пространство имен модели в *EngSchool_ FamiliyModel* и нажмите **Готово**.



Подтверждаем выполнение шаблона

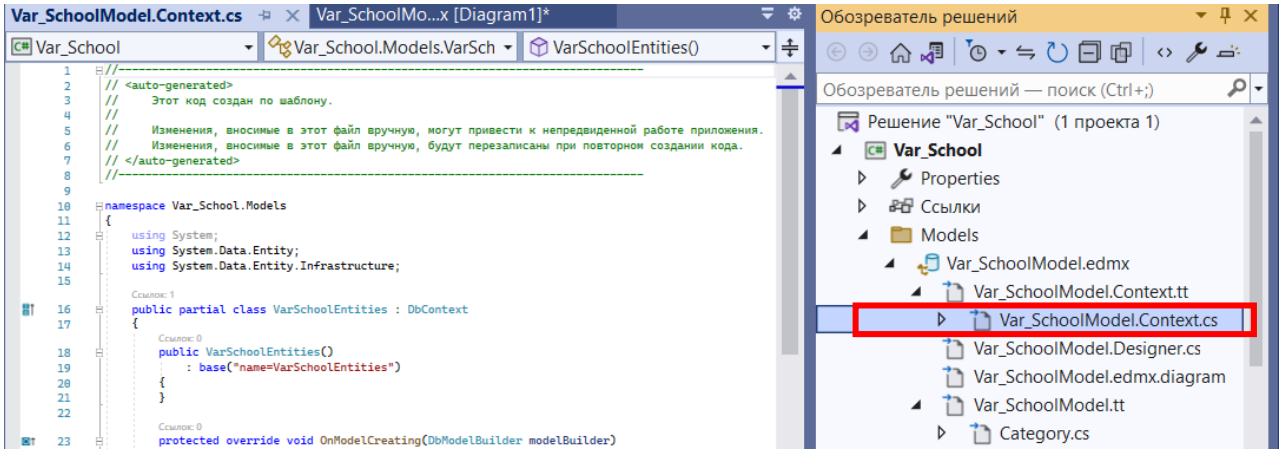




9. Дождитесь завершения импорта базы. В папке **Models** появится модель данной БД, в которой каждая таблица представляется в виде **Класса** и откроется окно схемы данных, которое можно закрыть.

Добавление контекста подключения

1. Откройте файл **EngSchool_ FamiliyModel.Context.cs** из папки **Models**.



Будем использовать паттерн **Singleton** (Одиночка). **Синглтон** позволяет создать объект только при его необходимости. Если объект не нужен, то он не будет создан. В этом отличие синглтона от глобальных переменных.

Добавьте внутрь класса **EngSchool_ Familiy.Entities : DbContext**

следующий код

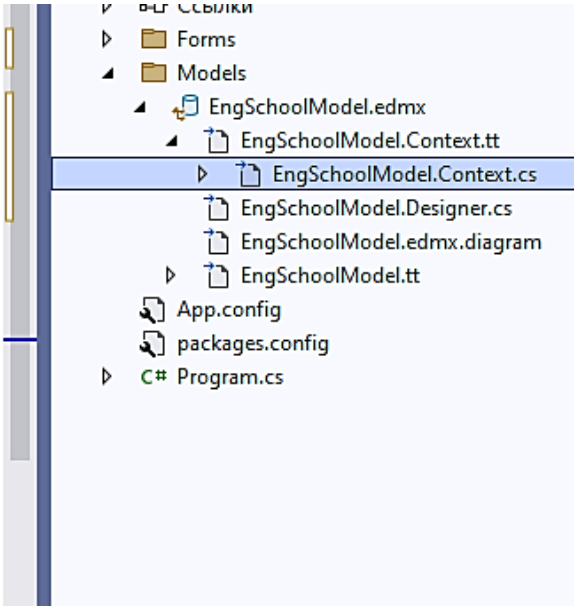
```
namespace WinFormsEngSchool.Models
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Infrastructure;

    public partial class BD26_EVEntities : DbContext
    {
        public static BD26_EVEntities _context;

        public BD26_EVEntities()
            : base("name=BD26_EVEntities")
        {
        }

        public static BD26_EVEntities GetContext()
        {
            if (_context== null)
            {
                _context= new BD26_EVEntities();
            }
            return _context;
        }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            throw new UnintentionalCodeFirstException();
        }
```



***Примечание**. Так как модель привязана к базе данных, то иногда может модель обновляться и добавленные строчки в этот файл могут исчезать. Сохраните резервную копию кода*

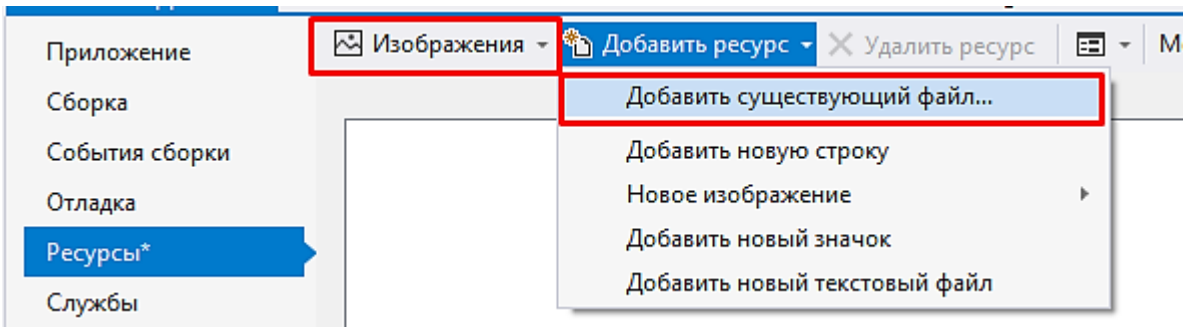
Проектирование каркаса приложения.

1. Добавьте ресурсы: иконку и изображение (предварительно скопируйте файлы на свой ПК).

