

Практическая работа 58

Создание в Microsoft Visual Studio кнопочной формы, а также простых и усложненных ленточных форм для работы с данными и реализация вычисляемых полей.

Создание табличных форм для отображения данных

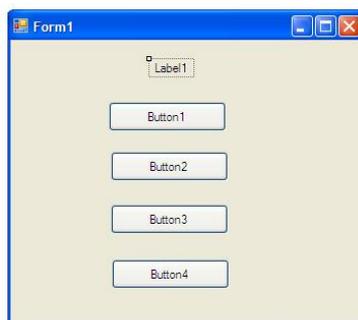
Цель работы: Получить практический опыт создания в Microsoft Visual Studio кнопочной формы.

Перечень оборудования и программного обеспечения

Персональный компьютер
Microsoft Office (Word, Visio, Access)
MS SQL Server
MS Visual Studio

Краткие теоретические сведения

Создание пользовательского интерфейса начнем с создания главной кнопочной формы. Запустите Microsoft Visual Studio и создайте новый проект. После появления стандартного окна среды разработки в рабочей области на форму поместите надпись (Label) и четыре кнопки (Button).



После создания объектов перейдем к настройке их свойств. Начнем с настройки свойств формы. На панели свойств задайте свойства формы как представлено ниже:

FormBorderStyle (Стиль границы формы): Fixed3D;

MaximizeBox (Кнопка разворачивания формы во весь экран): False;

MinimizeBox (Кнопка свертывания формы на панель задач): False;

Text (Текст надписи в заголовке формы): База данных "Студент".

Задайте свойства надписи следующим образом:

AutoSize (Авторазмер): False;

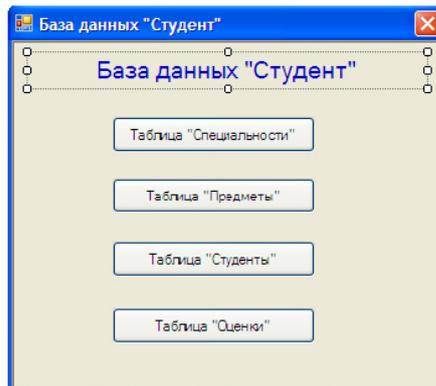
Font (Шрифт): MicrosoftSansSerif, размер 14;

ForeColor (Цвет текста): Темно синий;

Text (Текст надписи): База данных "Студент";

TextAlign (Выравнивание текста): MiddleCenter.

У кнопок задайте надписи (свойство "Text") как показано на рисунке



Теперь перейдем к созданию простых ленточных форм для работы с данными. Для начала создадим ленточную форму, отображающую таблицу "Специальности". Добавьте в проект новую пустую форму.



В верхней части новой формы создайте надпись (Label), как это показано на рисунке.

На панели свойств задайте свойства формы следующим образом:

FormBorderStyle (Стиль границы формы): Fixed3D;

MaximizeBox (Кнопка разворачивания формы во весь экран): False;

MinimizeBox (Кнопка сворачивания формы на панель задач): False;

Text (Текст надписи в заголовке формы): Таблица "Специальности".

Задайте свойства надписи как показано ниже:

AutoSize (Авторазмер): False;

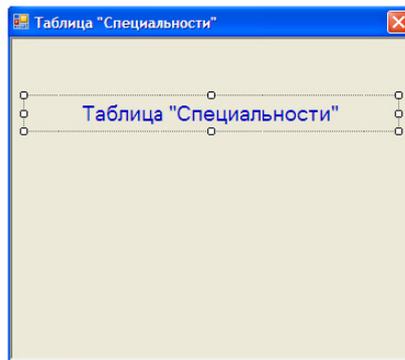
Font (Шрифт): MicrosoftSansSerif, размер 14;

ForeColor (Цвет текста): Темно синий;

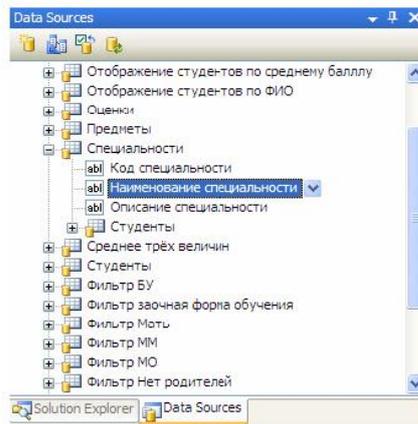
Text (Текст надписи): Таблица "Специальности";

TextAlign (Выравнивание текста): MiddleCenter.

После настройки всех вышеперечисленных свойств форма будет выглядеть следующим образом:



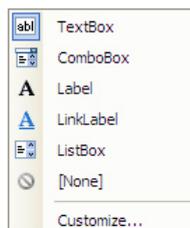
Теперь поместим на форму поля таблицы "Специальности". Сначала откройте панель "Источники данных" (DataSources) на вкладке Данные – показать источники данных. На панели "Источники данных" отобразите поля таблицы "Специальности", щелкнув по значку "+", расположенному слева от имени таблицы.



