

Уральский государственный колледж имени И. И. Ползунова

Государственное автономное профессиональное образовательное
учреждение Свердловской области
«Уральский государственный колледж имени И.И. Ползунова»



Раздел № 4

«Приложение базы данных»

Лабораторная работа № 4.1

«Разработка приложения базы данных»

Цель: разработать пользовательское приложение для работы с базой данных.

Екатеринбург, 2022

СОЗДАНИЕ ПРОЕКТА В VISUAL STUDIO

1. Запустите Visual Studio.
2. В правой части открывшегося окна, необходимо выбрать «Создание проекта», как показано на рисунке 1.

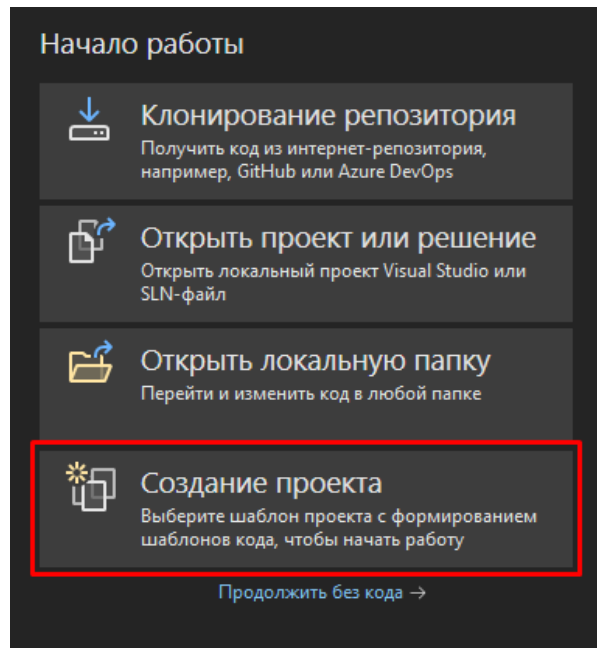


Рисунок 1 – Создание проекта

3. Выбираем шаблон «Приложение Windows Forms», как показано на рисунке 2. После этого необходимо указать название проекта («Агентство недвижимости») и его расположение на диске

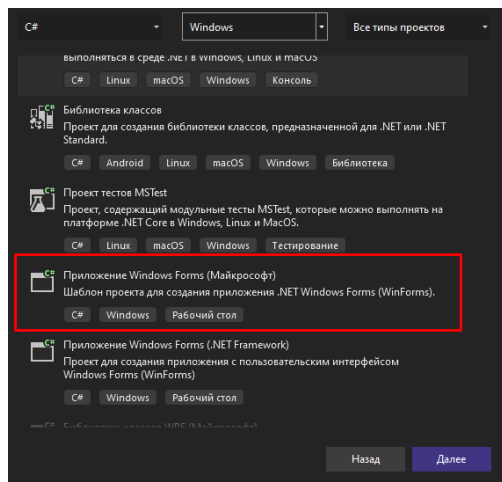


Рисунок 2 – Выбор шаблона проекта

СОЗДАНИЕ ФОРМЫ ПОДКЛЮЧЕНИЯ К БАЗЕ ДАННЫХ

1. Необходимо добавить новую форму Windows в проект. В обозревателе решений правой кнопкой мыши нажимаем на сам проект («Агентство недвижимости»), затем «Добавить» и выбираем «Форма (Windows Forms)», как показано на рисунке 3.

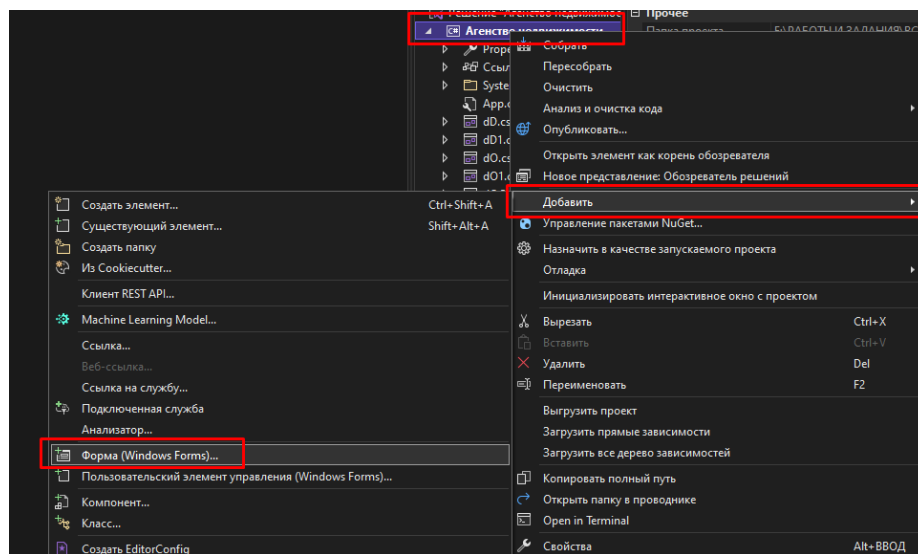


Рисунок 3 – Создание формы

2. В предложенном списке необходимо выбрать «Форма (Windows Forms)». Форму создадим с именем «login.cs», как показано на рисунке 4.