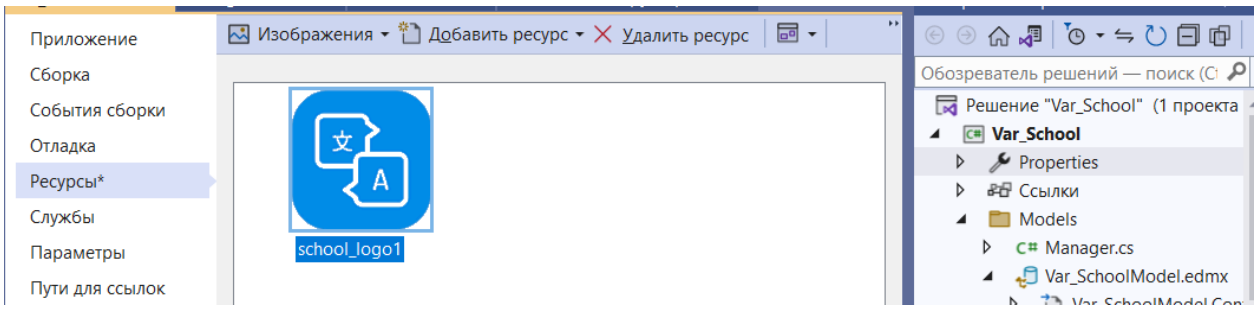
Дважды щелкните левой кнопкой мыши по пункту **Properties**.

2. Выберите вкладку **Ресурсы**

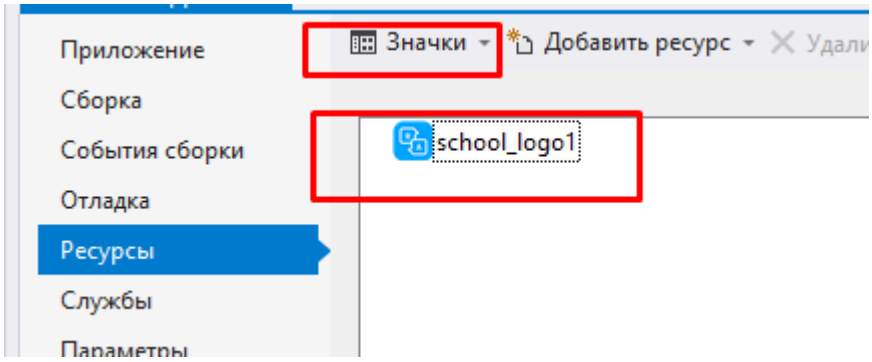
3. Выберите Изображение. **Добавить ресурс\Добавить существующий файл**.



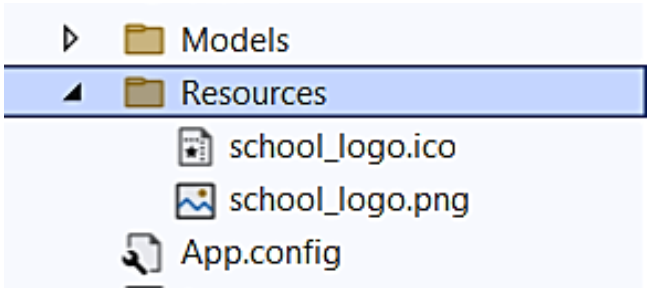
4. В папке **Общие ресурсы** первой сессии выберите нужный файл и нажмите **Открыть**.



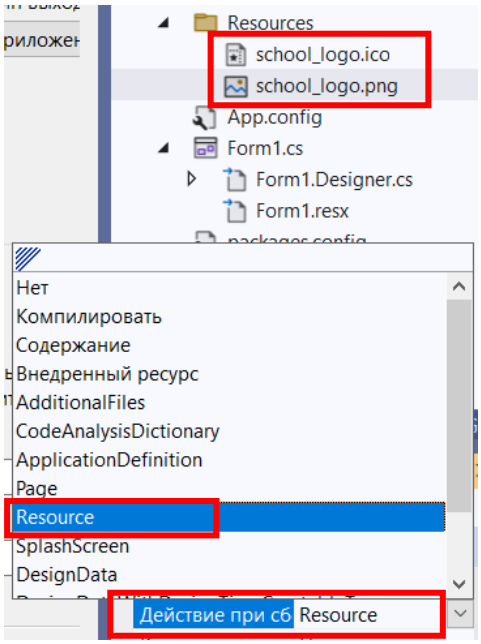
5. Аналогичным образом добавьте иконку



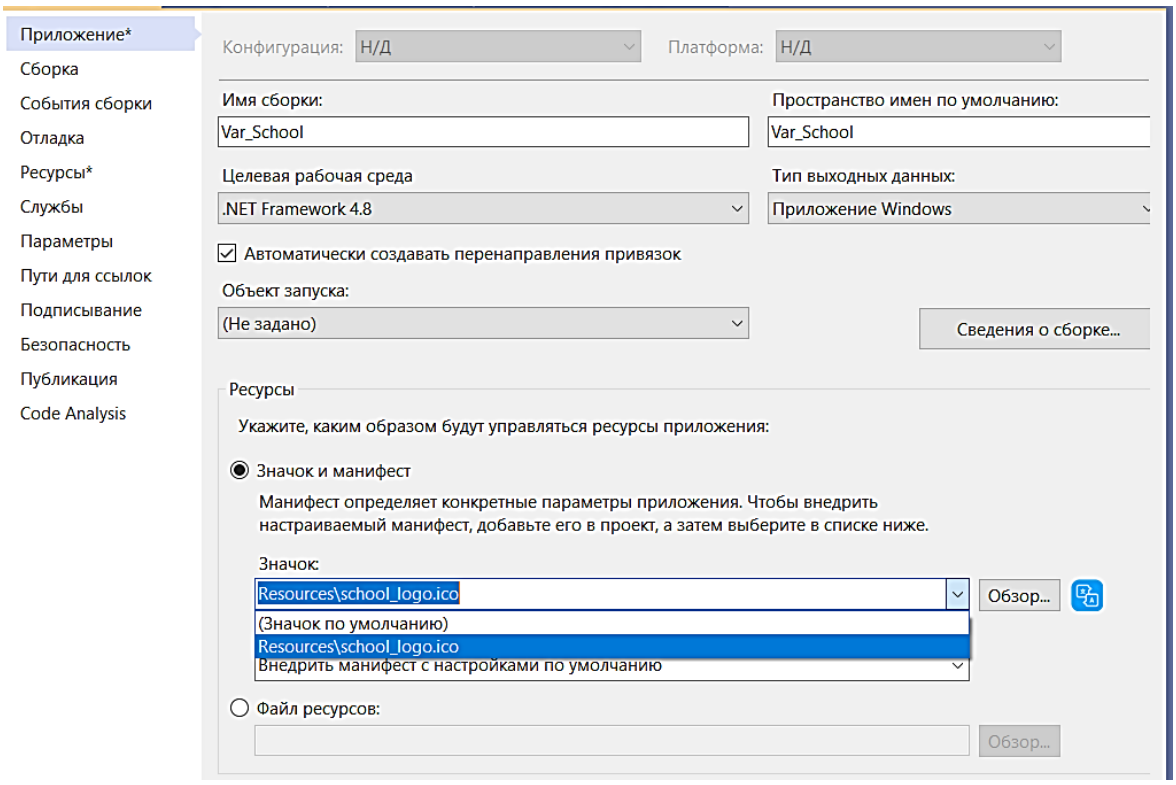
6. В обозревателе решений в папке **Resources** появятся данные файлы



