

5. Работа с формой Клиент

1. Далее создаём форму, на которой будет располагаться интерфейс для клиентов (см. рисунок 1), для этого перейдём в вкладку «Проект», затем «Добавить форму». Зададим ей название FormClient.

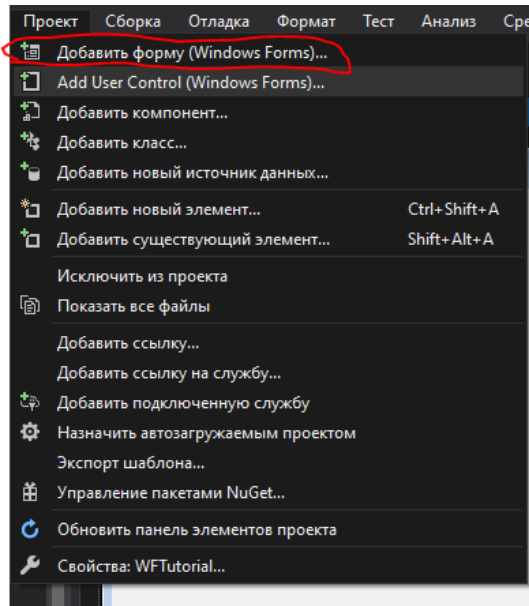


Рисунок 1 – Добавление формы

2. Сделаем ссылку на эту форму с кнопки «Клиенты»:
 - Перейдем в конструктор формы Menu.
 - Два раза щелкнем правой кнопкой мыши по кнопке «Клиенты» и добавим в метод вызова формы «Клиент» (см. рисунок 2).

```
private void buttonOpenClients_Click(object sender, EventArgs e)
{
    //Задаем новую форму из класса Клиент и открываем ее
    Form formClient = new FormClient();
    formClient.Show();
}
```

Рисунок 2 – Открытие формы «Клиент»

3. Перейдем снова на форму «Клиенты». Затем добавим необходимые элементы из «Панель элементов» (см. рисунок 3):

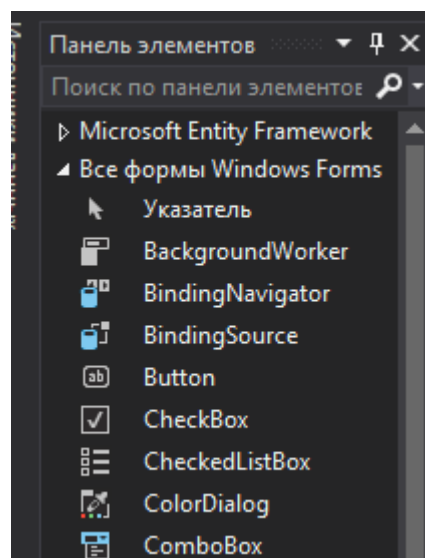


Рисунок 3 – Панель элементов

4. В данном интерфейсе нам нужны: textbox (5), label (5), listView (1), button (3).
Перетаскиваем на форму и переименовываем каждый элемент для удобства.

Наименование (Name) элементов textbox:

- textBoxFirstName
- textBoxMiddleName
- textBoxLastName
- textBoxPhone
- textBoxEmail

Наименование (Name) и текстовое отображение (Text) элементов label:

- labelFirstName – Имя
- labelMiddleName – Отчество
- labelLastName – Фамилия
- labelPhone – Телефон
- labelEmail – Почта

Наименование (Name) элемента listView:

- listViewClient

Наименование (Name) и текстовое отображение (Text) элементов button:

- buttonAdd – Создать
- buttonEdit – Изменить
- buttonDel – Удалить

ID	Имя	Фамилия	Отчество	Телефон	Почта

Рисунок 4 – Вид формы «Клиент»

5. После добавления элементов нажимаем на форме на кнопку «Создать» левой кнопкой мыши два раза для открытия окна написания кода (В дальнейшем, после определения всех кнопок можно использовать клавишу F7). (см. рисунок 5):

