

1. Работа с SSMS

1. Откроем SSMS и подключимся к серверу.
2. Зайдем в свою базу данных WFTutorial и создадим в ней новую таблицу RealEstateSet, которую будем использовать для работы с любым объектом недвижимости. Зададим ключевое поле Id счетчиком (auto_increment).


	Имя столбца	Тип данных	Разрешить ...
	Id	int	<input type="checkbox"/>
	Address_City	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Address_Street	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Address_House	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Address_Number	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Coordinate_latitude	float	<input checked="" type="checkbox"/>
	Coordinate_longitude	float	<input checked="" type="checkbox"/>
	Type	int	<input type="checkbox"/>
	TotalArea	float	<input checked="" type="checkbox"/>
	TotalFloors	int	<input checked="" type="checkbox"/>
	Rooms	int	<input checked="" type="checkbox"/>
	Floor	int	<input checked="" type="checkbox"/>

Рисунок 1 – Таблица RealEstateSet

3. В созданной таблице есть поле Type – тип объекта недвижимости. Мы сделали его int и поэтому каждый тип объекта будет задаваться своим номером: Квартира – 0, Дом – 1, Земля – 2. Это понадобится при работе с элементами формы.

2. Работа с формой «Объекты недвижимости»

4. Переходим в Visual Studio, откроем проект Esoft_Project.
5. Удалим уже созданную модель ModelTutorial.edmx и создадим ее снова (если вы не помните, как это сделать, обратитесь к документу https://vyatsu-my.sharepoint.com/:b:/g/personal/usr11033_vyatsu_ru/EYYIUb5DNWZBvycCVy3hvNwBw0II_BBRRhZJt11_FUqw_jg?e=cNdxB1).
6. Далее создаём форму, на которой будет располагаться интерфейс для объектов недвижимости, для этого перейдём в вкладку «Проект», затем «Добавить форму». Зададим ей название FormRealEstate.

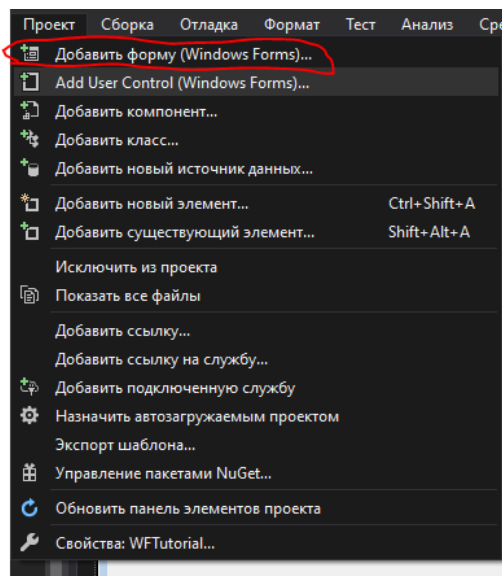


Рисунок 2 – Добавление формы

7. Сделаем ссылку на эту форму с кнопки «Объекты недвижимости»:

- Перейдем в конструктор формы Menu.
- Два раза щелкнем правой кнопкой мыши по кнопке «Объекты недвижимости» и добавим в метод вызова формы «Объекты недвижимости».

```
private void buttonOpenRealEstates_Click(object sender, EventArgs e)
{
    //Задаем новую форму из класса Объекты недвижимости и открываем ее
    Form formRealEstate = new FormRealEstate();
    formRealEstate.Show();
}
```

Рисунок 3 – Открытие формы «Объекты недвижимости»

8. Перейдем снова на форму «Объекты недвижимости». Добавим необходимые элементы из «Панели элементов».

9. В данном интерфейсе нам нужны: textBox (10), label (11), listView (3), button (3), comboBox(1). Перетаскиваем на форму и переименовываем каждый элемент для удобства.