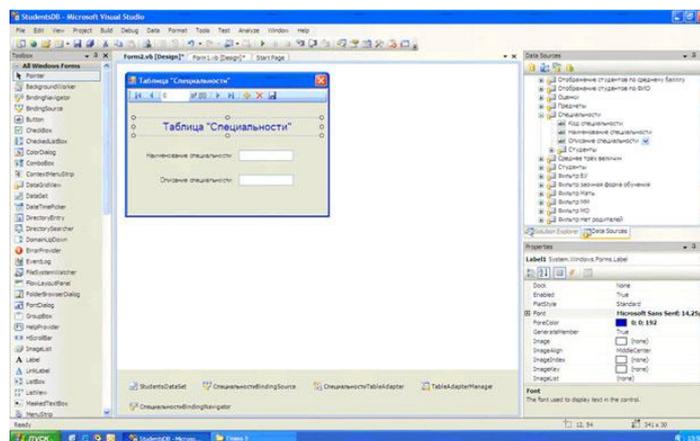
Панель "Источники данных" примет вид, представленный на рисунке.

Под полями таблицы специальности в виде подтаблицы располагается таблица "Студенты". Подтаблица показывает, что таблица "Студенты" является вторичной по отношению к таблице специальности.

При выделении, какого либо поля таблицы, оно будет отображаться в виде выпадающего списка, позволяющего выбирать объект, отображающий содержимое выделенного поля.



Для того чтобы поместить на новую форму поля таблицы их необходимо перетащить из панели "Источники данных" на форму. Из таблицы "Специальности" перетащите мышью на форму поля "Наименование специальности" и "Описание специальности". Форма примет вид, представленный на рисунке



После перетаскивания полей с панели "Источники данных" на форму в верхней части формы появилась навигационная панель, а в нижней части рабочей области среды разработки появились пять невидимых объектов. Эти объекты предназначены для связи формы с таблицей "Специальности", расположенной на сервере. Рассмотрим функции этих объектов:

StudentDataSet (Набор данных Student) - обеспечивает подключение формы к конкретной БД на сервере (в нашем случае это БД Students);

СпециальностиBindingSource (Источник связи для таблицы "Специальности") - обеспечивает подключение к конкретной таблице (в нашем случае к таблице специальности), а также позволяет управлять таблицей;

СпециальностиTableAdapter (Адаптер таблиц для таблицы "Специальности") - обеспечивает передачу данных с формы в таблицу и наоборот.

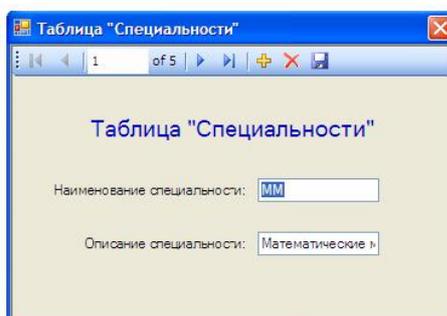
TableAdapterManager (Менеджер адаптера таблиц) - управляет работой объекта СпециальностиTableAdapter ;

СпециальностиBindingNavigator (Панель управления таблицей "Специальности") - голубая панель с кнопками управления таблицей, расположенная в верхней части формы

Теперь необходимо проверить работоспособность новой формы. Для отображения формы "Специальности" ее необходимо подключить к главной кнопочной форме, а затем запустить проект и открыть форму "Специальности" при помощи кнопки на главной кнопочной форме.

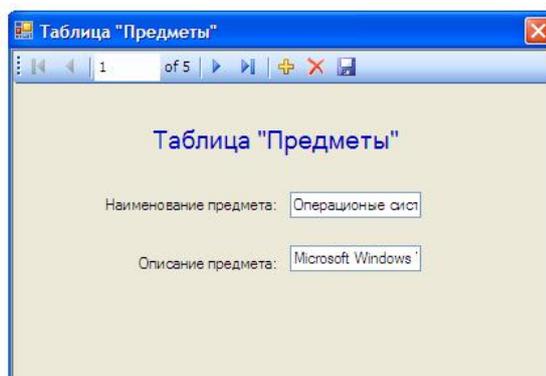
Запустите проект.

На экране появится главная кнопочная форма. Для открытия формы, отображающей таблицу "Специальности" на главной кнопочной форме нажмите кнопку "Таблица "Специальности"". Появится форма с соответствующей таблицей.



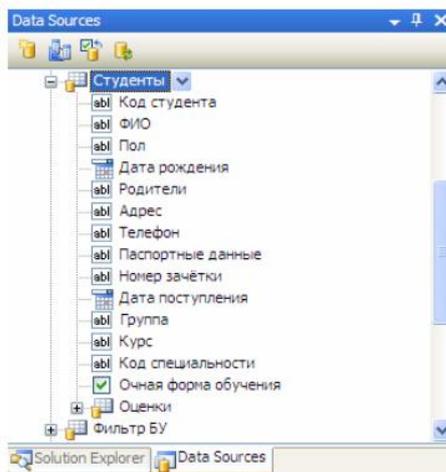
Проверьте работу панели навигации, расположенной в верхней части формы, понажимав на ней различные кнопки. Вернитесь в среду разработки, просто закрыв форму с таблицей "Специальности" и главную кнопочную форму.