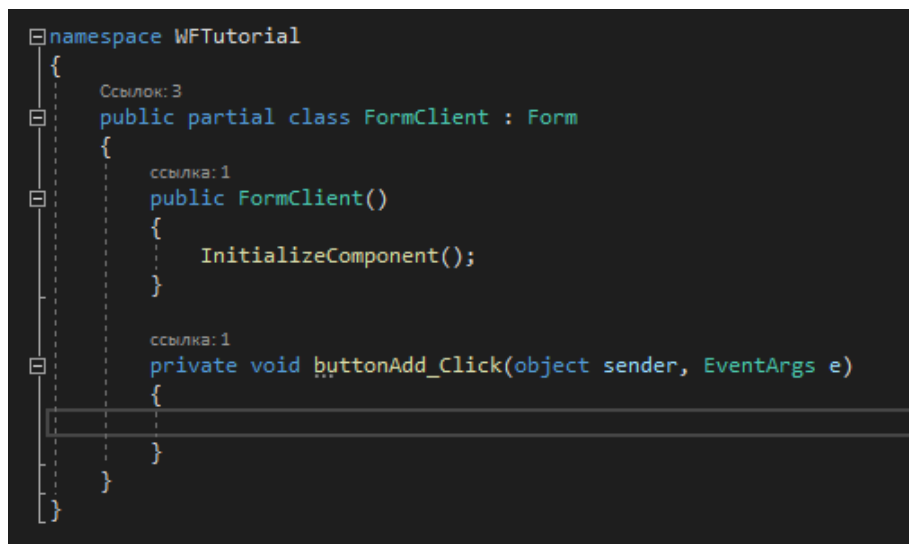


Рисунок 5 – Метод кнопки Add

6. После открытия кнопки двойным нажатием для создаётся пустой метод, в котором пишется дальнейший код.
7. Чтобы добавить клиента, нужно написать следующий код (см. рисунок 6):

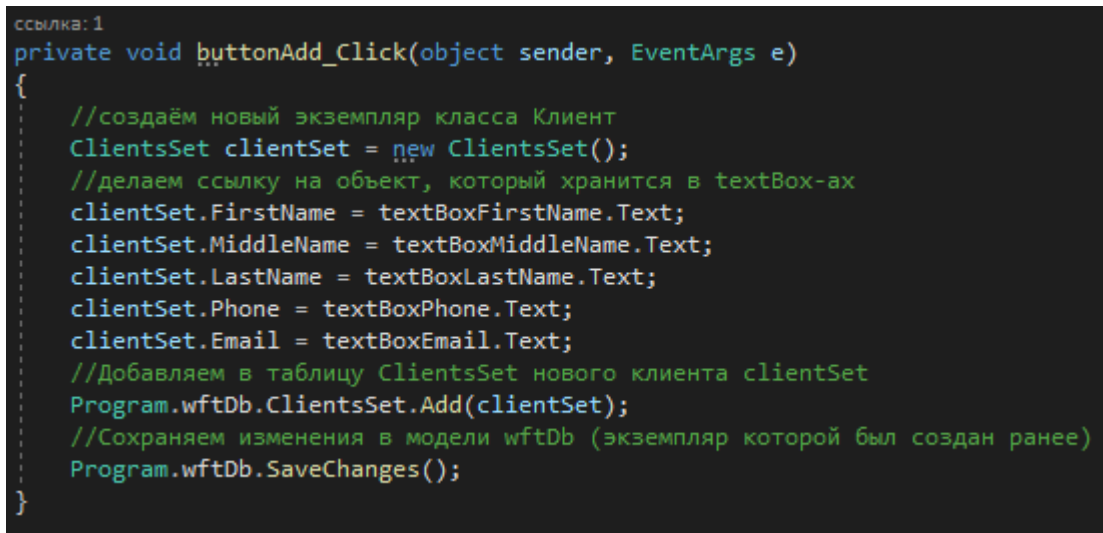


Рисунок 6 – Метод создание клиента

В комментариях к коду были описаны действия для добавления клиента в таблицу базы данных.

8. После написания сохраняем изменения на Ctrl+S и нажмём «Пуск» (см. рисунок 7):

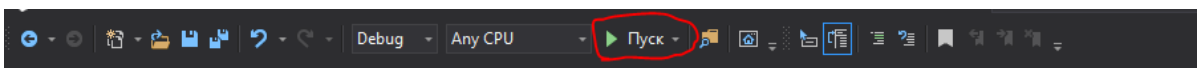


Рисунок 7 – Запуск программы

9. После нажатия, запустится программа, введём данные для проверки на форме (см. рисунок 8):

