**Лабораторная работа**

**Фильтрация и сортировка табличных данных**

**Рассмотрим задание.**

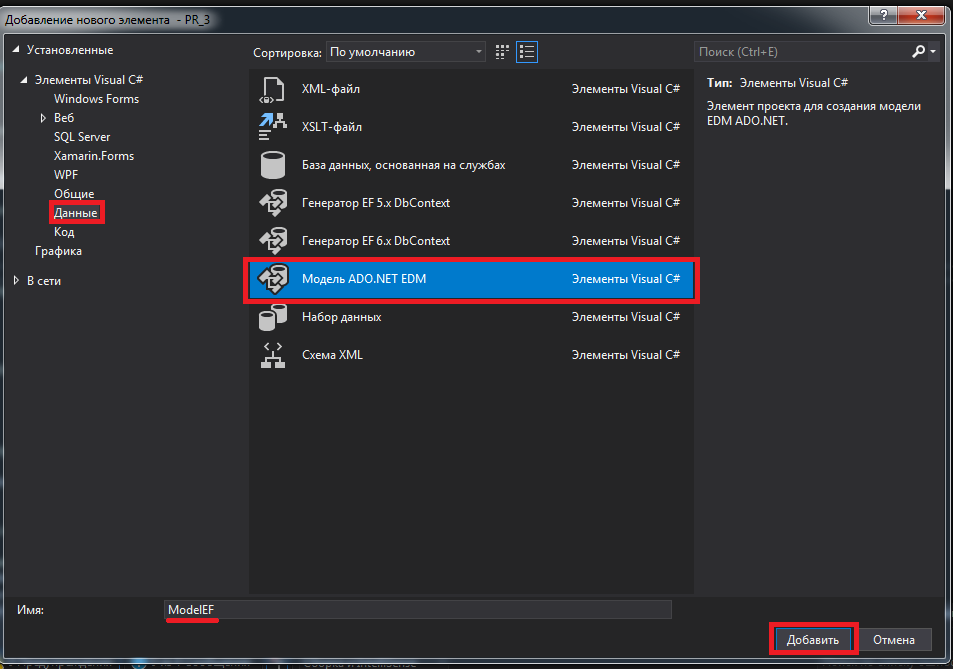
|  |
| --- |
| **Требуется**: создать программу, в которой будет модель работы с БД User\_Data и будет включать 2 таблицы Users, Roles. Осуществить авторизацию пользователей и вход на определённую форму в зависимости от роли. |

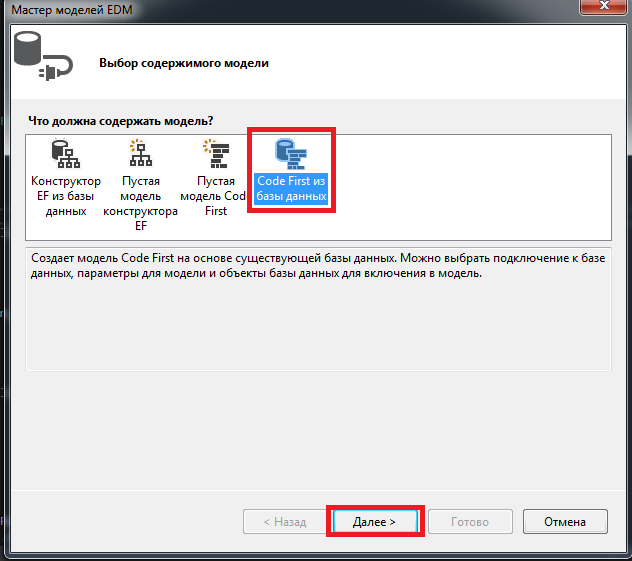
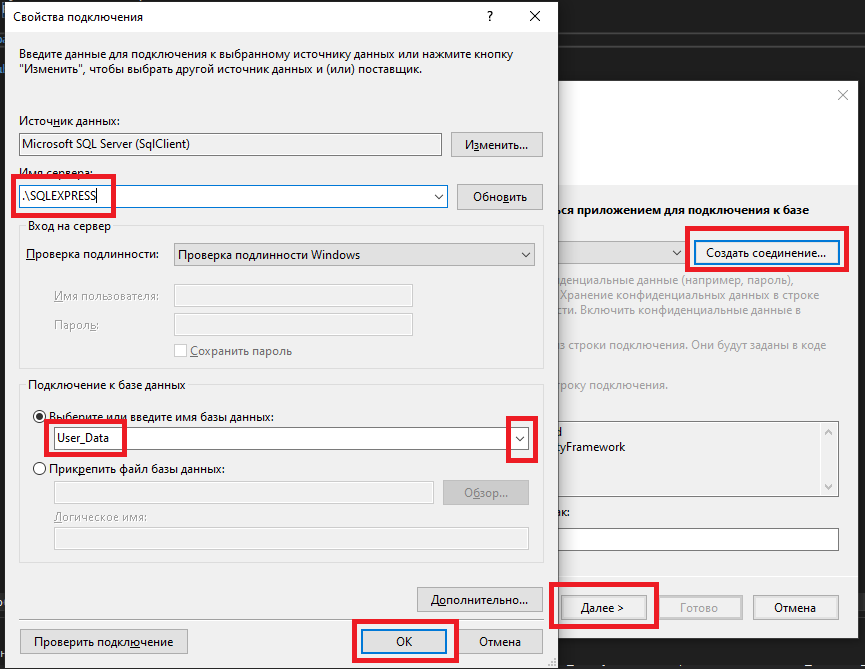
## 1) Создаем в MS SQL базу данных User\_Data, с помощью скрипта (User\_Data.sql)