Наименование (Name) элементов textBox:

- textBoxAddress_City
- textBoxAddress_Street
- textBoxAddress_House
- textBoxAddress_Number
- textBoxCoordinate_latitude
- textBoxCoordinate_longitude
- textBoxTotalArea
- textBoxRooms
- textBoxFloor
- textBoxTotalFloors

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

- labelAddress_City – Город
- labelAddress_Street – Улица
- labelAddress_House – Номер дома
- labelAddress_Number – Номер квартиры
- labelCoordinate_latitude – Ширина
- labelCoordinate_longitude – Долгота
- labelTotalArea – Площадь
- labelType – Тип объекта недвижимости
- labelRooms – Количество комнат
- labelFloor – Этаж
- labelTotalFloors – Количество этажей

Наименование (Name) элементов listView:

- listViewRealEstateSet_Apartment
- listViewRealEstateSet_House
- listViewRealEstateSet_Land

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

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

Наименование (Name) элемента comboBox:

- comboBoxType

Настройка элементов listView:

- Измените некоторые свойства объектов:

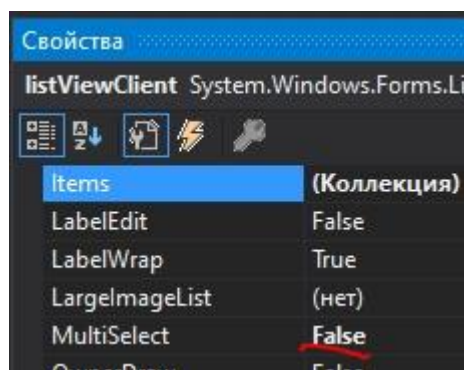


Рисунок 4 – Изменение свойств listView №1

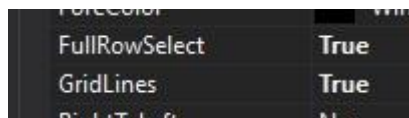


Рисунок 5 – Изменение свойств listView №2

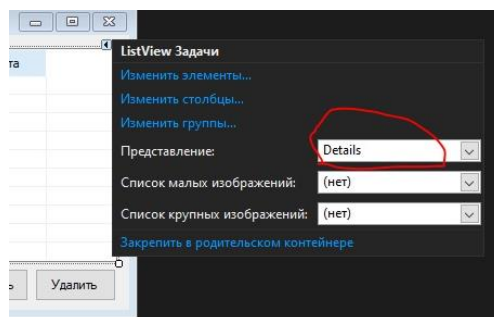


Рисунок 6 – Изменение представлений listView

- Для каждого listView зададим необходимые столбцы:

Город	Улица	Номер дома	Номер квартиры	Широта	Долгота	Площадь	Количество комнат	Этаж
-------	-------	------------	----------------	--------	---------	---------	-------------------	------

Рисунок 7 – Столбцы listViewRealEstateSet_Apartment

Город	Улица	Номер дома	Номер квартиры	Широта	Долгота	Площадь
-------	-------	------------	----------------	--------	---------	---------

Рисунок 7 – Столбцы listViewRealEstateSet_Land

Город	Улица	Номер дома	Номер квартиры	Широта	Долгота	Площадь	Количество этажей
-------	-------	------------	----------------	--------	---------	---------	-------------------

Рисунок 8 – listViewRealEstateSet_House

Настройка элемента comboBoxType:

- Выделим объект и нажмем «Изменить элементы».

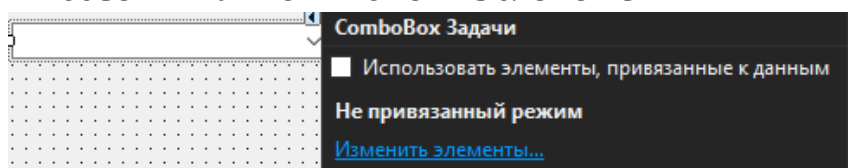


Рисунок 9 – Настройка comboBoxType

- Добавим в коллекцию строк необходимые типы объектов недвижимости: Квартира, Дом, Земля.