Теперь создадим форму для просмотра таблицы предметы. Добавьте в проект новую форму. На форму добавьте надпись. Настройте свойства формы и надписи, как это было сделано для формы таблицы "Специальности". Затем из таблицы "Предметы" на новую форму поместите поля "Наименование предмета" и "Описание предмета". После выполнения всех вышеописанных действий форма для таблицы предметы будет выглядеть следующим образом:



Проверьте работу панели навигации, нажатием на кнопки на данной панели в верхней части формы. Для возвращения в среду разработки закройте форму таблицы "Предметы" и главную кнопочную форму.

Теперь создадим простую ленточную форму для отображения таблицы "Студенты". Для начала отобразите поля таблицы "Студенты" на панели "Источники данных", щелкнув по знаку "+", расположенному слева от названия таблицы. Отобразятся все поля таблицы "Студенты".



Поля "Дата рождения" и "Дата поступления" отображаются объектом "Выбор даты" (DatePicker), так как данные поля содержат значения дат. Поле "Очная форма обучения" является логическим, следовательно, для его отображения используется объект "Переключатель" (CheckBox). Остальные поля отображаются при помощи текстовых полей ввода (TextBox).

ФИО:	Иванов А.И.
Пол:	Мужской
Дата рождения:	12 декабря 1983 г.
Родители:	Отец и Мать
Адрес:	Москва
Телефон:	+74957898674
Паспортные данные:	8567-557543
Номер зачётки:	13245
Дата поступления:	1 сентября 2007 г.
Группа:	ММ11
Курс:	1
Код специальности:	1
Очная форма обучения:	<input checked="" type="checkbox"/>

Создайте новую форму и поместите в ее верхнюю часть надпись. Задайте заголовок формы Таблица "Студенты". В верхнюю часть формы поместите надпись. Настройте свойства формы и надписи, аналогично формам созданным ранее. На форму с панели "Источники данных" переместите все поля кроме поля "Код студента", так как данное поле является первичным полем связи.

Аналогичным образом создайте форму для отображения таблицы "Оценки". Форма для отображения таблицы "Оценки" будет выглядеть следующим образом:

Подключите вновь созданную форму таблицы "Оценки" к главной кнопочной форме. Проверьте таблицу "Оценки", запустив проект, и на главной кнопочной форме нажмите кнопку "Таблица "Оценки"". Появится вновь созданная форма.

Модернизируем форму для таблицы "Студенты". Сначала программно продублируем кнопки панели навигации, расположенной в верхней части формы. Откройте проект, разработанный в предыдущей работе и отобразите форму таблицы студенты. В нижней части формы расположите семь кнопок, как это показано на рисунке.

В качестве обработки кнопок формы "Студенты" наберите команды для перехода по записям

СтудентыBindingSource.MoveFirst(); - к первой записи

СтудентыBindingSource.MovePrevious(); - к предыдущей записи

СтудентыBindingSource.AddNew(); - добавить запись

СтудентыBindingSource.MoveLast(); - к последней записи

СтудентыBindingSource.MoveNext(); - к следующей записи

СтудентыBindingSource.RemoveCurrent();- удалить запись

Сохранить запись:

Me.Validate() - проверяет введенные в поля данные на соответствие типам данных полей;

Me.СтудентыBindingSource.EndEdit() - закрывает подключение с сервером;

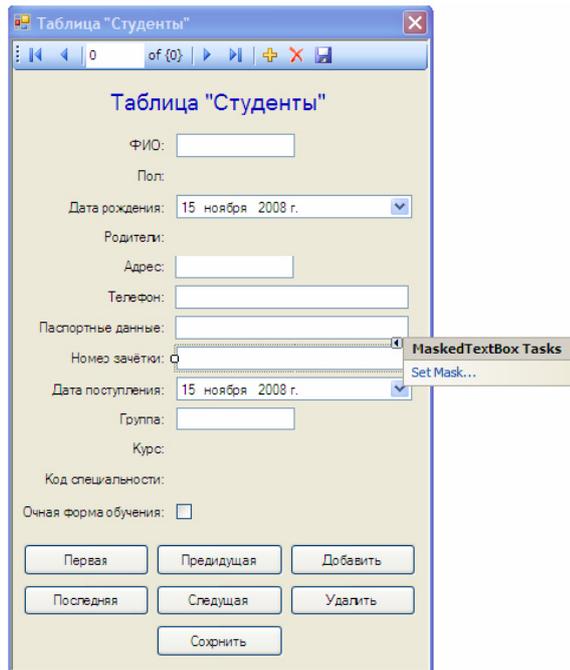
Me.TableAdapterManager.UpdateAll(Me.StudentsDataSet) - обновляет данные на сервере.

Для проверки работы созданных кнопок запустите проект откройте форму "Таблица "Студенты"" и нажмите каждую из кнопок.