7. В обозревателе решений выделите оба файла и измените свойство **Действие при сборке** на **Resource**



8. В Окне **Properties** выделите пункт **Приложение** и в качестве значка задайте иконку



Форма авторизации

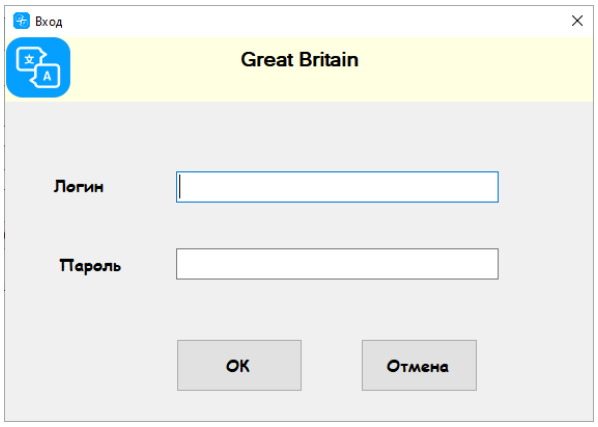
1. Переместите первую форму приложения *Form1* в папку **Forms** и переименуйте ее, например, ***AdminWindow***.

2. При необходимости установите ее запуск при первом запуске приложения в файле **program.cs** (измените в строке *Application.Run(new Form1 ())*;название формы на ваше название, например **AdminWindow**)

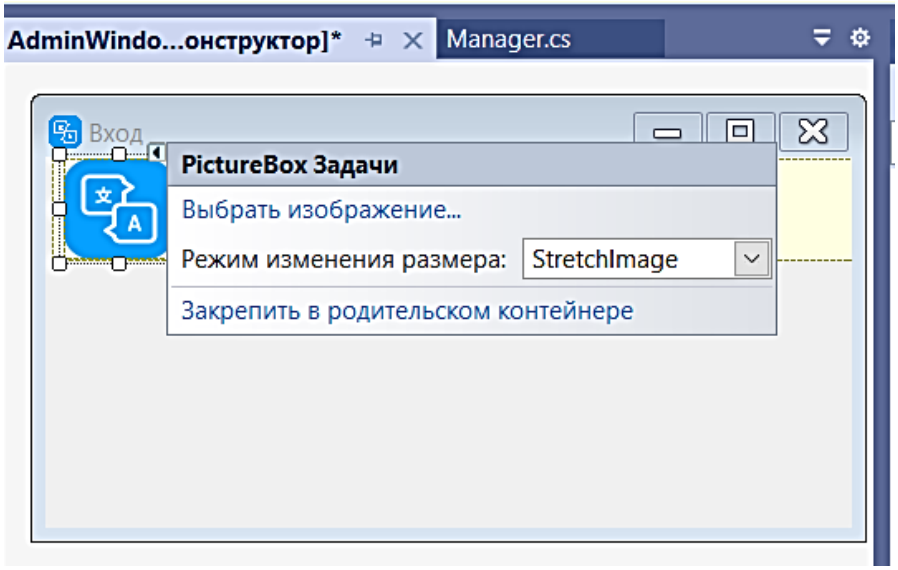
```
15          static void Main()
16          {
17              Application.EnableVisualStyles();
18              Application.SetCompatibleTextRenderingDefault(false);
19              Application.Run(new AdminWindow());
20          }
```

форма AdminWindow

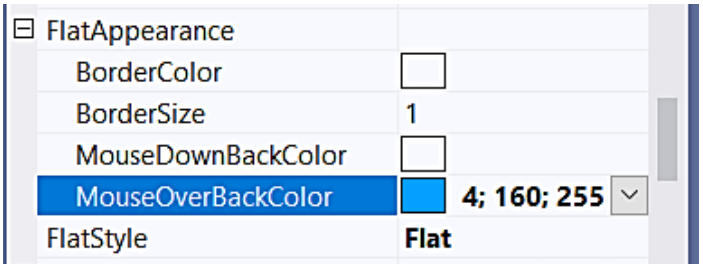
1. Разработайте внешний вид формы авторизации:



2. Изменить свойства формы:
   - ✓ **Icon** – выбрать наш значок, добавленный в **Ресурсы**
   - ✓ Значения **MaximizeBox**, **MinimizeBox = False**
   - ✓ **Text**
3. На форму добавить элементы:
   - ✓ **Panel** в верхней части формы, установить для него цвет
   - ✓ **PictureBox** и **Label** на **Panel**, изменить свойства изображения



4. Добавить, настроить остальные элементы и дать им соответствующие имена:
   - ✓ тестовые поля - **TxtBoxLogin**, **TxtBoxPassword**
   - ✓ кнопки – **BtnOk**, **BtnCancel**
5. Для кнопок настроить свойство **DialogResult**. Для акцентирования кнопок измените значения свойства ***FlatAppearance***:



6. Можно также задать цвет кнопки при нажатии кнопки мыши на ней.
7. Заполните в БД таблицу **User** необходимыми данными
8. В соответствии с руководством по стилю установите шрифт для элементов формы: выделите название формы **AdminWindow.cs,** нажмите **ПКМ** и выберите **Перейти к коду**. Затем после инициализации формы задайте следующий код:

```
public AdminWindow()
{
    InitializeComponent();
    //задание шрифта для элементов формы
    this.Font = new Font("Comic Sans MS", 12, FontStyle.Bold);
}
```

9. Файл программного кода **AdminWindow.cs**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using System.Windows.Forms;