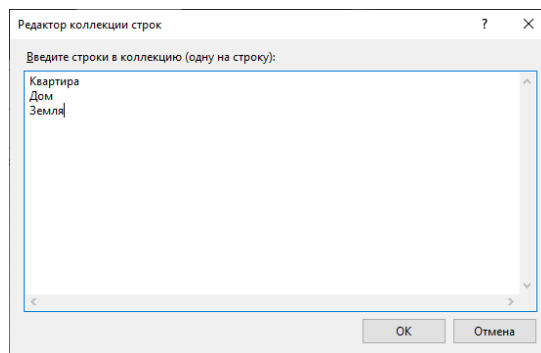


Рисунок 10 – Добавление в коллекцию строк comboBoxType

10. После настройки форма «Объекты недвижимости» должна выглядеть примерно так:

Рисунок 11.1 – Вид формы «Объекты недвижимости»

Рисунок 11.2 – Вид формы «Объекты недвижимости»

11. Главная идея работы с этой формой – в нужное время скрывать или открывать некоторые элементы. Для этого используется свойство объектов Visible.
12. Будем считать, что при первом открытии формы должны открываться Квартиры, поэтому все элементы, не относящиеся к Типу «Квартира» (а это: textBoxTotalFloors, labelTotalFloors, listViewRealEstateSet_House, listViewRealEstateSet_Land) сделаем невидимыми. Для этого у каждого из указанных элементов в свойствах найдем пункт Visible и значение поменяем на false:

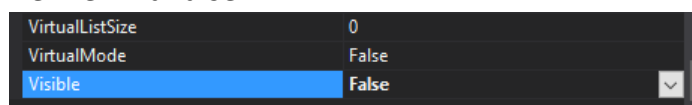


Рисунок 12 – Изменение свойства Visible

13. Так как начинаем мы с квартир, то при загрузке формы comboBoxType должен показывать строчку «Квартира» (в элементе comboBoxType это строка с индексом 0). Учтем это и добавим код в загрузку:

```
public FormRealEstate()
{
    InitializeComponent();
    comboBoxType.SelectedIndex = 0;
}
```

Рисунок 13 – Изменение начальных данных comboBoxType

14. При изменении строки в comboBoxType должны меняться соответствующие элементы. Для этого мы будем использовать метод comboBoxType_SelectedIndexChanged (его можно открыть, щелкнув дважды по comboBox).

```
private void comboBoxType_SelectedIndexChanged(object sender, EventArgs e)
{
}
```

Рисунок 14 – Метод comboBoxType_SelectedIndexChanged

15. Добавим информацию по строке Квартира (в элементе comboBoxType это строка с индексом 0). Часть элементов мы должны сделать видимыми, часть невидимыми.

```
private void comboBoxType_SelectedIndexChanged(object sender, EventArgs e)
{
    //Изменения формы, если выбрана строчка "Квартира" (ее индекс 0)
    if (comboBoxType.SelectedIndex == 0)
    {
        //Делаем видимыми нужные элементы
        listViewRealEstateSet_Apartment.Visible = true;
        labelFloor.Visible = true;
        textBoxFloor.Visible = true;
        labelRooms.Visible = true;
        textBoxRooms.Visible = true;

        //Скрываем ненужные элементы
        listViewRealEstateSet_House.Visible = false;
        listViewRealEstateSet_Land.Visible = false;
        labelTotalFloors.Visible = false;
        textBoxTotalFloors.Visible = false;

        //Очищаем все видимые элементы
        textBoxAddress_City.Text = "";
        textBoxAddress_House.Text = "";
        textBoxAddress_Street.Text = "";
        textBoxAddress_Number.Text = "";
        textBoxCoordinate_latitude.Text = "";
        textBoxCoordinate_longitude.Text = "";
        textBoxTotalArea.Text = "";
        textBoxRooms.Text = "";
        textBoxFloor.Text = "";
    }
}
```

Рисунок 15 – Видимость элементов для строки «Квартира»

16. Точно также делаем для строк «Дом» (у него индекс 1), «Земля» (индекс 2). В этом же методе продолжим:

```
//Изменения формы, если выбрана строчка "Дом" (ее индекс 1)
else if (comboBoxType.SelectedIndex == 1)
{
    //Делаем видимыми нужные элементы
    listViewRealEstateSet_House.Visible = true;
    labelTotalFloors.Visible = true;
    textBoxTotalFloors.Visible = true;

    //Скрываем ненужные элементы
    listViewRealEstateSet_Land.Visible = false;
    listViewRealEstateSet_Apartment.Visible = false;
    labelFloor.Visible = false;
    textBoxFloor.Visible = false;
    labelRooms.Visible = false;
    textBoxRooms.Visible = false;

    //Очищаем все видимые элементы
    textBoxAddress_City.Text = "";
    textBoxAddress_House.Text = "";
    textBoxAddress_Street.Text = "";
    textBoxAddress_Number.Text = "";
    textBoxCoordinate_latitude.Text = "";
    textBoxCoordinate_longitude.Text = "";
    textBoxTotalArea.Text = "";
    textBoxTotalFloors.Text = "";
}
}
```