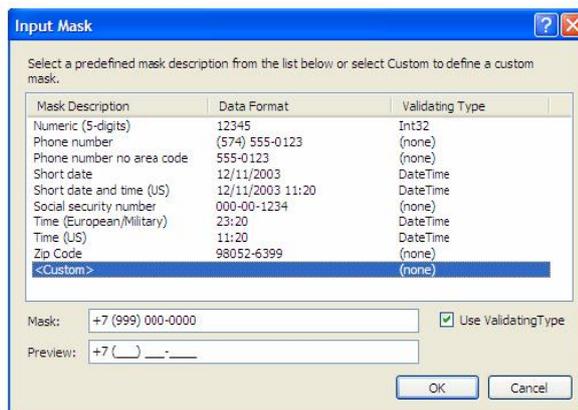
Измените объекты, отображающие поля для более удобного ввода информации. Для начала удалите текстовые поля ввода (TextBox), отображающие следующие поля таблицы "Студенты": "Пол", "Родители", "Телефон", "Паспортные данные", "Номер зачетки", "Курс" и "Код специальности". После удаления, перечисленных полей форма, отображающая таблицу "Студенты" примет следующий вид:

Для отображения полей "Телефон", "Паспортные данные" и "Номер зачетки" будем использовать текстовые поля ввода по маске (MaskedTextBox). Объект текстовое поле ввода по маске отсутствует в выпадающем списке объектов для отображения полей в окне "Источники данных", поэтому будем создавать данные объекты при помощи панели объектов (Toolbox), а затем подключать их к соответствующим полям

вручную. Для создания текстовых полей ввода по маске на панели объектов используется кнопка



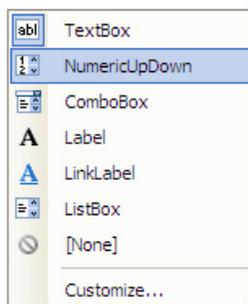
Создайте текстовые поля ввода по маске справа от надписей "Телефон", "Паспортные данные" и "Номер зачетки" и установите для них маски ввода.



Теперь нам необходимо подключить созданные текстовые поля ввода по маске к соответствующим полям. Для этого с панели "Источники данных" (DataSources) перетащите поле "Номер зачетки" на текстовое поле ввода по маске, расположенное справа от надписи "Номер зачетки". Прделайте такую же операцию с полями "Паспортные данные" и "Телефон", перетащив их на соответствующие им текстовые поля ввода по маске.

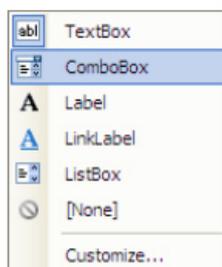
На этом мы заканчиваем работу с текстовыми полями ввода по маске и переходим к отображению поля "Курс" при помощи числового счетчика (объект NumericUpDown). Для этого, на панели "Источники данных" нажмите кнопку, расположенную справа от поля "Курс" и в выпадающем

списке выберите объект для отображения данного поля как "NumericUpDown".



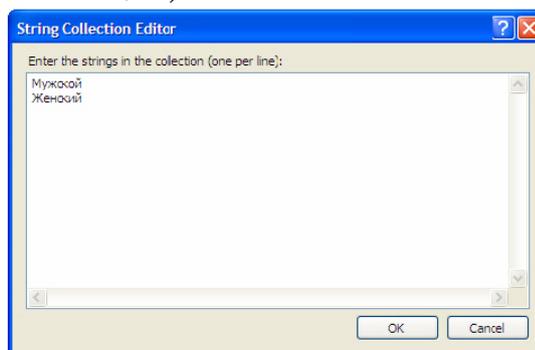
Затем перетащите поле на форму мышью, расположив, его справа от надписи "Курс".

Отобразим поля "Пол" и "Родители" в виде выпадающих списков (Объект ComboBox). Для этого, на панели "Источники данных" нажмите кнопку, расположенную справа от поля "Пол" и в выпадающем списке выберите объект для отображения данного поля как "ComboBox".



Такую же операцию проделайте с полем "Родители". Затем перетащите мышью поля на форму, расположив их напротив соответствующих надписей.

Теперь заполним выпадающие списки. Выделите выпадающий список, отображающий поле "Пол". На панели свойств (Properties) и нажмите кнопку в свойстве "Items" (Элементы списка). Появится окно "StringCollectionEditor" (Редактор строковых коллекций).

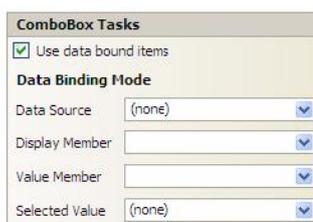


В появившемся окне в отдельных строках наберите элементы выпадающего списка: "Мужской" и "Женский". Затем нажмите кнопку "Ok".

Для выпадающего списка, отображающего поле "Родители", проделайте аналогичную операцию, только в качестве пунктов списка задайте: "Отец и Мать", "Мать", "Отец" и "Нет".