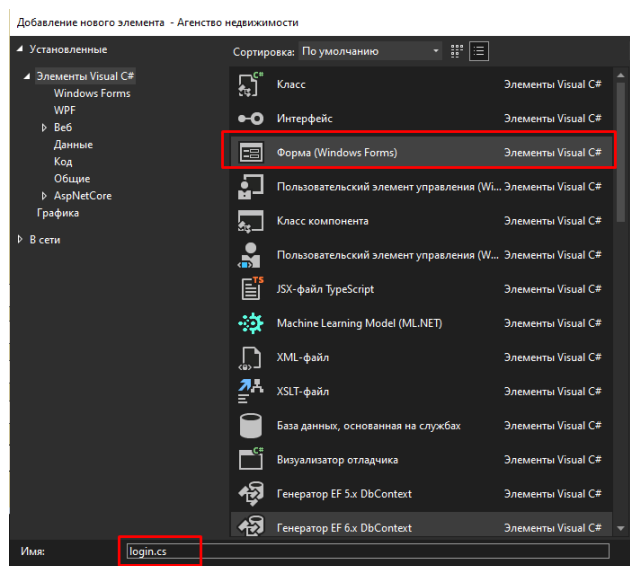


Рисунок 4 – Выбор нужной формы и присваивание ей имени

3. С помощью «Панели элементов», добавим в форму «login.cs» следующие элементы: 1 «Button» (кнопка, при нажатии на которую выполняется какая-либо операция), 4 «Label» (используется для вывода текста или изображения, которые не могут быть изменены пользователем), 4 «TextBox» (используется для получения входных данных от пользователя). Элементы «Label» и «TextBox» необходимо расположить так, чтобы пользователю системы было понятно, какая функция принадлежит каждому полю. За изменение подписи элемента отвечает свойство «Text» в окне свойств элемента. Пример оформления формы для подключения к базе данных приведен на рисунке 5.

Подключение к базе данных

Агентство недвижимости "Свой ключ"

Войдите в систему

label1
Адрес сервера

label2
Имя базы данных

label3
Имя пользователя

label4
Пароль пользователя

textBox1

textBox2

textBox3

textBox4

Соединиться button1

Рисунок 5 – Пример оформления окна подключения к БД

4. Для соединения формы подключения к базе данных («login.cs») с главной формой («Form1.cs»), перейдем в класс главной формы («Form1.Designer.cs»). Затем добавим фрагмент кода, помеченный на рисунке 6. «logwin» — это название объекта подкласса.