Рисунок 16 – Видимость элементов для строки «Дом»

```
//Изменения формы, если выбрана строчка "Земля" (ее индекс 2)
else if (comboBoxType.SelectedIndex == 2)
{
    //Делаем видимыми нужные элементы
    listViewRealEstateSet_Land.Visible = true;

    //Скрываем ненужные элементы
    listViewRealEstateSet_House.Visible = false;
    listViewRealEstateSet_Apartment.Visible = false;
    labelFloor.Visible = false;
    textBoxFloor.Visible = false;
    labelRooms.Visible = false;
    textBoxRooms.Visible = false;
    labelTotalFloors.Visible = false;
    textBoxTotalFloors.Visible = false;

    //Очищаем все видимые элементы
    textBoxAddress_City.Text = "";
    textBoxAddress_House.Text = "";
    textBoxAddress_Street.Text = "";
    textBoxAddress_Number.Text = "";
    textBoxCoordinate_latitude.Text = "";
    textBoxCoordinate_longitude.Text = "";
    textBoxTotalArea.Text = "";
}
}
```

Рисунок 16 – Видимость элементов для строки «Земля»

17. Запустите программу и протестируйте изменения. Если часть элементов не скрывается или не появляется, обратитесь снова к коду и проверьте его правильность.
18. После настройки всех элементов нажимаем на форме на кнопку «Создать» левой кнопкой мыши два раза для открытия окна написания кода

```
private void buttonAdd_Click(object sender, EventArgs e)
{
}
}
```

Рисунок 17 – Метод кнопки Add

19. Чтобы добавить объект недвижимости, нужно учитывать, какой тип объекта недвижимости выбран (то есть, какая строка выделена в comboBoxType):

```
private void buttonAdd_Click(object sender, EventArgs e)
{
    //Создаем новый экземпляр класса Объект недвижимости
    RealEstateSet realEstate = new RealEstateSet();
    //Делаем ссылку на объект, который хранится в textBox-ах (сначала общие поля)
    realEstate.Address_City = textBoxAddress_City.Text;
    realEstate.Address_House = textBoxAddress_House.Text;
    realEstate.Address_Street = textBoxAddress_Street.Text;
    realEstate.Address_Number = textBoxAddress_Number.Text;
    realEstate.Coordinate_latitude = Convert.ToDouble(textBoxCoordinate_latitude.Text);
    realEstate.Coordinate_longitude = Convert.ToDouble(textBoxCoordinate_longitude.Text);
    realEstate.TotalArea = Convert.ToDouble(textBoxTotalArea.Text);
    //Дополнительные поля для типа "Квартира"
    if (comboBoxType.SelectedIndex == 0)
    {
        realEstate.Type = 0;
        realEstate.Rooms = Convert.ToInt32(textBoxRooms.Text);
        realEstate.Floor = Convert.ToInt32(textBoxFloor.Text);
    }
    //Дополнительные поля для типа "Дом"
    else if (comboBoxType.SelectedIndex == 1)
    {
        realEstate.Type = 1;
        realEstate.TotalFloors = Convert.ToInt32(textBoxTotalFloors.Text);
    }
    //Дополнительные поля для типа "Земля"
    else
    {
        realEstate.Type = 2;
    }
    //Добавляем в таблицу RealEstateSet новый объект недвижимости realEstate
    Program.wftDb.RealEstateSet.Add(realEstate);
    //Сохраняем изменения в модели wftDb
    Program.wftDb.SaveChanges();
}
```

Рисунок 18 – Метод создания объекта недвижимости

20. Протестируйте программу. Проверьте в SSMS, добавились ли объекты недвижимости в базу данных.