## 2) Создаем в Visual Studio проект по шаблону «Приложение Windows Forms (.Net Framework)».

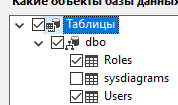
## 3) Добавляем в созданный проект модель EF

Для начала ***создадим новую папку*** (**DBContext)** для правильной архитектуры проекта. В ***новой папке*** создаём (добавляем) **Модель ADO .NET EMD**

Зададим имя модели Model1. Создадим соединение с БД User\_Data. И включим в модель таблицы Users, Roles.

Добавим в эту папку модель EF. Нажимаем правой кнопкой мыши на папку **«Добавить» 🡪»Создать Элемент» 🡪»Данные» 🡪 «Модель ADO .NET EMD» 🡪 «CodeFirst из базы данных» 🡪»Создать соединение» 🡪**

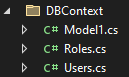
## В открывшемся окне в строку имя сервера ставим .\SQLEXPRESS А в строке базы данных выберите User\_Data 🡪 «OK»



**Далее выберем все пункты 🡪 «Готово»**

В результате к проекту добавятся семь файлов:

* класс контекста Model1.cs
* класс сущностей Roles.cs таблицы Mall.
* класс сущностей Users.cs таблицы Pavilion.

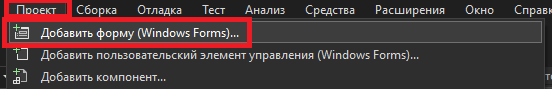
**Теперь нужно ОБЯЗАТЕЛЬНО выполнить хотя бы одну успешную сборку проекта!!! (Ctrl+Shift+B)**