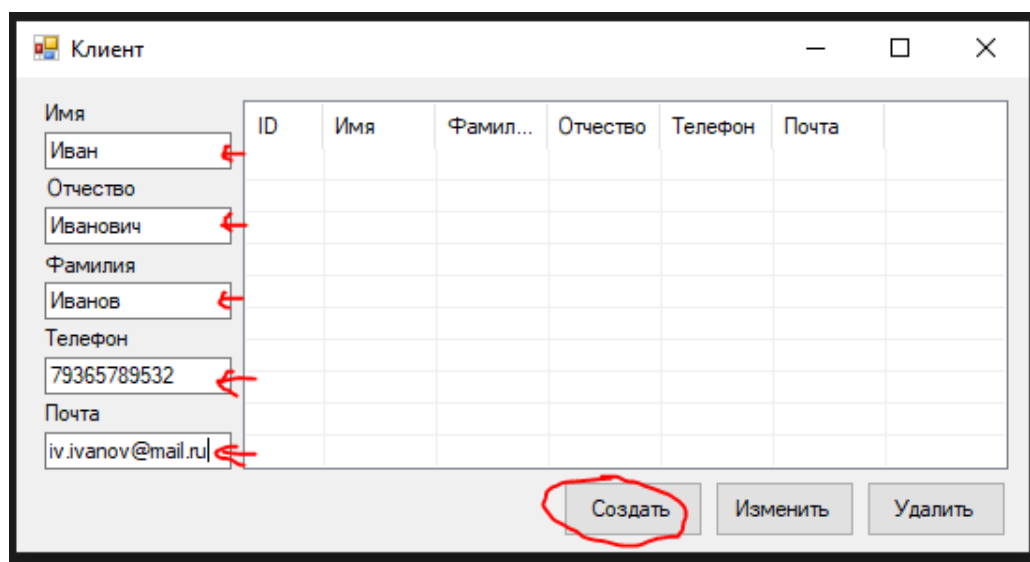


Рисунок 8 – Проверка добавления клиента

10. После чего, нажмём «Создать» и перейдём в SSMS, где проверим, добавилась ли запись в таблицу в базе данных (см. рисунок 9):

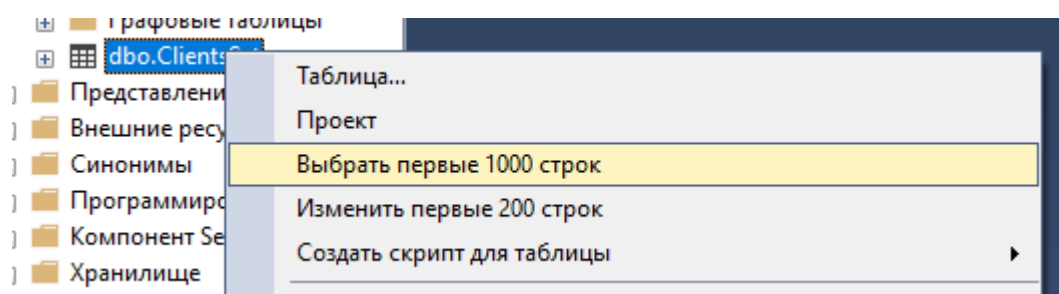


Рисунок 9 – Просмотр строк в таблице

11. Для просмотра записей, нажмем правой кнопкой мыши по таблице и выберем «Выбрать первые 1000 строк» (см. рисунок 10):

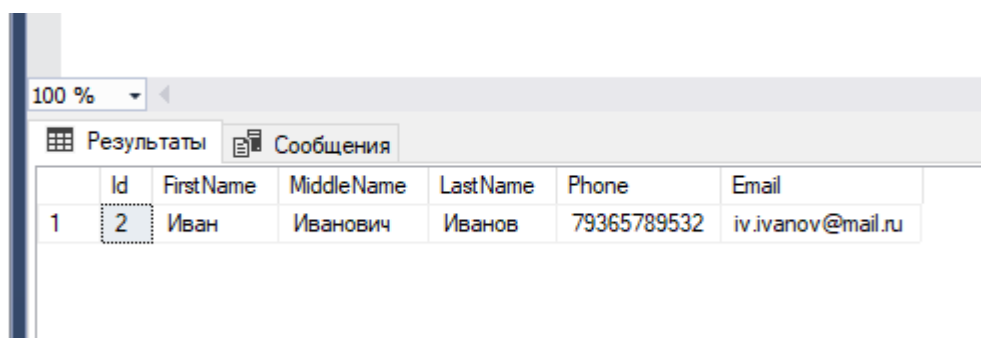


Рисунок 10 – Список записей

12. На рисунке 10 можно увидеть, что наши данные были успешно добавлены в базу данных. Но, на форме в listView при добавлении они не отображались.

13. Для этого нужно написать ещё один метод, переходим обратно в Visual Studio 2019 и нажимаем F7 находясь на форме, чтобы открыть окно написания кода.