21. Для отображения списков объектов недвижимости напомним следующий общий метод:

```
void ShowRealEstateSet()
{
    //Предварительно очищаем все listView
    listViewRealEstateSet_Apartment.Items.Clear();
    listViewRealEstateSet_House.Items.Clear();
    listViewRealEstateSet_Land.Items.Clear();
    //Проходим по коллекции клиентов, которые находятся в базе с помощью foreach
    foreach (RealEstateSet realEstate in Program.wftDb.RealEstateSet)
    {
        //отображение квартир в listViewRealEstateSet_Apartment
        if (realEstate.Type == 0)
        {
            //создадим новый элемент в listViewRealEstateSet_Apartment с помощью массива строк
            ListViewItem item = new ListViewItem(new string[]
            {
                //указываем необходимые поля
                realEstate.Address_City, realEstate.Address_Street, realEstate.Address_House,
                realEstate.Address_Number, realEstate.Coordinate_latitude.ToString(),
                realEstate.Coordinate_longitude.ToString(), realEstate.TotalArea.ToString(),
                realEstate.Rooms.ToString(), realEstate.Floor.ToString()
            });
            //указываем по какому тегу выбраны элементы
            item.Tag = realEstate;
            //добавляем элементы в listViewRealEstateSet_Apartment для отображения
            listViewRealEstateSet_Apartment.Items.Add(item);
        }
        //отображение домов в listViewRealEstateSet_House
        else if (realEstate.Type == 1)
        {
            //создадим новый элемент в listViewRealEstateSet_House с помощью массива строк
            ListViewItem item = new ListViewItem(new string[]
            {
                //указываем необходимые поля
                realEstate.Address_City, realEstate.Address_Street, realEstate.Address_House,
                realEstate.Address_Number, realEstate.Coordinate_latitude.ToString(),
                realEstate.Coordinate_longitude.ToString(), realEstate.TotalArea.ToString(),
                realEstate.TotalFloors.ToString()
            });
            //указываем по какому тегу выбраны элементы
            item.Tag = realEstate;
            //добавляем элементы в listViewRealEstateSet_House для отображения
            listViewRealEstateSet_House.Items.Add(item);
        }
    }
}
```

Рисунок 19 – Метод отображения клиентов в listView (начало)

```
    }
    //выравниваем столбцы во всех listView
    listViewRealEstateSet_Apartment.AutoSizeColumns(ColumnHeaderAutoResizeStyle.HeaderSize);
    listViewRealEstateSet_House.AutoSizeColumns(ColumnHeaderAutoResizeStyle.HeaderSize);
    listViewRealEstateSet_Land.AutoSizeColumns(ColumnHeaderAutoResizeStyle.HeaderSize);
}
```

Рисунок 20 – Метод отображения клиентов в listView (конец)

22. Затем вызовем данный метод в инициализации формы и в методе buttonAdd_Click:

```
public FormRealEstate()
{
    InitializeComponent();
    comboBoxType.SelectedIndex = 0;
    ShowRealEstateSet();
}
```

Рисунок 21 – Вызов метода отображения в инициализации формы

```
Program.wftDb.RealEstateSet.Add(realEstate);
//Сохраняем изменения в модели wftDb
Program.wftDb.SaveChanges();
ShowRealEstateSet();
}
```

Рисунок 22 – Вызов метода отображения в методе buttonAdd_Click

23. Протестируйте изменения. Проверьте правильность выполнения команд.

24. Далее перейдём к написанию кода на изменение информации об объекте недвижимости. Для этого нажмем двойным щелчком на кнопку «Изменить» и напишем следующий метод.

```
private void buttonEdit_Click(object sender, EventArgs e)
{
    //выбран тип "Квартира", работа с listViewRealEstateSet_Apartment
    if (comboBoxType.SelectedIndex == 0)
    {
        //если в listView выбран элемент
        if (listViewRealEstateSet_Apartment.SelectedItems.Count == 1)
        {
            //Ищем элемент из таблицы по тегу
            RealEstateSet realEstate = listViewRealEstateSet_Apartment.SelectedItems[0].Tag as RealEstateSet;
            //указываем, что может быть изменено
            realEstate.Address_City = textBoxAddress_City.Text;
            realEstate.Address_House = textBoxAddress_House.Text;
            realEstate.Address_Street = textBoxAddress_Street.Text;
            realEstate.Address_Number = textBoxAddress_Number.Text;
            realEstate.Coordinate_latitude = Convert.ToDouble(textBoxCoordinate_latitude.Text);
            realEstate.Coordinate_longitude = Convert.ToDouble(textBoxCoordinate_longitude.Text);
            realEstate.TotalArea = Convert.ToDouble(textBoxTotalArea.Text);
            realEstate.Rooms = Convert.ToInt32(textBoxRooms.Text);
            realEstate.Floor = Convert.ToInt32(textBoxFloor.Text);
            //сохраняем изменения в модели wftDb
            Program.wftDb.SaveChanges();
            //отображаем в listViewRealEstateSet_Apartment
            ShowRealEstateSet();
        }
    }
}
```