## 4) Добавляем новых форм

## Переименуйте стартовую Form1 в MainForm и добавьте дополнительно 1 новую. форму FormAddUsers.

## 

## Добавить новую форму можно через вкладку Проект 🡪 и нажимаем «Добавить форму».



## 5)Добавление элементов на формы

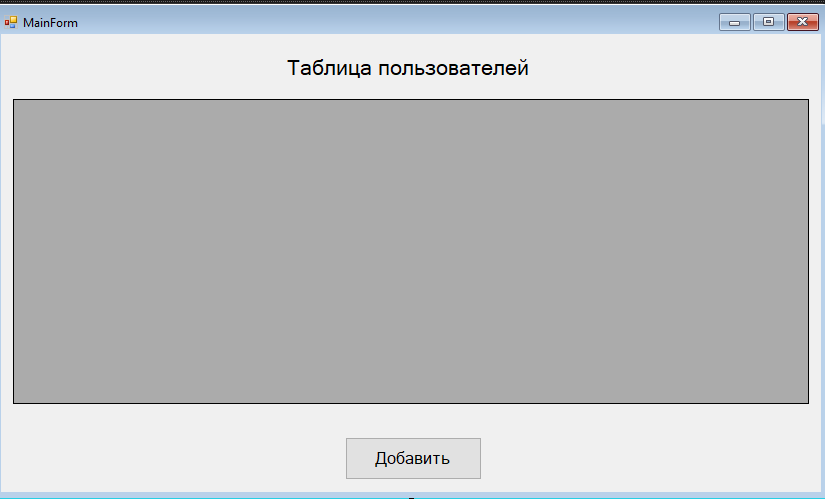
После чего добавим элементов на форму **MainForm**:

Один **DataGridView**

Один **Button (*buttonAdd*)**

Один **Label**

И соответственно поменять этим элементам свойство Name (Имя для кода) и Text (Имя которое видно на интерфейсе) как показано на картинке ниже.



ButtonAdd

Label

DataGridView

Дизайн второй формы (**FormAddUsers**) долен выглядеть ка на картинке ниже и иметь следующие элементы:

Один **MaskedTextBox** (***PhonemaskedTextBox***);

Один **ComboBox** (***roleIDComboBox***);

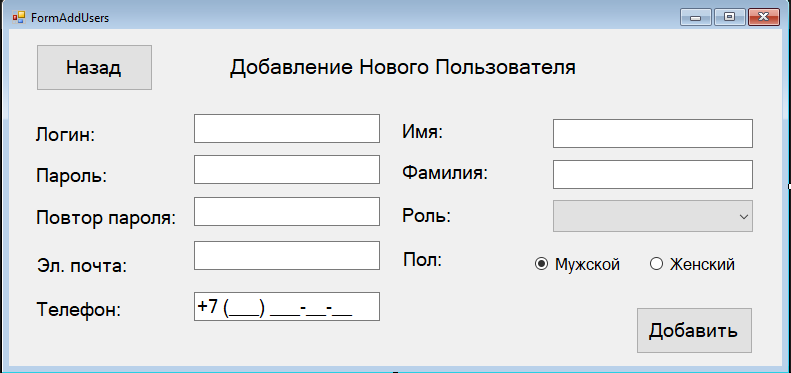
Два **Button (*buttonAdd, buttonBack*);**

Два **RadioButton** (***radioButtonMen, radioButtonWoman***);

Шесть **TextBox (*loginTextBox, passwordTextBox, passwordTextBox2,   
emailTextBox, first\_NameTextBox, second\_NameTextBox*);**

Десять **Label;**

И соответственно поменять этим элементам свойство ***Name*** (Имя для кода) и ***Text*** (Имя которое видно на интерфейсе) как показано на картинке ниже.



ButtonBack

Label

radioButtonMen(Woman)

ButtonAdd

roleIDComboBox

second\_NameTextBox

first\_NameTextBox

PhonemaskedTextBox

emailTextBox

passwordTextBox2

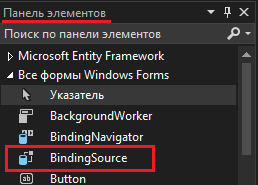
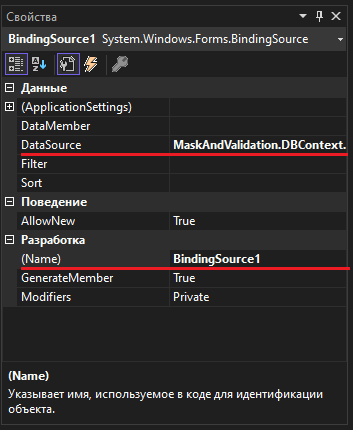
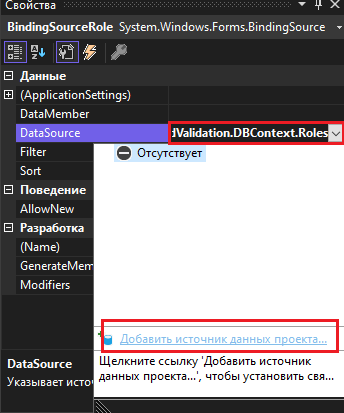
passwordTextBox