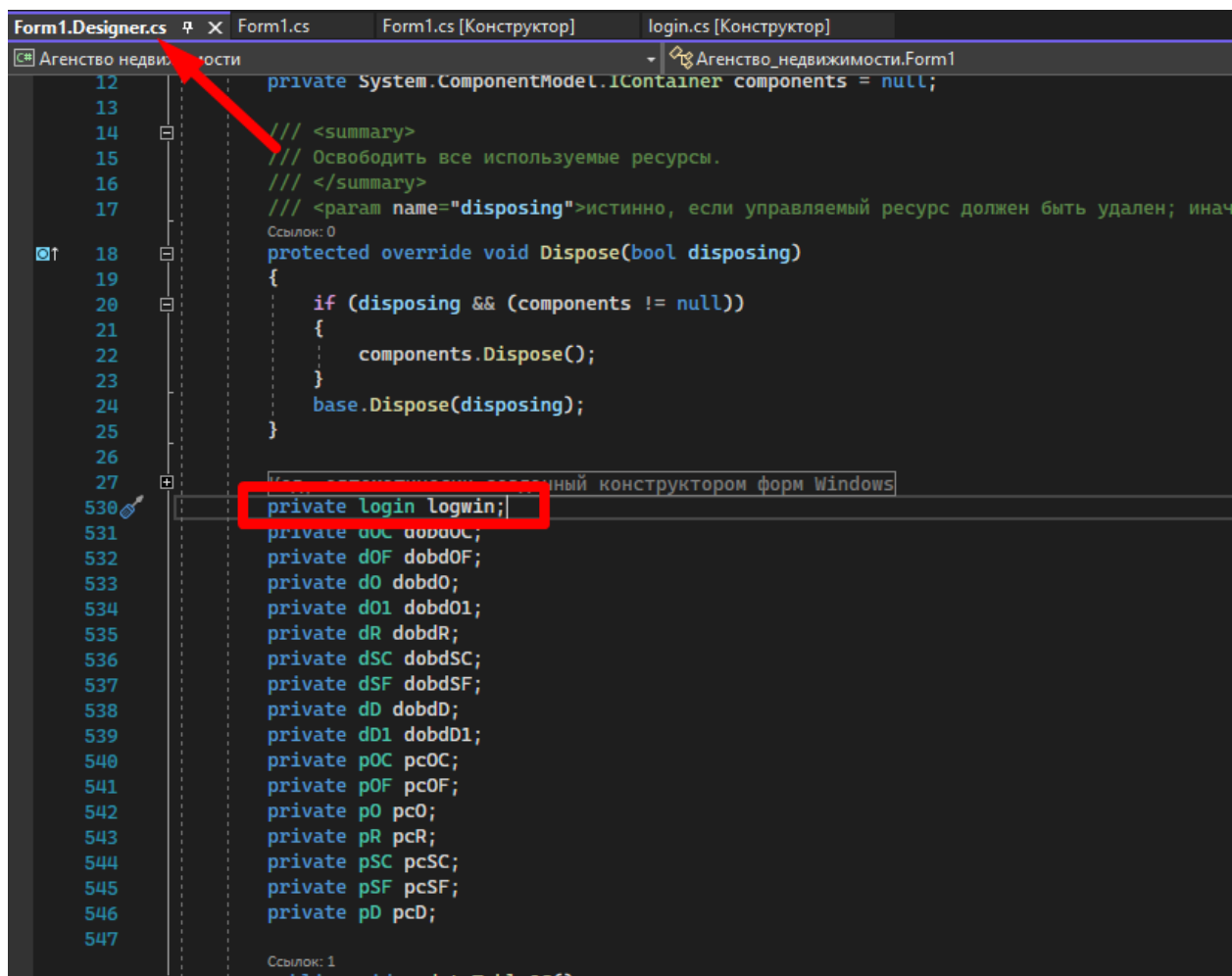


Рисунок 6 – Создание объекта подкласса

5. Перейдем в конструктор главной формы и добавим элемент «menuStrip». Расположим данный элемент в самом верху главной формы. Первый пункт назовем «Меню» и подпункт «Подключиться к БД», как показано на рисунке 7. С помощью этого подпункта будет выполняться вызов окна «login.cs». Два раза кликаем по подпункту «Подключиться к БД», чтобы перейти к коду данного раздела.

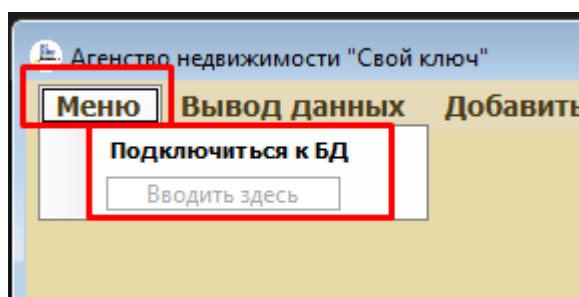


Рисунок 7 – Настройка элемента «menuStrip»

6. В открывшемся окне необходимо добавить фрагмент кода, представленный на рисунке 8. В данном фрагменте кода выполняется физическое создание подкласса «logwin», а также отображение содержимого окна «login.cs», с помощью параметра модальности (т.е. окно соединения будет преобладать над другими окнами)

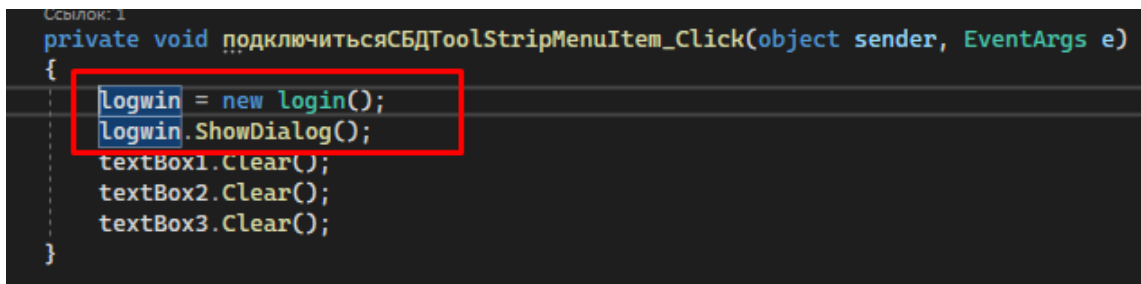


Рисунок 8 – Физическое создание подкласса «logwin»