Рисунок 23 – Метод изменения информации об объекте недвижимости (начало)

```

//выбран тип "Дом", работа с listViewRealEstateSet_House
else if (comboBoxType.SelectedIndex == 1)
{
    //если в listView выбран элемент
    if (listViewRealEstateSet_House.SelectedItems.Count == 1)
    {
        //Ищем элемент из таблицы по тегу
        RealEstateSet realEstate = listViewRealEstateSet_House.SelectedItems[0].Tag as RealEstateSet;
        //указываем, что может быть изменено
        realEstate.Address_City = textBoxAddress_City.Text;
        realEstate.Address_House = textBoxAddress_House.Text;
        realEstate.Address_Street = textBoxAddress_Street.Text;
        realEstate.Address_Number = textBoxAddress_Number.Text;
        realEstate.Coordinate_latitude = Convert.ToDouble(textBoxCoordinate_latitude.Text);
        realEstate.Coordinate_longitude = Convert.ToDouble(textBoxCoordinate_longitude.Text);
        realEstate.TotalArea = Convert.ToDouble(textBoxTotalArea.Text);
        realEstate.TotalFloors = Convert.ToInt32(textBoxTotalFloors.Text);
        //сохраняем изменения в модели wftDb
        Program.wftDb.SaveChanges();
        //отображаем в listViewRealEstateSet_House
        ShowRealEstateSet();
    }
}
}

```

Рисунок 24 – Метод изменения информации об объекте недвижимости (середина)

```

//выбран тип "Земля", работа с listViewRealEstateSet_Land
else
{
    //если в listView выбран элемент
    if (listViewRealEstateSet_Land.SelectedItems.Count == 1)
    {
        //Ищем элемент из таблицы по тегу
        RealEstateSet realEstate = listViewRealEstateSet_Land.SelectedItems[0].Tag as RealEstateSet;
        //указываем, что может быть изменено
        realEstate.Address_City = textBoxAddress_City.Text;
        realEstate.Address_House = textBoxAddress_House.Text;
        realEstate.Address_Street = textBoxAddress_Street.Text;
        realEstate.Address_Number = textBoxAddress_Number.Text;
        realEstate.Coordinate_latitude = Convert.ToDouble(textBoxCoordinate_latitude.Text);
        realEstate.Coordinate_longitude = Convert.ToDouble(textBoxCoordinate_longitude.Text);
        realEstate.TotalArea = Convert.ToDouble(textBoxTotalArea.Text);
        //сохраняем изменения в модели wftDb
        Program.wftDb.SaveChanges();
        //отображаем в listViewRealEstateSet_House
        ShowRealEstateSet();
    }
}
}

```

Рисунок 25 – Метод изменения информации об объекте недвижимости (конец)