// для работы с моделями вашей БД нужно добавить путь вида
// НазваниеВашегоПриложения.Models
using Var_School.Models;

namespace Var_School
{
    /// <summary>
    /// Логика взаимодействия для AdminWindow.cs
    /// </summary>
    Ссылок: 4
    public partial class AdminWindow : Form
    {
        Ссылок: 1
        public AdminWindow()
        {
            InitializeComponent();
        }
```

```
private void BtnOk_Click(object sender, EventArgs e)
{
    try
    {  //загрузка всех пользователей из БД в список
        List<User> users = VarSchoolEntities.GetContext().Users.ToList();
        //попытка найти пользователя с указанным паролем и логином
        //если такого пользователя не будет обнаружено то переменная u будет равна null
        User u = users.FirstOrDefault(p => p.username == TxtBoxLogin.Text && p.password == TxtBoxPassword.Text);

        if (u != null)
        {
            // логин и пароль корректные, запускаем главную форму приложения
            MainForm mainWindow = new MainForm();
            mainWindow.Owner = this;
            this.Hide();
            mainWindow.Show();
        }
        else
        {
            MessageBox.Show("Не верный логин или пароль");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString());
    }
}

//код кнопки Отмена
Ссылок: 0
private void BtnCancel_Click(object sender, EventArgs e)
{
    this.Close();
}
```

10. Проверьте работу.
11. Реализуйте сокрытие введенных в текстовое поле формы символов пароля.

12. А также установите обратную связь с пользователем при попытке закрытия приложения – дважды выберите для формы администрирования событие **FormClosing** и задайте следующий программный код:

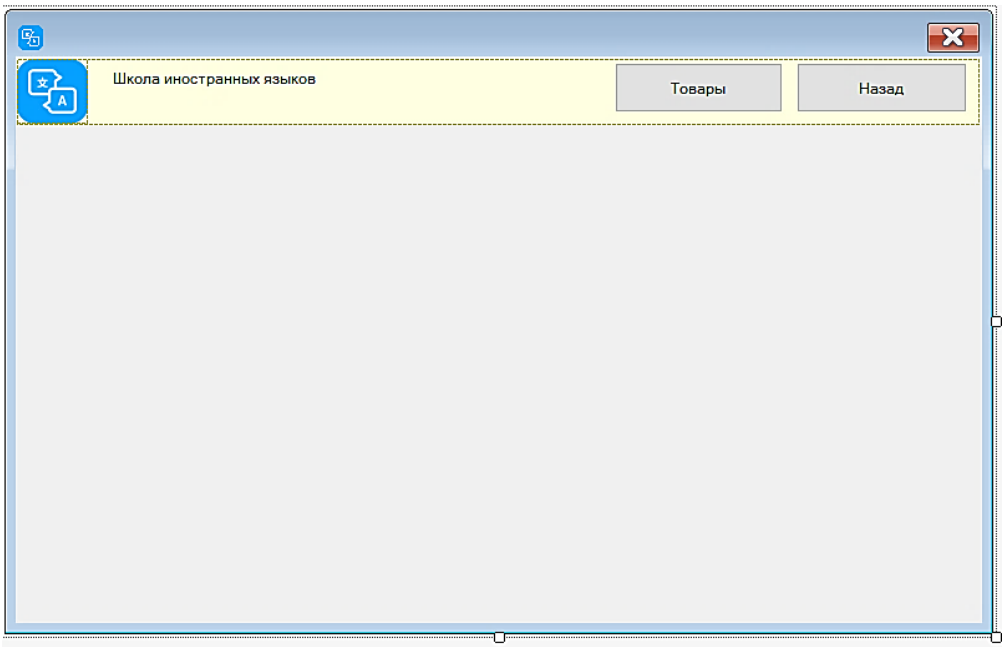
```
using MessageBox = System.Windows.MessageBox;

// попытка закрыть приложение
Ссылок: 1
private void AdminWindow_FormClosing(object sender, FormClosingEventArgs e)
{
    //на экране отображается форма с двумя кнопками
    MessageBoxResult x = MessageBox.Show("Вы действительно хотите закрыть приложение?",
        "Выйти", MessageBoxButton.OKCancel, MessageBoxImage.Question);
    if (x == MessageBoxResult.Cancel)
        e.Cancel = true;
}
```

Примечание: При появлении ошибки и в процессе решения проблемы добавьте ссылку на ***PresentationFramework***

Главная форма приложения - форма MainForm

1. Разработайте главную форму приложения



2. В соответствии с руководством по стилю установите шрифт для элементов формы, аналогично формы администрирования.
3. Реализуйте переход по кнопке **Назад** на форму авторизации (назначить событие двойным щелчком по кнопке), например, следующим образом

```
private void BtnBack_Click(object sender, EventArgs e)
{
    AdminWindow f2 = new AdminWindow();
    this.Hide();
    f2.ShowDialog();
}
```

или `f2.Visible = true;`

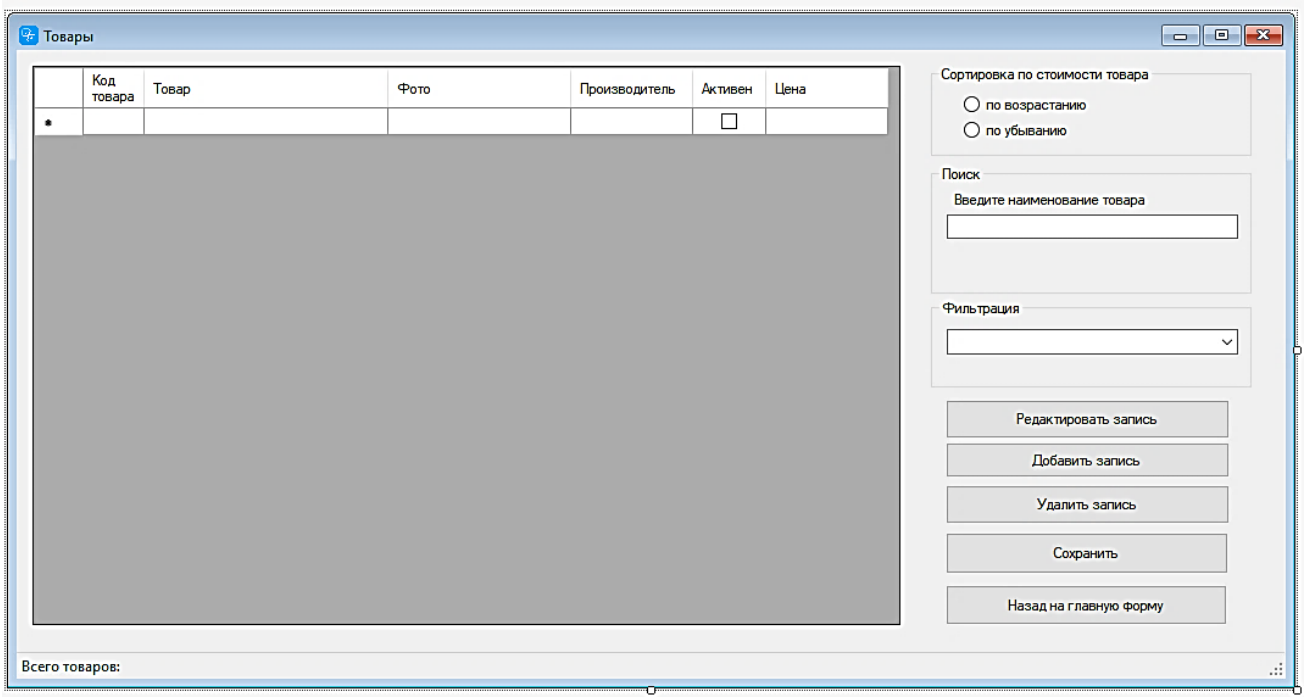
Форма списка GoodsForm

1. Добавьте в проект новую форму **GoodsForm.**
2. Установите переход по кнопке **Товары** из формы **MainForm,** например:

```
private GoodsForm Goods;
Ссылок: 1
private void BtnEditGoods_Click(object sender, EventArgs e)
{
    Goods = new GoodsForm();
    Goods.Visible = true;
}
```

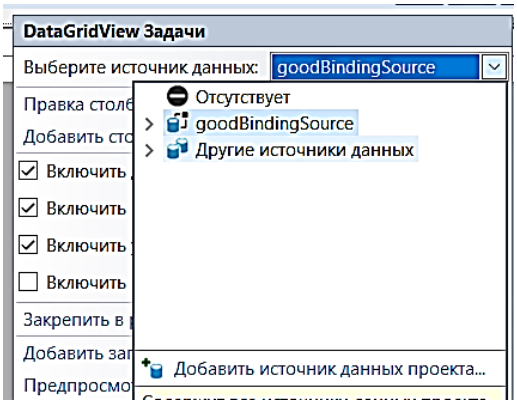
При необходимости можно скрыть предыдущее текущее окно главной формы перед открытием формы со списком: `this.Hide();`

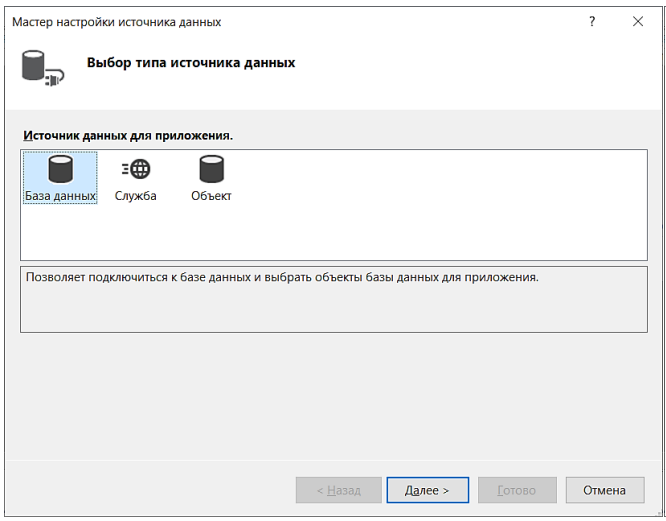
Проверьте работу кнопок

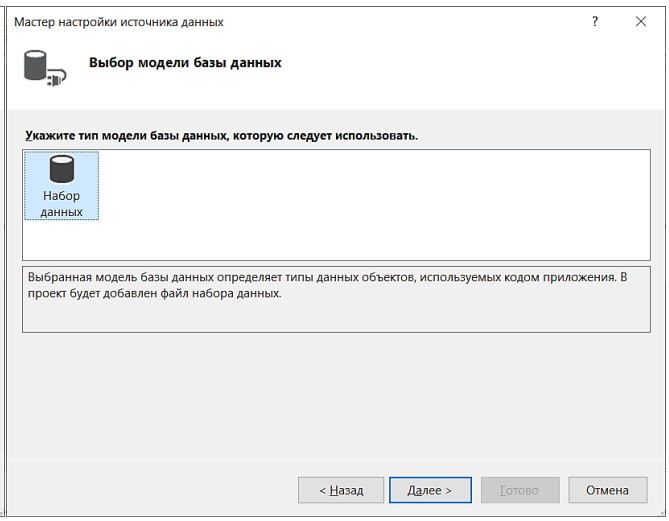


3. Измените иконку формы
4. В соответствии с руководством по стилю установите шрифт для элементов формы
5. Задайте для формы минимальный и максимальный размер, который она может иметь
6. Добавьте на форму из вкладки «**Данные**» компонент **DataGridView**

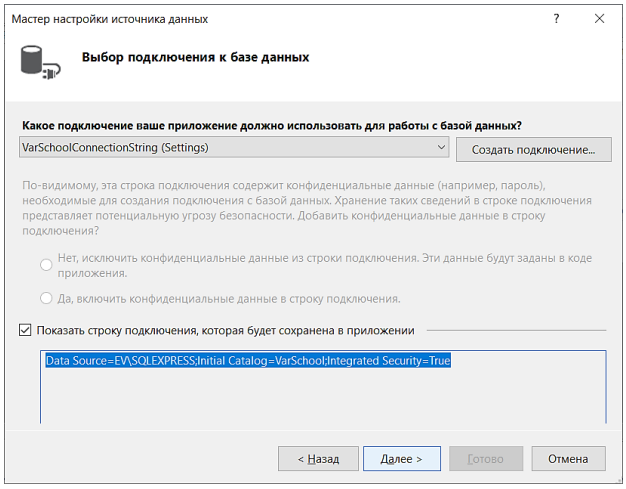
7. Далее необходимо настроить свойства компонента **dataGridView1**
   - ✓ Переименовать в **dataGridViewGood**
   - ✓ в области задач **dataGridViewGood** выбрать **Добавить источник данных проекта...**
   - ✓ Выбрать **База данных – Далее – Набор данных - Далее**

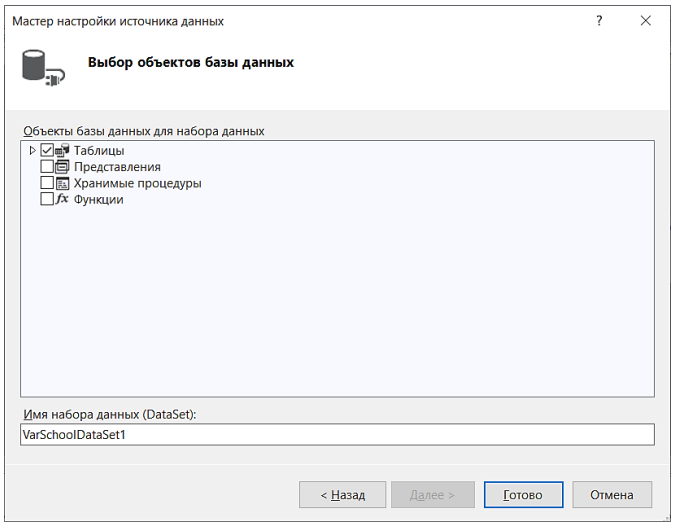