loginTextBox

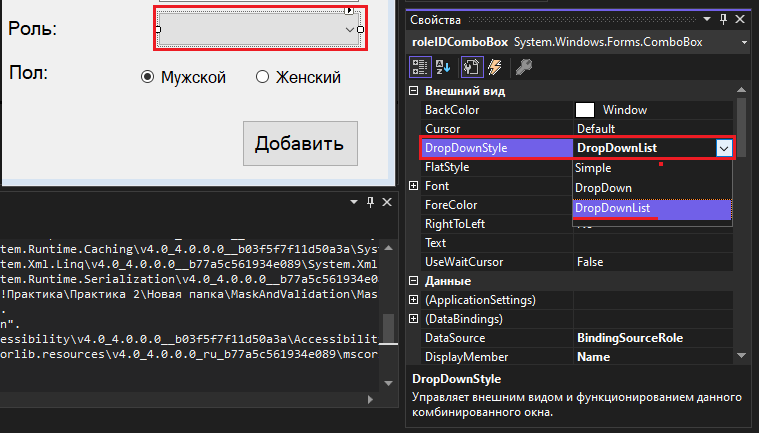
## 5)Работа с элементами управления на формы

На форму необходимо переместить элемент **BindingSource** и поменять свойство ***Name*** на ***BindingSourceRole.***

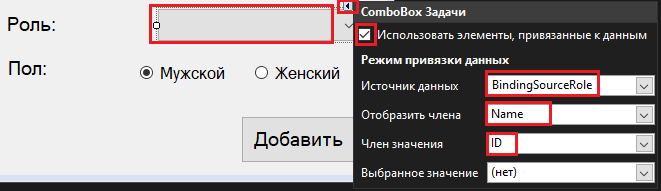
Также у ***BindingSourceRole*** необходимо поменять свойство ***DataSourse.*** «Добавить источник данных проекта». **🡪**



## После у вас открывается окно для добавления нового источника данных. Выбираем 🡪 «Объект» 🡪 «Далее» 🡪После выбираем класс Role в зависимости где вы его создавали в папке или нет (У меня он находится в папке, поэтому нужно раскрыть объект с названием проекта, потом с названием папки (DBContext) и после выбрать класс Role). 🡪 «Готово».

Далее перейдём к редактированию элемента **roleIDComboBox.** Измените свойство *DropDownStyle* на ***DropDownList.***

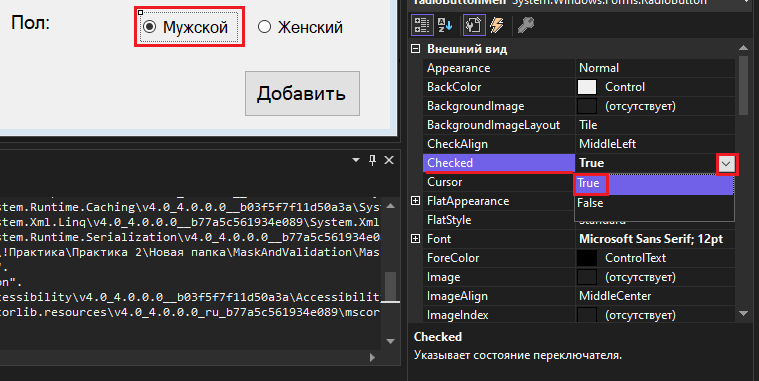
Теперь откройте Задачи **roleIDComboBox** (или найдите поле ***DataSourse*** в его свойствах), выберите галочку в задачах 🡪 в строке «**Источник данных**» выберите ранее созданный ***BindingSourceRole.*** 🡪 ниже, в отражаемом значении выбираем «***Name***» 🡪 далее в поле возвращаемое значение «***ID***».