7. Далее необходимо описать соединение с базой данных. Перейдем к классу окна «login.cs» («login.Designer.cs»), с помощью окна «Обозреватель решений». В классе описываем объект SqlConnection (он необходим для того, чтобы соединиться с базой данных и проводить различные операции в ней) с названием «conn», как показано на рисунке 9. Также необходимо добавить библиотеку, которая позволяет работать с SqlConnection (SqlClient). В самом начале кода необходимо добавить следующий фрагмент кода: using System.Data.SqlClient;

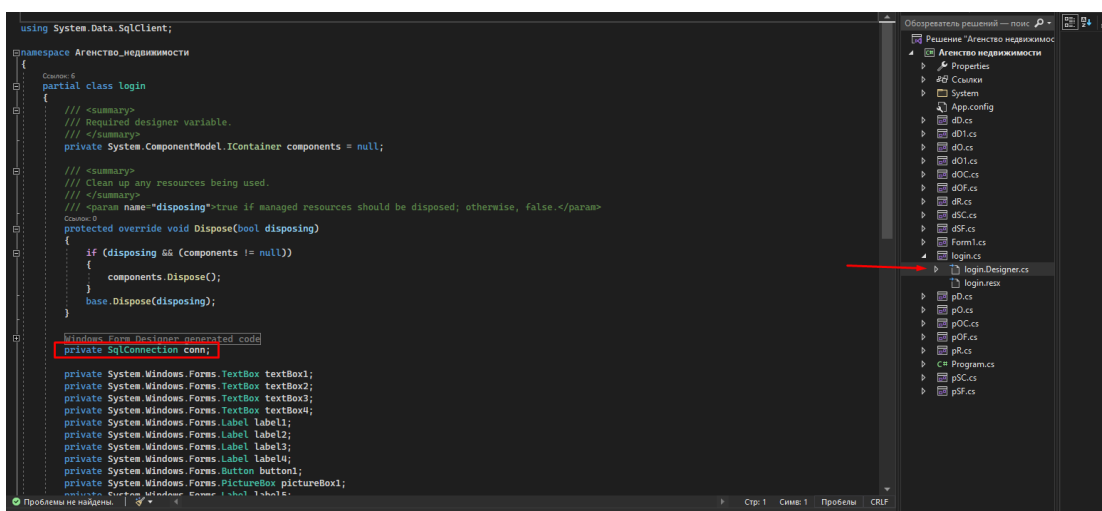


Рисунок 9 – Описание соединения с БД

8. Чтобы пользователю не нужно было при постоянно вводить параметры соединения при работе с приложением, необходимо создать глобальный класс в разделе «Program.cs» (этот раздел находится в обозревателе решений), у которого будет статичная глобальная переменная, в которую будет записываться строка соединения. Фрагмент кода представлен на рисунке 10.

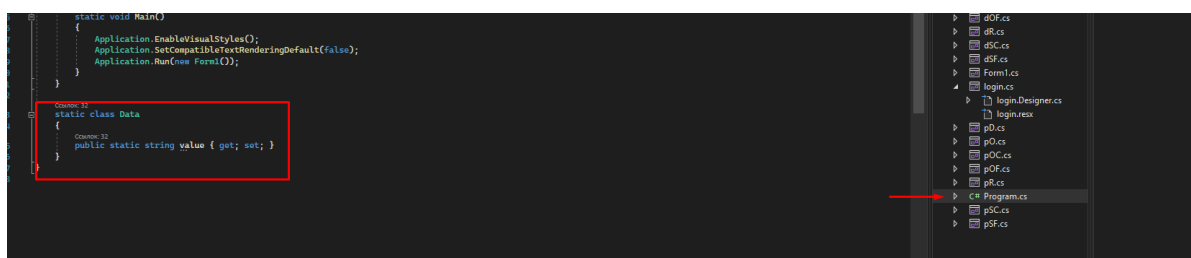


Рисунок 10 – Создание глобального класса

9. Для того, чтобы выполнить соединение с базой данных, необходимо описать элемент «Button1», который расположен в форме «login.cs». Два раза кликаем по данному элементу и редактируем код, как представлено на рисунке 11.

Пояснение к коду.