Поместите справа от надписи "Код специальности", неподключенный ни к каким полям выпадающий список. Для создания выпадающего списка на панели объектов воспользуйтесь кнопкой  ComboBox

После создание выпадающего списка подключим его к полю "Код специальности" из таблицы "Студенты" и настроим заполнение списка значениями поля "Наименование специальности" из таблицы студенты. Для этого выделите вновь созданный выпадающий список, отобразите меню действий и в меню действий включите опцию "Usedatabounditems" (Использовать связанные с данными элементы списка).



В панели действий под опцией "Usedatabounditems" расположены следующие параметры:

DataSource (Источник данных) - определяет таблицу или запрос из которого заполняется список;

DisplayMember (Член отображения) - определяет поле значениями которого заполняется список;

ValueMember (Член значений) - определяет значения какого поля подставляются в связанное с выпадающим списком поле;

SelectedValue (Выбранное значение) - определяет связанное с выпадающим списком поле.

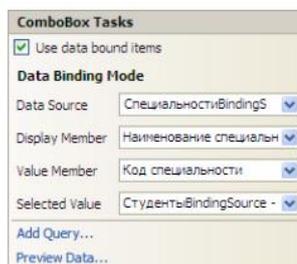
Параметр "DataSource" задайте как "Other Data Sources\Project Data Sources\StudentsDataSet\Специальности";

Параметр "DataMember" задайте как "Наименование специальности";

Параметр "ValueMember" задайте как "Код специальности";

Параметр "SelectedValue" задайте как "СтудентыBindingSource\Код специальности".

После задания всех вышеперечисленных параметров панель действий выпадающего списка примет вид:



Обратите внимание на то, что на панели невидимых объектов, расположенной в нижней части рабочей области среды разработки, появилось два новых объекта: "Специальности Binding Source" и "Специальности Table Adapter". Данные объекты предназначены для заполнения выпадающего списка значениями поля "Наименование специальности" таблицы "Специальности".

Форма "Студенты" должна иметь следующий вид:

Скриншот формы "Таблица 'Студенты'". Форма содержит следующие поля:

- ФИО:
- Пол:
- Дата рождения:
- Родители:
- Адрес:
- Телефон:
- Паспортные данные:
- Номер зачёта:
- Дата поступления:
- Группа:
- Курс:
- Код специальности:
- Очная форма обучения:

Кнопки: Первая, Предыдущая, Добавить, Последняя, Следущая, Удалить, Сохранить.

Для реализации вычисляемых полей рассмотрим форму, отображающую таблицу "Оценки". Рассмотрим вычисление поля "Средний балл" на основе среднего трех полей.

На форму для таблицы "Оценки" справа от поля "Средний балл" поместите кнопку Вычислить.

Скриншот формы "Таблица 'Оценки'". Форма содержит следующие поля:

- Код студента:
- Дата экзамена 1:
- Код предмета 1:
- Оценка 1:
- Дата экзамена 2:
- Код предмета 2:
- Оценка 2:
- Дата экзамена 3:
- Код предмета 3:
- Оценка 3:
- Средний балл:

Кнопка: Вычислить.

В обработчике процедуры "Button1_Click" наберите код, представленный на рисунке, вычисляющий среднее значение оценок:

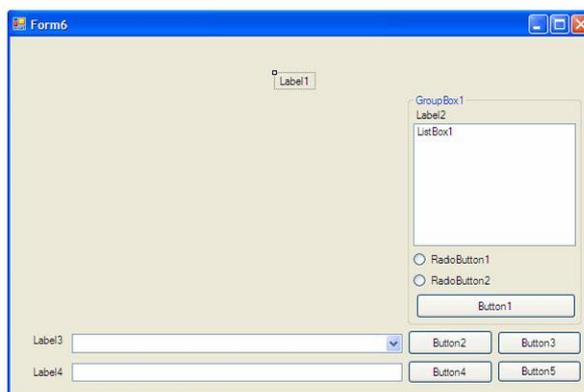
```
Средний_балл TextBox.Text = (Val(Оценка_1 TextBox.Text)+  
Val(Оценка_2 TextBox.Text)+Val(Оценка_3 TextBox.Text))/3;
```

На форме, отображающей таблицу "Оценки" нажмите кнопку "Вычислить". Будет вычислен средний балл по оценкам. Если нажать кнопку сохранения на панели инструментов формы, то средний балл будет сохранен в таблицу "Оценки".

Создадим табличные формы для фильтрации, сортировки и отображения данных.

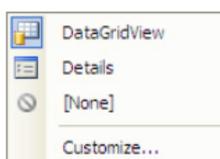
Табличную форму создадим на примере формы, отображающей таблицу "Студенты". Добавьте в проект новую форму и на нее поместите следующие объекты:

- четыре надписи (Label),
 - пять кнопок (Button),
 - выпадающий список (ComboBox),
 - текстовое поле ввода (TextBox),
 - группирующую рамку (GroupBox),
 - список (ListBox),
 - два переключателя (RadioButton).
- Расположите объекты как показано на рисунке.