14. Для отображения списка клиентов напишем следующий метод (см. рисунок 11):

```

Ссылка: 2
void ShowClient()
{
    //предварительно очищаем listView
    listViewClient.Items.Clear();
    //проходимся по коллекции клиентов, которые находятся в базе с помощью foreach
    foreach (ClientsSet clientsSet in Program.wftDb.ClientsSet)
    {
        //создаем новый элемент в listView
        //для этого создаем новый массив строк
        ListViewItem item = new ListViewItem(new string[]
        {
            //указываем необходимые поля
            clientsSet.Id.ToString(), clientsSet.FirstName, clientsSet.MiddleName,
            clientsSet.LastName, clientsSet.Phone, clientsSet.Email
        });
        //указываем по какому тегу будем брать элементы
        item.Tag = clientsSet;
        //добавляем элементы в listView для отображения
        listViewClient.Items.Add(item);
    }
    //выравниваем колонки в listView
    listViewClient.AutoSizeColumnsMode(ColumnHeaderAutoSizeStyle.HeaderSize);
}

```

Рисунок 11 – Метод отображения клиентов в listView

15. Затем вызовем данный метод в инициализации формы (см. рисунок 48):

```

ссылка: 1
public FormClient()
{
    InitializeComponent();
    ShowClient();
}

ссылка: 1
private void buttonAdd_Click(object sender, EventArgs e)
{
    //создаём новый экземпляр класса Клиент
    ClientsSet clientSet = new ClientsSet();
    //делаем ссылку на объект, который хранится в textBox-ах
    clientSet.FirstName = textBoxFirstName.Text;
    clientSet.MiddleName = textBoxMiddleName.Text;
    clientSet.LastName = textBoxLastName.Text;
    clientSet.Phone = textBoxPhone.Text;
    clientSet.Email = textBoxEmail.Text;
    //Добавляем в таблицу ClientsSet нового клиента clientSet
    Program.wftDb.ClientsSet.Add(clientSet);
    //Сохраняем изменения в модели wftDb (экземпляр которой был создан ранее)
    Program.wftDb.SaveChanges();
    ShowClient();
}

```

Рисунок 12 – Вызов метода отображения

16. Далее запускаем программу клавишей F5 или «Пуск» и проверяем, отобразилась ли наша запись в listView (см. рисунок 13):