Физически создается объект «conn», с помощью которого выполняется подключение к базе данных через библиотеку `SqlClient`, где в качестве параметров необходимо добавить строку соединения с БД (данные будут считываться с полей `textbox`, ранее добавленные в форму «login»). При удачной операции, выполняется соединение, с помощью метода `Open` без параметров. При удачном соединении, пользователю показывается окно, информирующее его о том, что соединение выполнено успешно. После этого окно соединения закрывается и происходит запись в статичную глобальную переменную «value». В случае, если операция не была выполнена успешно, пользователю на экран будет выведена ошибка, из-за которой не удалось выполнить соединение.

```
Свой код: 1
private void button1_Click(object sender, EventArgs e)
{
    using (conn = new System.Data.SqlClient.SqlConnection("Server=" + textBox1.Text + ";Database=" + textBox2.Text + ";uid=" + textBox3.Text + ";pwd=" + textBox4.Text + ""))
    {
        try
        {
            conn.Open();
            MessageBox.Show("Вы подключились к базе данных");
            this.Close();

            Data.value = "Server=" + textBox1.Text + ";Database=" + textBox2.Text + ";uid=" + textBox3.Text + ";pwd=" + textBox4.Text + "";
        }
        catch (Exception help)
        {
            MessageBox.Show(help.Message);
        }
    }
}
```

Рисунок 11 – Описание элемента «Button1»

10. Выполним пробное подключение к базе данных, как показано на рисунке 12. Для подключения можно использовать любого ранее созданного пользователя, но он должен обладать правами администратора.

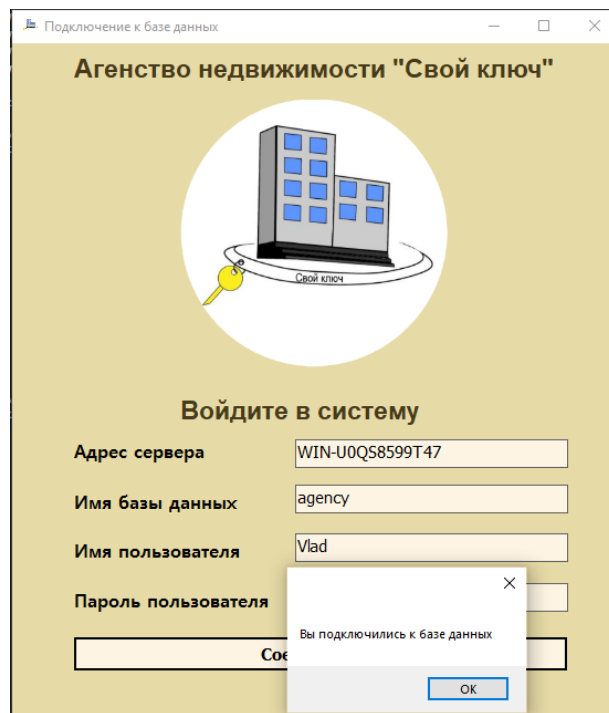


Рисунок 12 – Подключение к БД

ВЫВОД ДАННЫХ ИЗ ТАБЛИЦЫ

1. Разместим на главной форме объект «DataGridView», как показано на рисунке 13. Этот объект нужен для того, чтобы отображать данные, взятые из таблицы в базе данных

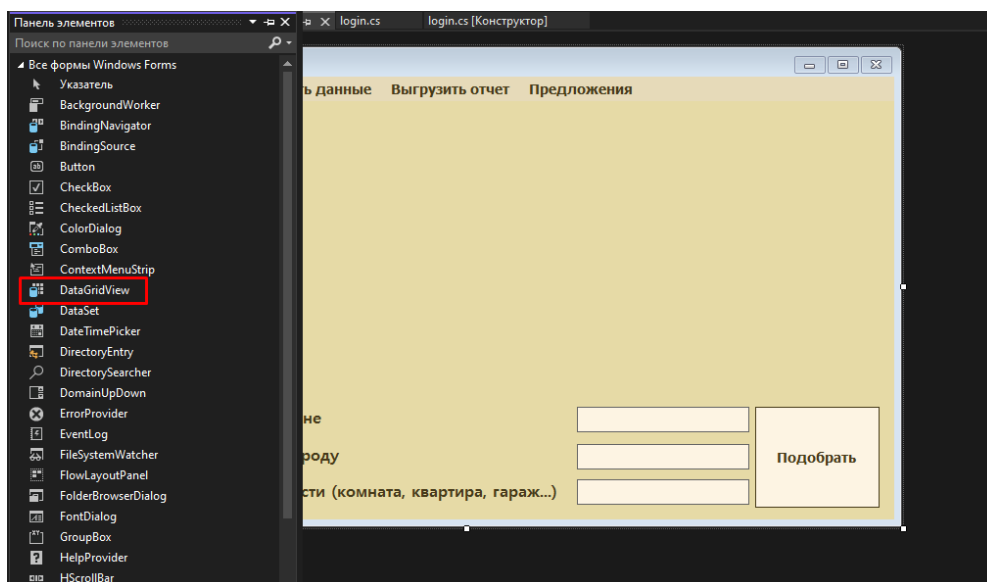


Рисунок 13 – Объект «DataGridView»

2. Также необходимо добавить новый пункт в уже добавленном объекте «MenuStrip». Этот пункт назовем «Вывод данных», а первым подпунктом сделаем «Риэлторы», как показано на рисунке 14.