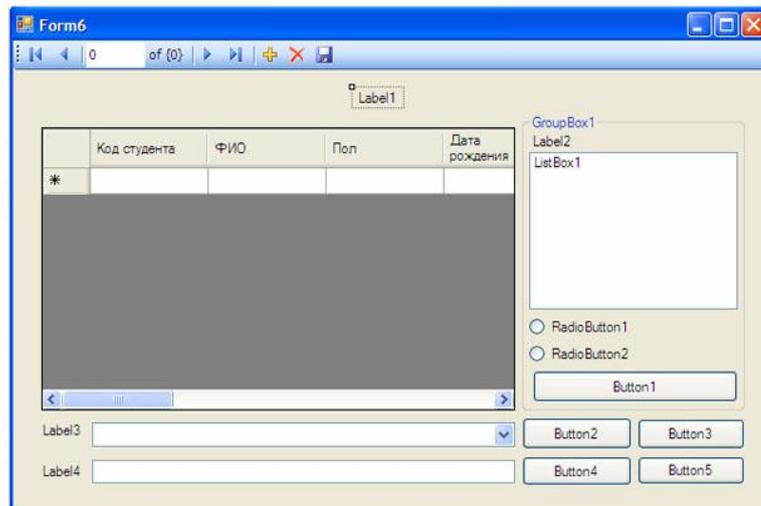


Для создания объекта группирующая рамка используется кнопка ### на панели объектов (Toolbox), а для создания переключателя - кнопка ##.

Добавим на форму таблицу для отображения данных (DataGridView) из таблицы "Студенты". Для этого на панели "Источники данных" (DataSources), нажмите кнопку , расположенную справа от таблицы "Студенты". В появившемся списке объектов для отображения всей таблицы выберите "DataGridView".



Перетащите таблицу "Студенты" из панели "Источники данных" на форму. Форма примет следующий вид:



Обратите внимание на то, что на форме появилась таблица для отображения данных, подключенная к таблице "Студенты". Также появились объекты связи и панель навигации



Теперь перейдем к настройке свойств объектов. Начнем с настройки свойств формы. Задайте свойства формы следующим образом:

FormBorderStyle (Стиль границы формы): Fixed3D;

MaximizeBox (Кнопка разворачивания формы во весь экран): False;

MinimizeBox (Кнопка свертывания формы на панель задач): False;

Text (Текст надписи в заголовке формы): Таблица "Студенты" (Табличный вид).

Задайте свойства надписей (Label1, Label2, Label3 и Label4) как:

AutoSize (Авторазмер): False;

Text (Текст надписи): "Таблица "Студенты" (Табличный вид)", "Поле для сортировки", "ФИО:" и "Критерий" (Соответственно для Label1, Label2, Label3 и Label4).

Для надписи Label1 задайте:

Font (Шрифт): MicrosoftSansSerif, размер 14;

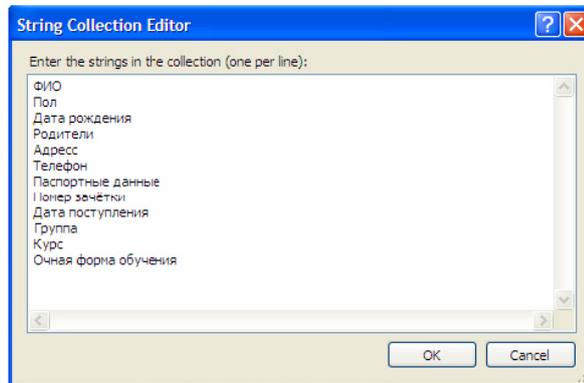
ForeColor (Цвет текста): Темно синий;

TextAlign (Выравнивание текста): MiddleCenter.

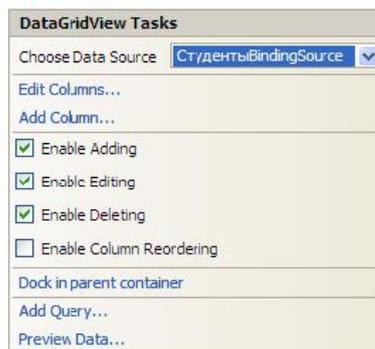
Задайте надписи на кнопках как: "Сортировать", "Фильтровать", "Показать все", "Найти" и "Закрыть" (Соответственно для кнопок Button1, Button2, Button3, Button4 и Button5). Для того чтобы нельзя было произвести сортировку не выбрав поля изначально заблокируем кнопку "Сортировать" (Button1).

У группирующей рамки задайте заголовок (Свойство Text) равным "Сортировка". У переключателей (Объекты RadioButton1 и RadioButton2) задайте надписи как "Сортировка по возрастанию" и "Сортировка по убыванию", а у переключателя "Сортировка по возрастанию" (RadioButton1) задайте свойство Checked (Включен) равное True (Истина).