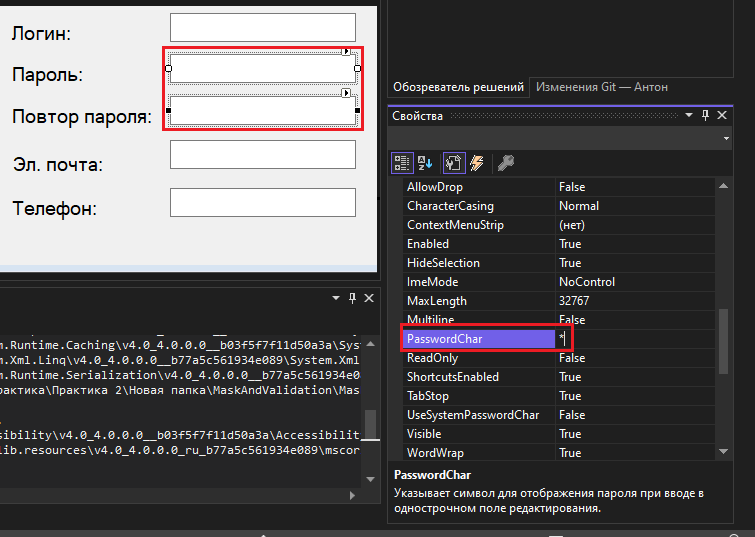


## После переходим к изменению элемента PhonemaskedTextBox. Выбираем элемент и в его свойствах находим поле *Mask* и пишем туда 🡪

## +7 (000) 000-00-00

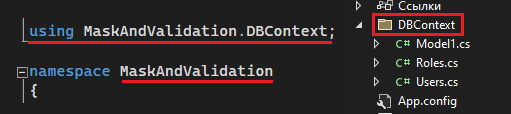
## (Нули означают любую цифру)

Дальше в **одном** из элементов **radioButtonMen** (или **radioButtonWoman**), ставим в свойство *Checked* в***True****.*

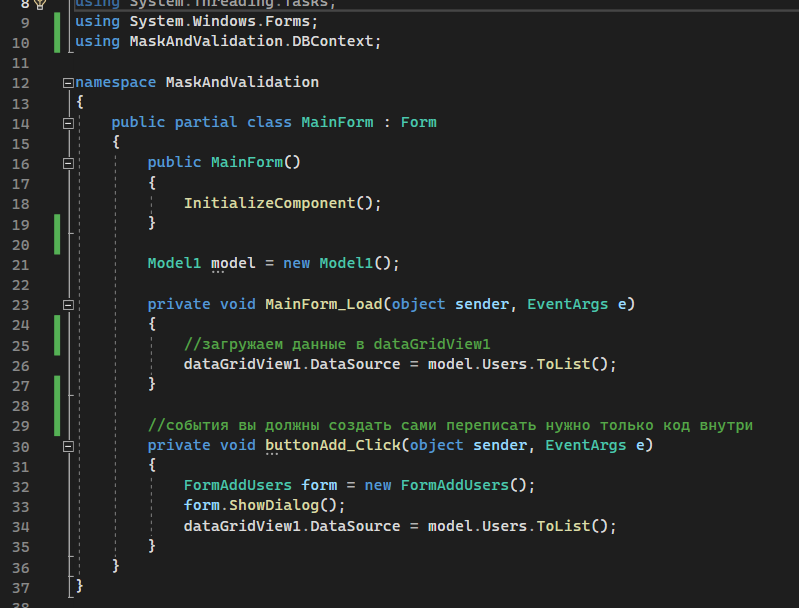
Для элементов **passwordTextBox** и **passwordTextBox2** меняем свойство *PasswordChar* на **\* .**

## 7) Написание кода программы

**Обязательно на всех формах добавляем библиотеку** !!!  
using ***Название\_Проекта.Название*\_*Папки\_с\_Моделью*;**

****

**Комментарии к коду переписывать не нужно!!!**

Ниже указан код **MainForm**. В пространстве класса необходимо создать статическую переменную класса Model (который мы ранее добавили через Code First), после добавить событие Form\_Load() и button\_Click();