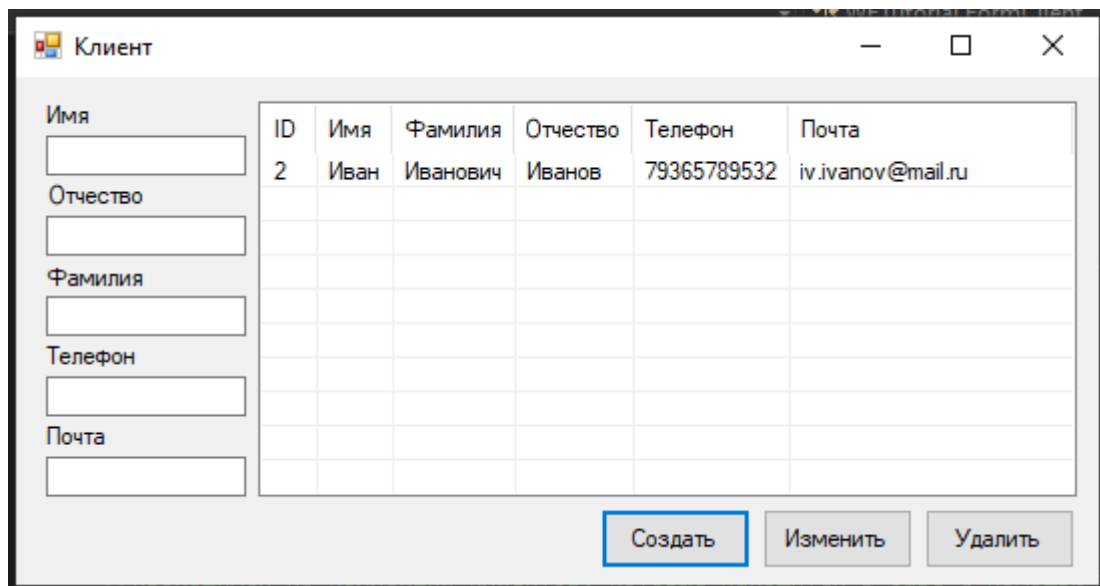


Рисунок 13 – Отображение в listView

17. Как можно увидеть на рисунке 13, теперь список клиентов будет отображаться на форме в listView.
18. Далее перейдём к написанию кода на изменение информации о клиенте. Для этого нажмем двойным щелчком на кнопку «Изменить» на форме (как было описано ранее) и напишем следующий метод (см. рисунок 14):

```

ссылка: 1
private void buttonEdit_Click(object sender, EventArgs e)
{
    //условие, если в listView выбран 1 элемент
    if (listViewClient.SelectedItems.Count == 1)
    {
        //ищем элемент из таблицы по тегу
        ClientsSet clientSet = listViewClient.SelectedItems[0].Tag as ClientsSet;
        //указываем, что может быть изменено
        clientSet.FirstName = textBoxFirstName.Text;
        clientSet.MiddleName = textBoxMiddleName.Text;
        clientSet.LastName = textBoxLastName.Text;
        clientSet.Phone = textBoxPhone.Text;
        clientSet.Email = textBoxEmail.Text;
        //Сохраняем изменения в модели wftDb (экземпляр которой был создан ранее)
        Program.wftDb.SaveChanges();
        //отображение в listView
        ShowClient();
    }
}

```

Рисунок 14 – Метод изменения информации о клиенте

19. Затем напишем ещё один метод, который будет осуществлять отображение выбранного элемента, для этого на форме щелкнем два раза по самому listView и напишем следующее (см. рисунок 15):

```

ССЫЛКА: 1
private void listViewClient_SelectedIndexChanged(object sender, EventArgs e)
{
    //условие, если выбран 1 элемент
    if (listViewClient.SelectedItems.Count == 1)
    {
        //ищем элемент из таблицы по тегу
        ClientsSet clientSet = listViewClient.SelectedItems[0].Tag as ClientsSet;
        //указываем, что может быть изменено
        textBoxFirstName.Text = clientSet.FirstName;
        textBoxMiddleName.Text = clientSet.MiddleName;
        textBoxLastName.Text = clientSet.LastName;
        textBoxPhone.Text = clientSet.Phone;
        textBoxEmail.Text = clientSet.Email;
    }
    else
    {
        //условие, иначе, если не выбран ни один элемент, то задаем пустые поля
        textBoxFirstName.Text = "";
        textBoxMiddleName.Text = "";
        textBoxLastName.Text = "";
        textBoxPhone.Text = "";
        textBoxEmail.Text = "";
    }
}

```

Рисунок 15 – Метод выбора элементов

20. Далее запускаем программу и проверяем, изменим поле «Имя» с «Иван» на «Петр» и нажмем «Изменить» (см. рисунок 16):

ID	Имя	Фамилия	Отчество	Телефон	Почта
2	Петр	Иванович	Иванов	79365789532	iv.ivanov@mail.ru

Рисунок 16 – Проверка метода на изменение

21. Далее напишем метод для удаления клиента из базы данных (см. рисунок 17):

```

ссылка: 1
private void buttonDel_Click(object sender, EventArgs e)
{
    //пробуем совершить действие
    try
    {
        //если выбран 1 элемент из listView
        if (listViewClient.SelectedItems.Count == 1)
        {
            //ищем этот элемент, сверяем его
            ClientsSet clientSet = listViewClient.SelectedItems[0].Tag as ClientsSet;
            //удаляем из модели и базы данных
            Program.wftDb.ClientsSet.Remove(clientSet);
            //сохраняем изменения
            Program.wftDb.SaveChanges();
            //отображаем обновлённый список
            ShowClient();
        }
        //очищаем textBox-ы
        textBoxFirstName.Text = "";
        textBoxMiddleName.Text = "";
        textBoxLastName.Text = "";
        textBoxPhone.Text = "";
        textBoxEmail.Text = "";
    }
    //если возникает какая-то ошибка, к примеру, запись используется, выводим всплывающее сообщение
    catch
    {
        //вызываем метод для всплывающего окна, в котором указываем текст, заголовок, кнопку и иконку
        MessageBox.Show("Невозможно удалить, эта запись используется!", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

Рисунок 17 – Метод удаление клиента

22. Далее снова запускаем программу и проверяем работу написанного метода (см. рисунок 18):

23. Выбираем из списка клиента и нажимаем «Удалить».

Рисунок 18 – Проверка метода на удаление записи

24. На рисунке 18 можно увидеть, что клиент был удалён.

На этом написание кода закончено.

25. Теперь поработаем немного над оформлением:

В свойствах формы изменим некоторые параметры:

- Текст в строке заголовка (Text) – Клиенты.
- Размеры формы (Size) – 700; 500.
- Стартовая позиция (StartPosition) – CenterScreen.

На форму добавим изображение (элемент PictureBox) – логотип компании Esoft.

Цвет, положение, шрифт объектов и самой формы необходимо будет изменить в соответствии с **Руководством по стилю!**