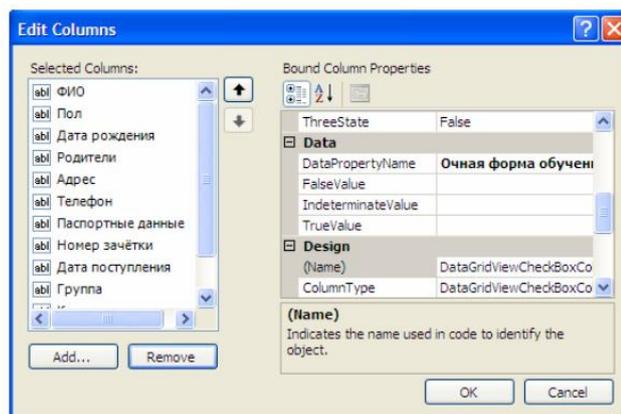
Заполните список (ListBox1) значениями, представленными на рис. 5, а затем нажмите кнопку "Ok".



Настроим таблицу для отображения данных, удалив из нее поля с кодами. Выделите таблицу на форме и отобразите ее меню действий, щелкнув по кнопке, расположенной в верхнем правом углу таблицы. В меню действий выберите пункт "Edit columns...".

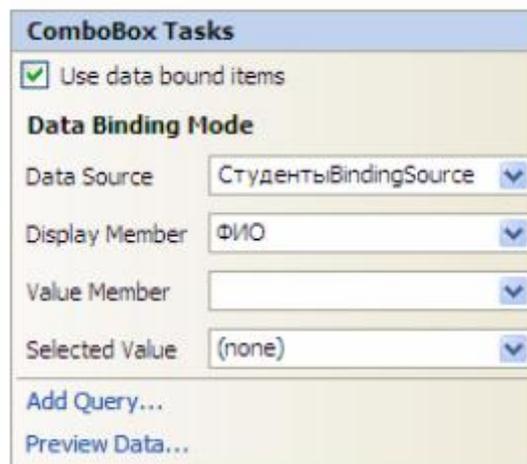


Появится окно настройки свойств полей таблицы "EditColumns".



В окне "EditColumns" из списка полей удалите поля "Код студента" и "Код специальности", выделив их и нажав кнопку "Remove" (Удалить). Список полей примет вид показанный на рисунке. Для закрытия окна редактирования полей, и сохранения изменений нажмите кнопку "Ok".

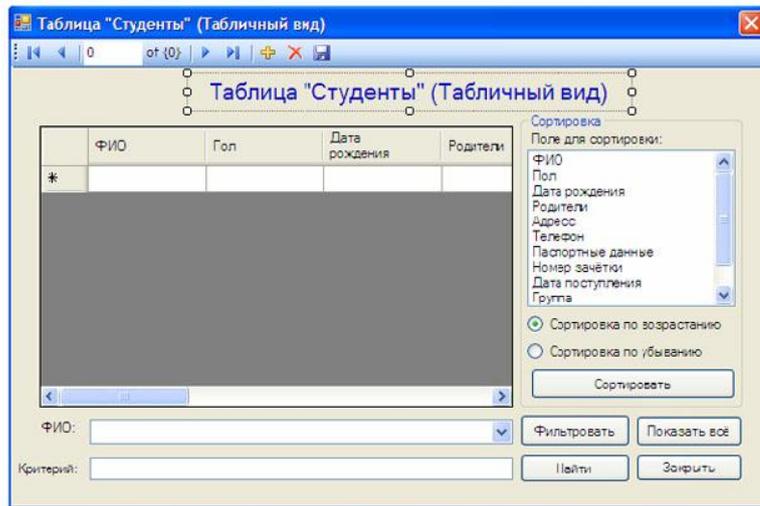
Настроим заполнение выпадающего списка именами студентов из таблицы студенты. Отобразите меню действий выпадающего списка. Включите опцию "UseDataBoundItems". Установите параметр "DataSource" равным "OtherDataSources\ProjectDataSources\StudentsDataSet\Студенты", а параметр "DisplayMember" равным "ФИО". Остальные параметры оставьте без изменений.



Закройте окно действий выпадающего списка. На панели невидимых объектов появится дополнительный объект связи "СтудентыBindingSource1", предназначенный для заполнения выпадающего списка.



После настройки всех вышеперечисленных свойств объектов новая форма примет вид:



На этом мы заканчиваем настройку свойств объектов и переходим к написанию кода обработчиков событий объектов.

Работу с кодом начнем с написания кода для разблокирования кнопки "Сортировать", при выборе пункта списка (`ListBox1`). Для создания процедуры события дважды щелкните по списку. Появится процедура обработки события, происходящего при выборе пункта списка

`ListBox1_SelectedIndexChanged`. В процедуре наберите команду разблокировки кнопки "Сортировать" (`Button1`): `Button1.Enabled = True;`