✓ Выбрать свою базу данных из списка, просмотреть строку подключения – **Далее**, установить флажок **Таблицы - Готово**

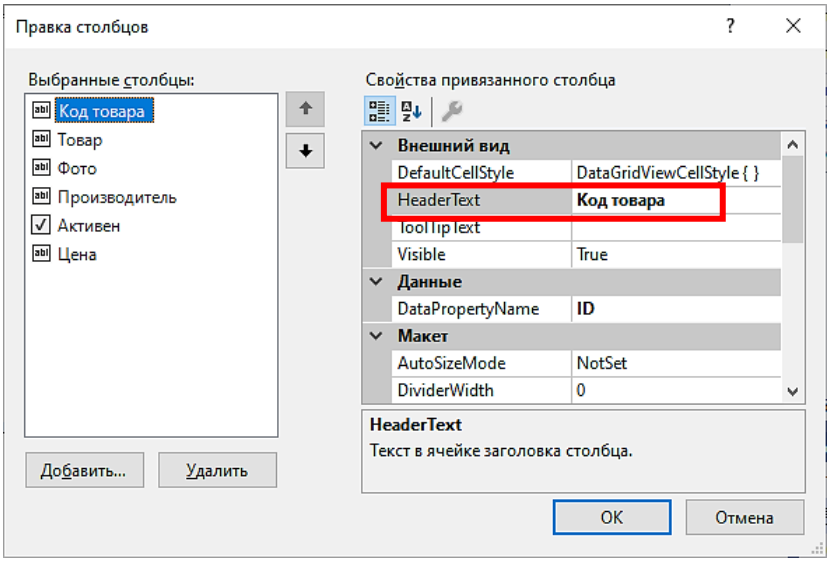




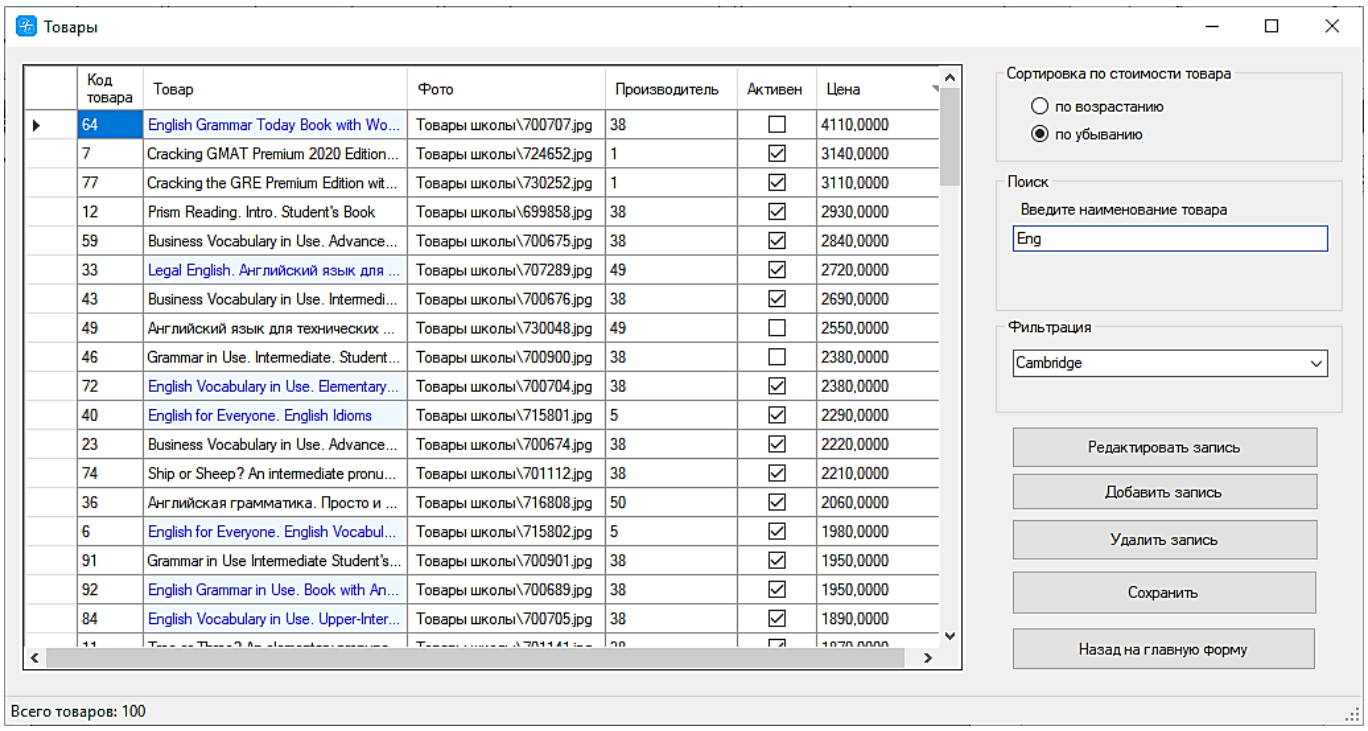
✓ Далее из появившегося списка **goodBindingSource** выберите таблицу **Product**

✓ Появятся заголовки столбцов таблицы.

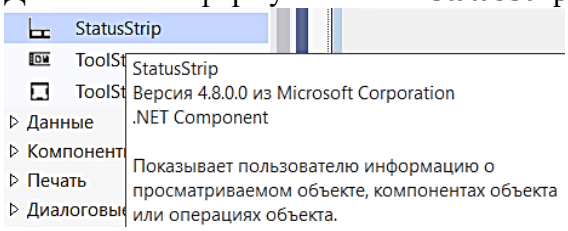
8. Настройте отображение таблицы - в **dataGridViewGood** разверните задачи и выберите **Правка столбцов…**
   - переименуйте столбцы (дайте им названия на русском языке): для каждого поля изменить свойство **HeaderText**, например:
   - установите для столбца **Код товара** свойство **Только для чтения**
   - остальные свойства столбцов настройте на свое усмотрение



9. Создайте на форме элементы для дальнейшей обработки данных:



10. Добавьте на форму элемент StatusStrip для отображения количества записей в таблице



и напишите код строки подсчета количества строк в таблице (за исключение заголовка) в обработчик загрузки табличной формы списка (в моем случае это GoodsForm_Load)