То есть данные будем выводить из таблицы «Риэлторы» (realtors) в объект «DataGridView»

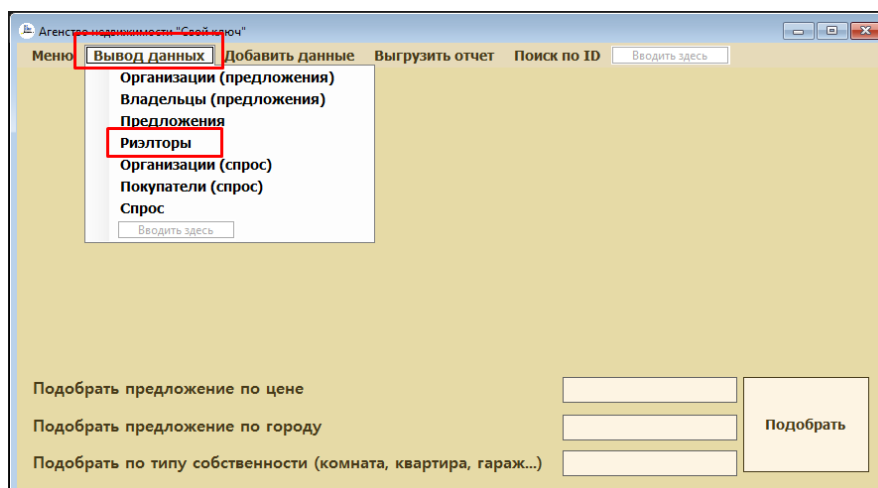


Рисунок 14 – Настройка элемента «MenuStrip»

3. Перейдем к классу главной формы и добавим библиотеки `System.Data` и `System.Data.SqlClient`, как показано на рисунке 15.

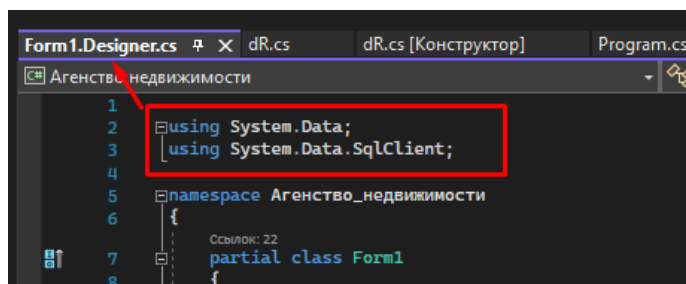
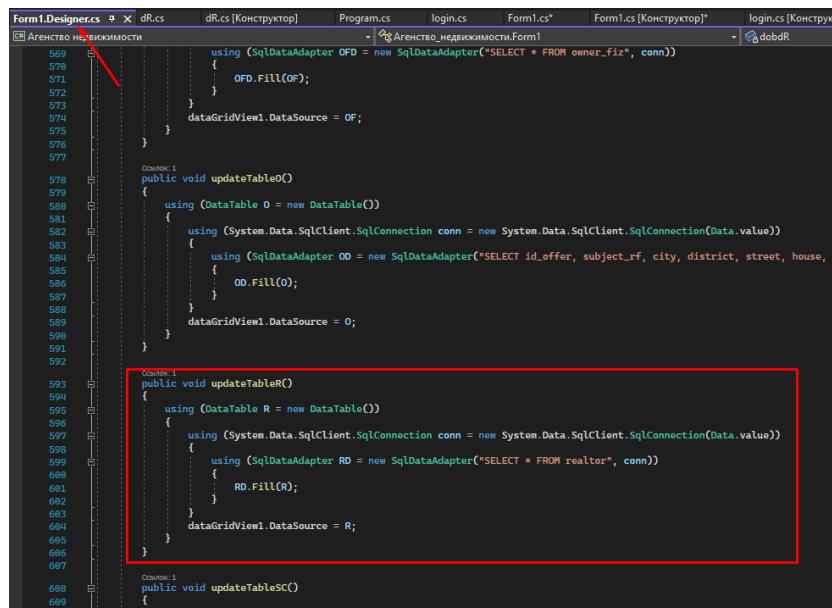


Рисунок 15 – Добавление библиотек

4. Вывод данных из таблицы «Риэлторы» будет осуществляться посредством процедуры. Код процедуры представлен на рисунке 16.