25. Затем напишем ещё один метод, который будет осуществлять отображение выбранного элемента, для этого на форме щелкнем два раза по listViewRealEstateSet_Apartment и напишем следующее:

```

private void listViewRealEstateSet_Apartment_SelectedIndexChanged(object sender, EventArgs e)
{
    //если выбран один элемент
    if (listViewRealEstateSet_Apartment.SelectedItems.Count == 1)
    {
        //ищем элемент из таблицы по тегу
        RealEstateSet realEstate = listViewRealEstateSet_Apartment.SelectedItems[0].Tag as RealEstateSet;
        //указываем, что может быть изменено
        textBoxAddress_City.Text = realEstate.Address_City;
        textBoxAddress_Street.Text = realEstate.Address_Street;
        textBoxAddress_House.Text = realEstate.Address_House;
        textBoxAddress_Number.Text = realEstate.Address_Number;
        textBoxCoordinate_latitude.Text = realEstate.Coordinate_latitude.ToString();
        textBoxCoordinate_longitude.Text = realEstate.Coordinate_longitude.ToString();
        textBoxTotalArea.Text = realEstate.TotalArea.ToString();
        textBoxRooms.Text = realEstate.Rooms.ToString();
        textBoxFloor.Text = realEstate.Floor.ToString();
    }
    else
    {
        //если не выбран ни один элемент, задаем пустые поля
        textBoxAddress_City.Text = "";
        textBoxAddress_House.Text = "";
        textBoxAddress_Street.Text = "";
        textBoxAddress_Number.Text = "";
        textBoxCoordinate_latitude.Text = "";
        textBoxCoordinate_longitude.Text = "";
        textBoxTotalArea.Text = "";
        textBoxRooms.Text = "";
        textBoxFloor.Text = "";
    }
}

```

Рисунок 26 – Метод выбора элементов listViewRealEstateSet_Apartment

26. Аналогично сделаем для остальных listView:

```

private void listViewRealEstateSet_House_SelectedIndexChanged(object sender, EventArgs e)
{
    //если выбран один элемент
    if (listViewRealEstateSet_House.SelectedItems.Count == 1)
    {
        //ищем элемент из таблицы по тегу
        RealEstateSet realEstate = listViewRealEstateSet_House.SelectedItems[0].Tag as RealEstateSet;
        //указываем, что может быть изменено
        textBoxAddress_City.Text = realEstate.Address_City;
        textBoxAddress_Street.Text = realEstate.Address_Street;
        textBoxAddress_House.Text = realEstate.Address_House;
        textBoxAddress_Number.Text = realEstate.Address_Number;
        textBoxCoordinate_latitude.Text = realEstate.Coordinate_latitude.ToString();
        textBoxCoordinate_longitude.Text = realEstate.Coordinate_longitude.ToString();
        textBoxTotalArea.Text = realEstate.TotalArea.ToString();
        textBoxTotalFloors.Text = realEstate.TotalFloors.ToString();
    }
    else
    {
        //если не выбран ни один элемент, задаем пустые поля
        textBoxAddress_City.Text = "";
        textBoxAddress_House.Text = "";
        textBoxAddress_Street.Text = "";
        textBoxAddress_Number.Text = "";
        textBoxCoordinate_latitude.Text = "";
        textBoxCoordinate_longitude.Text = "";
        textBoxTotalArea.Text = "";
        textBoxTotalFloors.Text = "";
    }
}

```

Рисунок 27 – Метод выбора элементов listViewRealEstateSet_House

```

private void listViewRealEstateSet_Land_SelectedIndexChanged(object sender, EventArgs e)
{
    //если выбран один элемент
    if (listViewRealEstateSet_Land.SelectedItems.Count == 1)
    {
        //ищем элемент из таблицы по тегу
        RealEstateSet realEstate = listViewRealEstateSet_Land.SelectedItems[0].Tag as RealEstateSet;
        //указываем, что может быть изменено
        textBoxAddress_City.Text = realEstate.Address_City;
        textBoxAddress_Street.Text = realEstate.Address_Street;
        textBoxAddress_House.Text = realEstate.Address_House;
        textBoxAddress_Number.Text = realEstate.Address_Number;
        textBoxCoordinate_latitude.Text = realEstate.Coordinate_latitude.ToString();
        textBoxCoordinate_longitude.Text = realEstate.Coordinate_longitude.ToString();
        textBoxTotalArea.Text = realEstate.TotalArea.ToString();
    }
    else
    {
        //если не выбран ни один элемент, задаем пустые поля
        textBoxAddress_City.Text = "";
        textBoxAddress_House.Text = "";
        textBoxAddress_Street.Text = "";
        textBoxAddress_Number.Text = "";
        textBoxCoordinate_latitude.Text = "";
        textBoxCoordinate_longitude.Text = "";
        textBoxTotalArea.Text = "";
    }
}