```
private void GoodsForm_Load(object sender, EventArgs e)
{
    // TODO: данная строка кода позволяет загрузить данные в таблицу "varSchoolDataSet.Dev
    this.developerTableAdapter.Fill(this.varSchoolDataSet.Developer);
    // TODO: данная строка кода позволяет загрузить данные в таблицу "varSchoolDataSet.Goo
    this.goodTableAdapter.Fill(this.varSchoolDataSet.Good);
    toolStripStatusLabel1.Text = $"Всего товаров: {dataGridViewGood.Rows.Count - 1}";

}
```

11. Реализуйте сортировку:

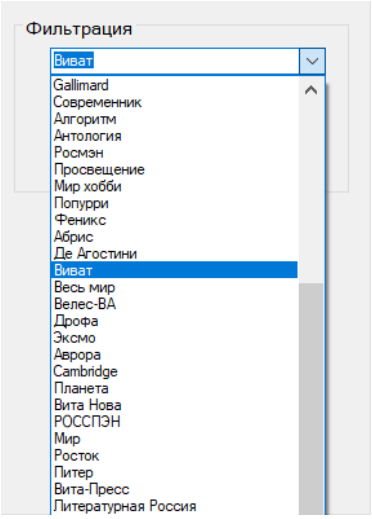
```
private void RadioBtnSortVozr_CheckedChanged(object sender, EventArgs e)
{
    if (RadioBtnSortVozr.Checked)
        dataGridViewGood.Sort(priceDataGridViewTextBoxColumn, System.ComponentModel.ListSortDirection.Ascending);
    else
        dataGridViewGood.Sort(priceDataGridViewTextBoxColumn, System.ComponentModel.ListSortDirection.Descending);

}
```

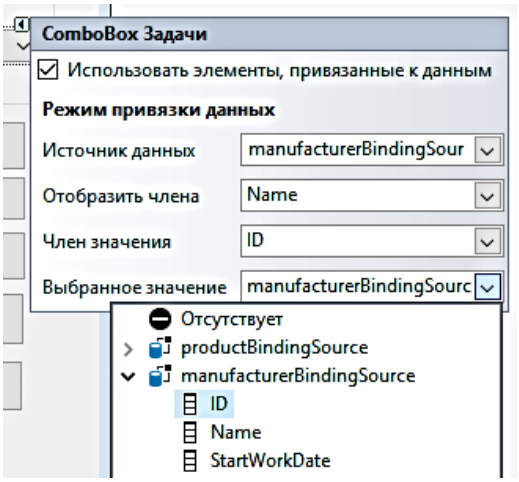
12. Реализуйте поиск любым известным способом.
Образец поиска выделением найденных элементов:

```
private void TxtBoxPoisk_TextChanged(object sender, EventArgs e)
{
    //перебирает все ячейки таблицы и устанавливает в них белый цвет фона
    // и чёрный цвет текста, то есть отменяет результаты предыдущего поиска
    for (int i = 0; i < dataGridViewGood.ColumnCount - 1; i++)
    {
        for (int j = 0; j < dataGridViewGood.RowCount - 1; j++)
        {
            dataGridViewGood[i, j].Style.BackColor = Color.White;
            dataGridViewGood[i, j].Style.ForeColor = Color.Black;
        }
    }
    //перебирает все ячейки таблицы и если они содержат текст, введённый в поле ввода (TxtBoxPoisk),
    // то устанавливает в них голубой цвет фона и синий цвет текста, чем выделяет искомые ячейки.
    for (int i = 0; i < dataGridViewGood.ColumnCount - 1; i++)
    {
        for (int j = 0; j < dataGridViewGood.RowCount - 1; j++)
        {
            if (dataGridViewGood[i, j].Value.ToString().IndexOf(TxtBoxPoisk.Text) != -1)
            {
                dataGridViewGood[i, j].Style.BackColor = Color.AliceBlue;
                dataGridViewGood[i, j].Style.ForeColor = Color.Blue;
            }
        }
    }
}
```

13. Попробуйте самостоятельно задать: если товар не активен, то фон элемента должен быть серого цвета, в остальных случаях - белого.
14. Настройте выпадающий список фильтрации по производителю, т.е. установите связь с таблицей ***Производители*** с выводом их названий:



Отобразите меню действий выпадающего списка. Включите опцию «**Использовать элементы, привязанные к данным**». В строке **Источник данных** выберите **Другие источники данных, Источники данных проекта, НаименованиеБДDataSet, manufacturerBindingSource**. В строке **Отобразить члена** выберите **Name.** В строке **Член значения** по какому полю искать данные - в нашем случае идентификатор производителя **Id** и в строке **Выбранное значение** указать какое поле связанно с таблицей производителей - **manufacturerBindingSource – ID**

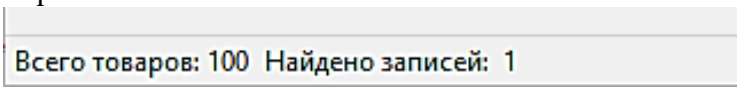


*В панели действий под опцией "Использовать связанные с данными элементы списка (**Use data bound items**") расположены следующие параметры:*

- ***Data Source (Источник данных)** - определяет таблицу или запрос из которого заполняется список;*
- ***Display Member (Член отображения)** - определяет поле значениями которого заполняется список;*
- ***Value Member (Член значений)** - определяет значения какого поля подставляются в связанное с выпадающим списком поле;*
- ***Selected Value (Выбранное значение)** - определяет связанное с выпадающим списком поле.*

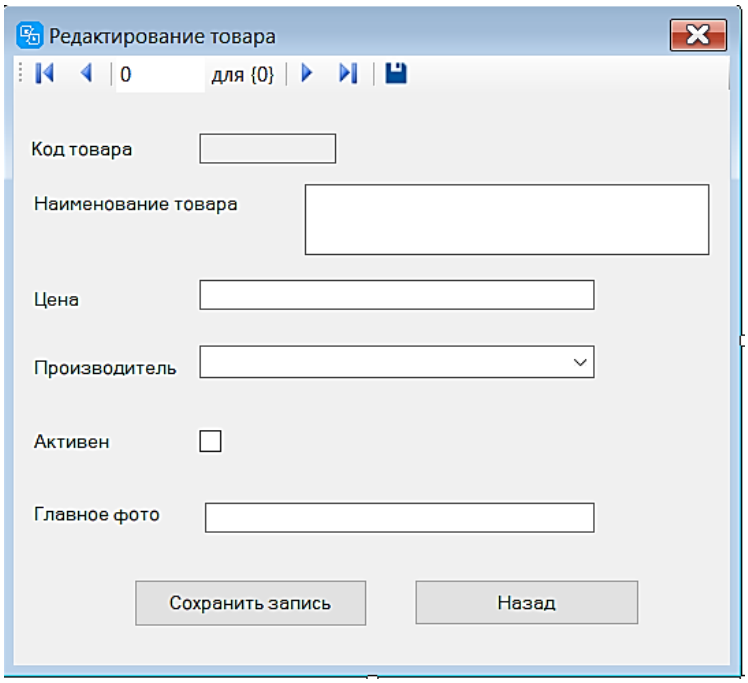
15. Реализуйте фильтрацию.
16. Добавьте на форму еще один элемент StatusStrip для отображения количества найденных записей в таблице и напишите код строки подсчета количества найденных строк в таблице (за исключение заголовка) для обработчика события **Paint** элемента **dataGridViewGood**

Образец вывода данных:



Форма редактирования:

Код товара установить свойство **ReadOnly** значение **True**



Вызов формы редактирования при нажатии на кнопку Редактировать запись

```
}
private EditGood goods;
Ссылок: 1
private void BtnEdit_Click(object sender, EventArgs e)
{
    goods = new EditGood();
    goods.Visible = true;
}
```

При необходимости можно спрятать предыдущую форму товаров

Программный код элементов формы и кнопок:

```
public partial class EditGood : Form
{
    Ссылок 1
    public EditGood()
    {
        InitializeComponent();
        //задание шрифта для элементов формы
        this.Font = new Font("Comic Sans MS", 10, FontStyle.Regular);
    }
    Ссылок 1
    private void goodBindingNavigatorSaveItem_Click(object sender, EventArgs e)
    {
        this.Validate();
        this.goodBindingSource.EndEdit();
        this.tableAdapterManager.UpdateAll(this.varSchoolDataSet);
    }
    Ссылок 1
    private void AddGood_Load(object sender, EventArgs e)
    {
        // TODO: данная строка кода позволяет загрузить данные в таблицу "varSchoolDataSet.Developer".
        // При необходимости она может быть перемещена или удалена.
        this.developerTableAdapter.Fill(this.varSchoolDataSet.Developer);
        // TODO: данная строка кода позволяет загрузить данные в таблицу "varSchoolDataSet.Good".
        // При необходимости она может быть перемещена или удалена.
        this.goodTableAdapter.Fill(this.varSchoolDataSet.Good);
    }
```

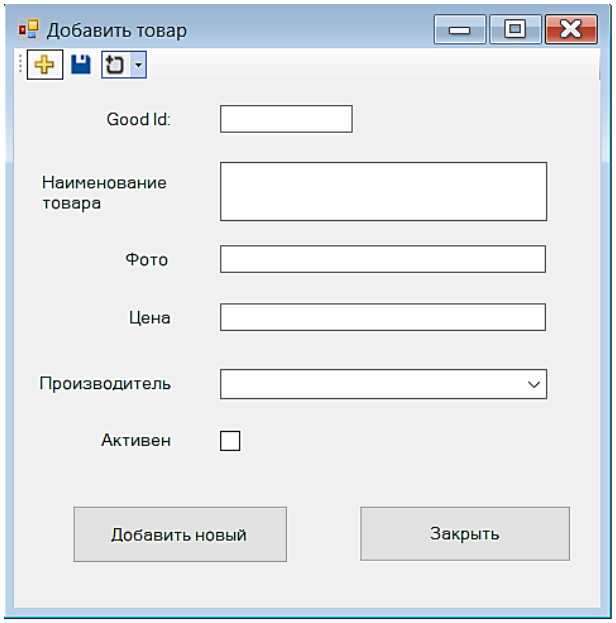
```
private void BtnSaveGood_Click(object sender, EventArgs e)
{
    this.Validate();
    this.goodBindingSource.EndEdit();
    this.tableAdapterManager.UpdateAll(this.varSchoolDataSet);
}

private void BtnNBack_Click(object sender, EventArgs e)
{
    this.Hide(); // прячем текущее окно
    this.tableAdapterManager.UpdateAll(this.varSchoolDataSet);
}
```

Удаление выделенных записей с подтверждением:

```
private void BtnDel_Click(object sender, EventArgs e)
{
    DialogResult dr = MessageBox.Show("Удалить запись?", "Удаление", MessageBoxButtons.OKCancel,
        MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2);
    if (dr == DialogResult.OK)
    {
        foreach (DataGridViewRow row in dataGridViewGood.SelectedRows)
            dataGridViewGood.Rows.Remove(row);
    }
    else
        this.Close();
}
```

Форма добавления



Вызов формы:

```
private void BtnAdd_Click(object sender, EventArgs e)
{
    AddGood addGood = new AddGood();
    addGood.Visible = true;
}
```

17. Реализуйте добавление товаров в таблицу с сохранением в БД.