Процедура является публичной и имеет название «updateTableR» без параметров. С помощью конструкции «using» создается объект «R» от класса «DataTable», который распределяет полученные данные по таблице. Внутри тела конструкции «using» необходимо вновь открыть соединение с базой данных, где в качестве параметров будет строка соединения, которая хранится в публичном классе «Data». Далее создается объект, который хранит все данные, полученные из базы данных («SqlDataAdapter») и в дальнейшем передаст эти данные в объект «DataGridView». В качестве параметров передается инструкция, позволяющая выполнить выборку данных из базы данных («SELECT»). После этого происходит заполнение объекта «R» и вывод содержимого в «DataGridView»



ДОБАВЛЕНИЕ ДАННЫХ В ТАБЛИЦУ

1. Необходимо добавить новый пункт в уже добавленном объекте «MenuStrip». Этот пункт назовем «Добавить данные», а первым подпунктом сделаем «Риэлторы», как показано на рисунке 19.

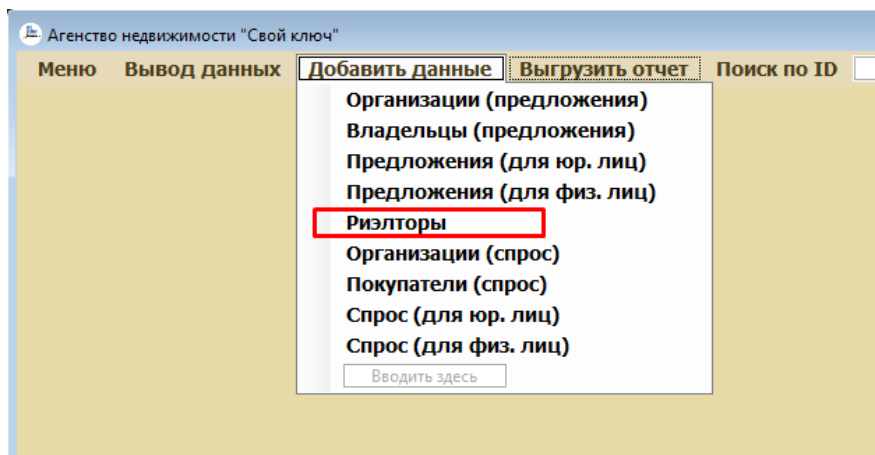


Рисунок 20 - Настройка элемента «MenuStrip»

2. Добавим новую форму Windows к проекту и назовем ее «dR». После этого разместим на ней следующие объекты: 4 Label, 4 TextBox, 1 Button, как показано на рисунке 20. Расположение элементов должно быть таким, чтобы пользователю было понятно, в какой столбец будет производиться ввод данных.

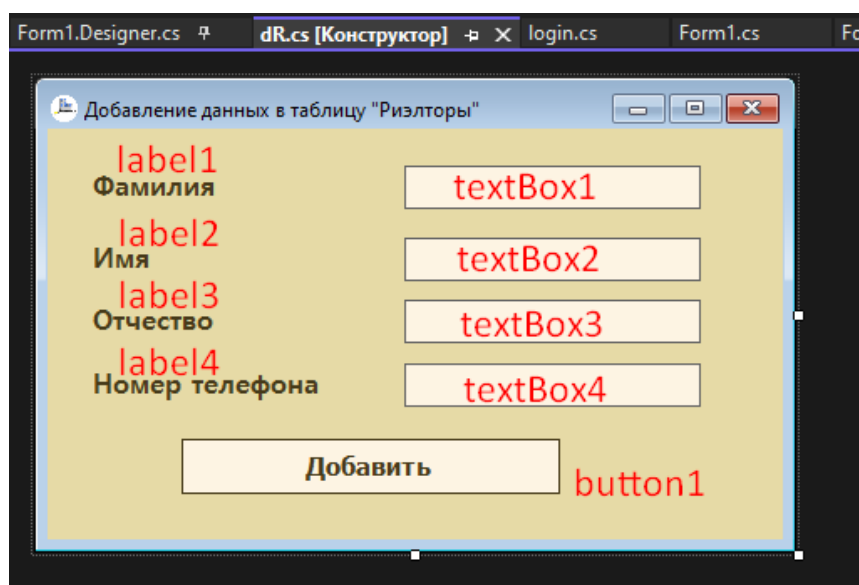


Рисунок 20 – Форма для добавления данных в таблицу

3. Перейдем к классу главной формы и добавим описание для класса «dR» с названием `dobdR`, как показано на рисунке 21