```

Рисунок 28 – Метод выбора элементов listViewRealEstateSet_Land

27. Протестируйте изменения. Проверьте правильность выполнения команд.
28. Далее напишем метод для удаления объекта недвижимости из базы данных:

```

private void buttonDel_Click(object sender, EventArgs e)
{
    //попробуем совершить действие
    try
    {
        //выбран тип "Квартира", работа с listViewRealEstateSet_Apartment
        if (comboBoxType.SelectedIndex == 0)
        {
            //если в listView выбран элемент
            if (listViewRealEstateSet_Apartment.SelectedItems.Count == 1)
            {
                //ищем этот элемент базе по тегу
                RealEstateSet realEstate = listViewRealEstateSet_Apartment.SelectedItems[0].Tag as RealEstateSet;
                //удаляем из модели базы данных
                Program.wftDb.RealEstateSet.Remove(realEstate);
                //сохраняем изменения
                Program.wftDb.SaveChanges();
                //отображаем обновленный список
                ShowRealEstateSet();
            }
            //очищаем текстовые поля
            textBoxAddress_City.Text = "";
            textBoxAddress_House.Text = "";
            textBoxAddress_Street.Text = "";
            textBoxAddress_Number.Text = "";
            textBoxCoordinate_latitude.Text = "";
            textBoxCoordinate_longitude.Text = "";
            textBoxTotalArea.Text = "";
            textBoxRooms.Text = "";
            textBoxFloor.Text = "";
        }
    }
}

```

Рисунок 29 – Метод удаления объекта недвижимости (начало)

```

else if (comboBoxType.SelectedIndex == 1)
{
    //если в listView выбран элемент
    if (listViewRealEstateSet_House.SelectedItems.Count == 1)
    {
        //ищем этот элемент базе по тегу
        RealEstateSet realEstate = listViewRealEstateSet_House.SelectedItems[0].Tag as RealEstateSet;
        //удаляем из модели базы данных
        Program.wftDb.RealEstateSet.Remove(realEstate);
        //сохраняем изменения
        Program.wftDb.SaveChanges();
        //отображаем обновленный список
        ShowRealEstateSet();
    }
    //очищаем текстовые поля
    textBoxAddress_City.Text = "";
    textBoxAddress_House.Text = "";
    textBoxAddress_Street.Text = "";
    textBoxAddress_Number.Text = "";
    textBoxCoordinate_latitude.Text = "";
    textBoxCoordinate_longitude.Text = "";
    textBoxTotalArea.Text = "";
    textBoxTotalFloors.Text = "";
}

```

Рисунок 30 – Метод удаления объекта недвижимости (середина)

```

//выбран тип "Земля", работа с listViewRealEstateSet_Land
else
{
    //если в listView выбран элемент
    if (listViewRealEstateSet_Land.SelectedItems.Count == 1)
    {
        //ищем этот элемент базе по тегу
        RealEstateSet realEstate = listViewRealEstateSet_Land.SelectedItems[0].Tag as RealEstateSet;
        //удаляем из модели базы данных
        Program.wftDb.RealEstateSet.Remove(realEstate);
        //сохраняем изменения
        Program.wftDb.SaveChanges();
        //отображаем обновленный список
        ShowRealEstateSet();
    }
    //очищаем текстовые поля
    textBoxAddress_City.Text = "";
    textBoxAddress_House.Text = "";
    textBoxAddress_Street.Text = "";
    textBoxAddress_Number.Text = "";
    textBoxCoordinate_latitude.Text = "";
    textBoxCoordinate_longitude.Text = "";
}
}
//если возникает какая-то ошибка
catch
{
    MessageBox.Show("Невозможно удалить, эта запись используется", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

```

Рисунок 31 – Метод удаления объекта недвижимости (конец)

29. На этом написание кода закончено.
30. Протестируйте изменения. Проверьте правильность выполнения команд.
31. Увеличьте высоту всех listView и наложите их друг на друга.
32. Также, наложите некоторые элементы формы, чтобы при переключении режимов в comboBoxType не было разрывов и пустых мест на форме.
33. Теперь поработаем немного над оформлением:
В свойствах формы изменим некоторые параметры:
 - Текст в строке заголовка (Text) – Объекты недвижимости.
 - Стартовая позиция (StartPosition) – CenterScreen.
34. 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!**