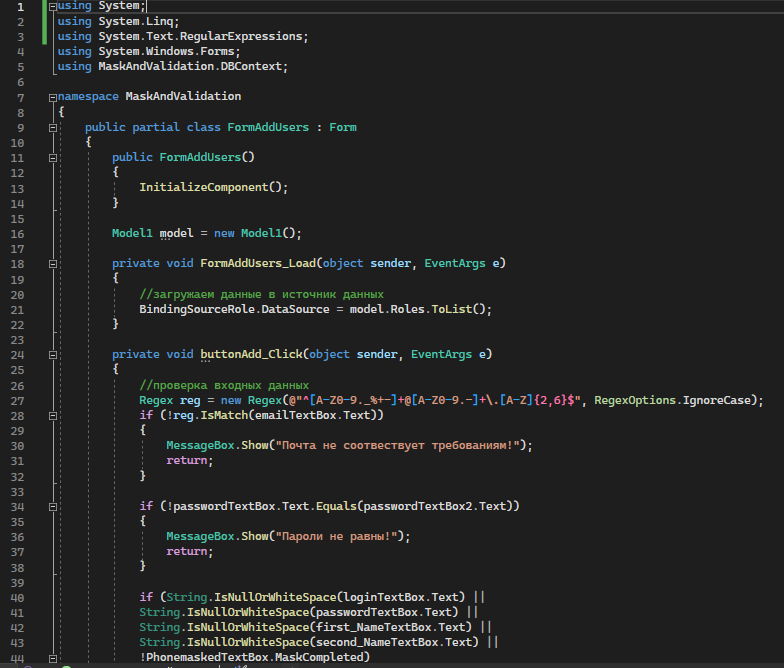
Ниже показан код, который необходимо написать на формах **FormAddUsers**. Необходимо передать данные в ранее созданный «Источник данных», это можно сделать в событии **FormAddUsers\_Load.**

Также для связи с БД нужно создать экземпляр класса **Model1.**

Также необходимо довить событие **button\_Click** и внутри него прописать проверки на валидность и корректность данных и внос новых данных в БД.

|  |
| --- |
| Model1 model = new Model1();  private void FormAddUsers\_Load(object sender, EventArgs e)  {  //загружаем данные в источник данных  BindingSourceRole.DataSource = model.Roles.ToList();  }  private void buttonAdd\_Click(object sender, EventArgs e)  {  //проверка входных данных  Regex reg = new Regex(@"^[A-Z0-9.\_%+-]+@[A-Z0-9.-]+\.[A-Z]{2,6}$",  RegexOptions.IgnoreCase);  if (!reg.IsMatch(emailTextBox.Text))  {  MessageBox.Show("Почта не соотвествует требованиям!");  return;  }  if (!passwordTextBox.Text.Equals(passwordTextBox2.Text))  {  MessageBox.Show("Пароли не равны!");  return;  }  if (String.IsNullOrWhiteSpace(loginTextBox.Text) ||  String.IsNullOrWhiteSpace(passwordTextBox.Text) ||  String.IsNullOrWhiteSpace(first\_NameTextBox.Text) ||  String.IsNullOrWhiteSpace(second\_NameTextBox.Text) ||  !PhonemaskedTextBox.MaskCompleted)  {  MessageBox.Show("Заполните все поля!");  return;  }  //Заполнение данных о новом пользователе  Users users = new Users();  users.ID = 0;  users.Login = loginTextBox.Text;  users.Password = passwordTextBox.Text;  users.Email = emailTextBox.Text;  users.Phone = PhonemaskedTextBox.Text;  users.First\_Name = first\_NameTextBox.Text;  users.Second\_Name = second\_NameTextBox.Text;  users.RoleID = (int)roleIDComboBox.SelectedValue;  users.Gender = radioButtonMen.Checked ? "Мужской" : "Женский";  try  {  //сохранение данных в БД  model.Users.Add(users);  model.SaveChanges();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  MessageBox.Show("Данные добавленны!");  Close();  }  private void buttonBack\_Click(object sender, EventArgs e)  {  Close();  } |

Повторюсь что события должны создаться автоматически, а уже код внутри них вы должны будете переписать.

**Скрин кода класса формы FormAddUsers:**