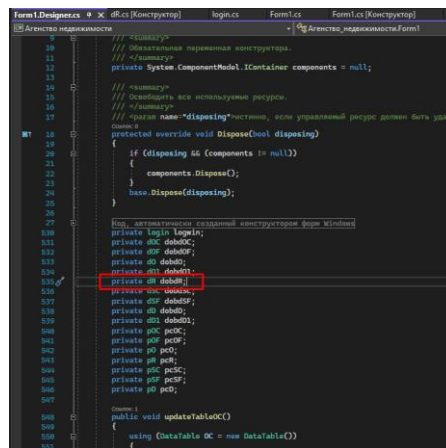


Рисунок 21 – Добавление описания для класса «dR»

4. Необходимо, чтобы при нажатии на подпункт «Риэлторы» в пункте «Добавить данные» на главной форме, происходил запуск формы «dR». Для этого необходимо отредактировать код подпункта «Риэлторы» так, как показано на рисунке 22.

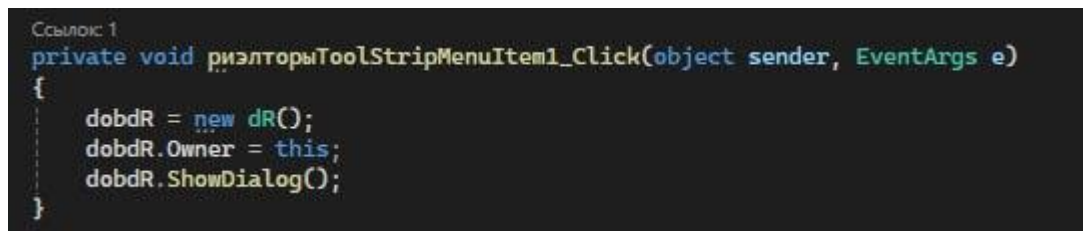


Рисунок 22 – Запуск формы «dR» с помощью кода

5. Проверьте, удалось ли открыть форму для добавления данных в таблицу «Риэлторы» (realtors).

6. Перейдем к коду объекта «button1», добавленного ранее во втором пункте. Отредактируйте код, как представлено на рисунке 23.

Используя конструкцию «using», устанавливаем соединение с базой данных, с помощью объекта класса «SqlConnection». Строковые данные находятся в классе «Data» в переменной «value».

Если соединение выполнилось, то необходимо его открыть. После этого, с помощью конструкции «using», требуется использовать объект

«SqlCommand» (этот объект позволяет добавлять данные в БД). В качестве параметра к этому объекту, передается SQL-конструкция, позволяющая добавить данные (INSERT INTO). Значения, которые будут добавлены в БД, будут считываться из элементов «textBox», ранее добавленные во втором пункте. После этого управление передается главной форме, где будут выведены добавленные данные, после обновления таблицы. Чтобы вывести данные, требуется, с помощью программы, нажать на кнопку, которые выводит данные (эта кнопка была добавлена в прошлой части этой лабораторной работы). Если во время добавления данных произошла ошибка, то с помощью «MessageBox», будет выведено сообщение с ошибкой.

Если соединение с БД выполнить не удалось, то с помощью «MessageBox», будет выведено сообщение с ошибкой.

```
private void button1_Click(object sender, EventArgs e)
{
    using (System.Data.SqlClient.SqlConnection conn = new System.Data.SqlClient.SqlConnection(Data.value))
    {
        try
        {
            conn.Open();
            using (SqlCommand cmd = new SqlCommand("INSERT INTO realtor (surname, name_realtor, patronymic, num_phone) VALUES ('" + textBox1.Text + "','" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "')", conn))
            {
                cmd.ExecuteNonQuery();
                Form1 ff = (Form1)this.Owner;
                ff.phantomToolStripMenu_Click(sender, e);
                MessageBox.Show("Добавлено");
                Close();
            }
        }
        catch (Exception helpExec)
        {
            MessageBox.Show(helpExec.Message);
        }
        catch (Exception helpConn)
        {
            MessageBox.Show(helpConn.Message);
        }
    }
}
```

Рисунок 23 – Описание элемента «button1»

7. Попробуйте добавить данные в таблицу «Риэлторы», используя созданную форму, как показано на рисунке 24

Добавление данных в таблицу "Риэлторы"

Фамилия: Яровой

Имя: Дмитрий

Отчество: Степанович

Номер телефона: +79006719810

Добавить

Добавлено

ОК

Рисунок 24 – Добавление данных в таблицу