Теперь перейдем к созданию кода сортирующего таблицу в зависимости от выбранного поля и порядка сортировки при нажатии кнопки "Сортировать". Дважды щелкните по кнопке "Сортировать". Появится процедура "`Button1_Click`", выполняемая при щелчке по кнопке. В процедуре наберите код:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim Col As System.Windows.Forms.DataGridViewColumn
    Select Case ListBox1.SelectedIndex
        Case 0
            Col = DataGridViewTextBoxColumn2
        Case 1
            Col = DataGridViewTextBoxColumn3
        Case 2
            Col = DataGridViewTextBoxColumn4
        Case 3
            Col = DataGridViewTextBoxColumn5
        Case 4
            Col = DataGridViewTextBoxColumn6
        Case 5
            Col = DataGridViewTextBoxColumn7
        Case 6
            Col = DataGridViewTextBoxColumn8
        Case 7
            Col = DataGridViewTextBoxColumn9
        Case 8
            Col = DataGridViewTextBoxColumn10
        Case 9
            Col = DataGridViewTextBoxColumn11
        Case 10
            Col = DataGridViewTextBoxColumn12
    End Select
    If RadioButton1.Checked Then
        СтудентыDataGridView.Sort(Col, System.ComponentModel.ListSortDirection.Ascending)
    Else
        СтудентыDataGridView.Sort(Col, System.ComponentModel.ListSortDirection.Descending)
    End If
End Sub
```

Команда `Dim Col As System.Windows.Forms.DataGridViewColumn` создает переменную `Col` для хранения имени выбранного столбца таблицы;

Блок `SelectCase...EndSelect`, присваивает переменной `Col` имя выбранного столбца таблицы в зависимости от номера выбранного пункта списка (`ListBox1.SelectedIndex`). Если выбран первый пункт списка, то в переменную `Col` записывается столбец `DataGridViewTextBoxColumn2`, если второй, то - `DataGridViewTextBoxColumn3` и так далее. Хотелось бы отметить тот факт, что нумерация пунктов списка начинается с нуля, а нумерация столбцов с единицы. Первый столбец "ФИО" носит имя `DataGridViewTextBoxColumn2`, так как имя `DataGridViewTextBoxColumn1` имеет столбец заголовков строк;

Блок `If...EndIf` выполняет следующую операцию: если включен переключатель "Сортировка по возрастанию" (`RadioButton1`), то отсортировать таблицу по полю заданному в переменной `Col` по возрастанию (`СтудентыDataGridView.Sort (Col, System.ComponentModel.ListSortDirection.Ascending)`), иначе по убыванию (`СтудентыDataGridView.Sort (Col, System.ComponentModel.ListSortDirection.Descending)`).

Рассмотрим код обработчика события нажатия кнопки "Фильтровать" (`Button2`). Дважды щелкните по кнопке "Фильтровать" и в процедуре обработки события "`Button2_Click`" наберите код:
`СтудентыBindingSource.Filter = "ФИО=" & ComboBox1.Text & ""` :

`СтудентыBindingSource.Filter = "ФИО=" & ComboBox1.Text & ""`;

У объекта `СтудентыBindingSource` имеется текстовое свойство `Filter`, которое определяет условие фильтрации. Условие фильтрации имеет синтаксис: "`<Имя поля><Оператор>'<Значение>`". В нашем случае значение поля "ФИО" приравнивается к значению, выбранному в выпадающем списке (`ComboBox1.Text`).

Теперь перейдем к кнопке "Показать все", отменяющей фильтрацию записей. Дважды щелкните по вышеперечисленной кнопке. Появится процедура `Button3_Click`. В появившейся процедуре наберите команду `СтудентыBindingSource.Filter = ""`

Заметим, что если присвоить свойству "`Filter`" значение пустой строки (`""`), то его действие будет отменено .

Далее рассмотрим реализацию поиска информации в таблице. Дважды щелкните по кнопке "Найти". В появившейся процедуре обработки нажатия кнопки "`Button4_Click`" наберите следующий код.

```

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    For i = 0 To СтудентыDataGridView.ColumnCount - 1
        For j = 0 To СтудентыDataGridView.RowCount - 1
            СтудентыDataGridView.Item(i, j).Style.BackColor = Color.White
            СтудентыDataGridView.Item(i, j).Style.ForeColor = Color.Black
        Next j
    Next i
    For i = 0 To СтудентыDataGridView.ColumnCount - 1
        For j = 0 To СтудентыDataGridView.RowCount - 1
            If InStr(СтудентыDataGridView.Item(i, j).Value, TextBox1.Text) Then
                СтудентыDataGridView.Item(i, j).Style.BackColor = Color.AliceBlue
                СтудентыDataGridView.Item(i, j).Style.ForeColor = Color.Blue
            End If
        Next j
    Next i
End Sub

```

Рассмотрим более подробно код вышеприведенной процедуры. Данная процедура состоит из двух частей:

Первый блок For i=0.....Next i. перебирает все ячейки таблицы и устанавливает в них белый цвет фона и черный цвет текста. То есть, отменяет результаты предыдущего поиска;

Второй блок For i=0.....Next i. перебирает все ячейки таблицы и если они содержат текст, введенный в поле ввода (TextBox1), то устанавливает в них голубой цвет фона и синий цвет текста, чем выделяет искомые ячейки.

Наконец рассмотрим код для кнопки "Заккрыть". Дважды щелкните по этой кнопке и в появившейся процедуре "Button5_Click" наберите команду "Form.Close()", закрывающую форму.

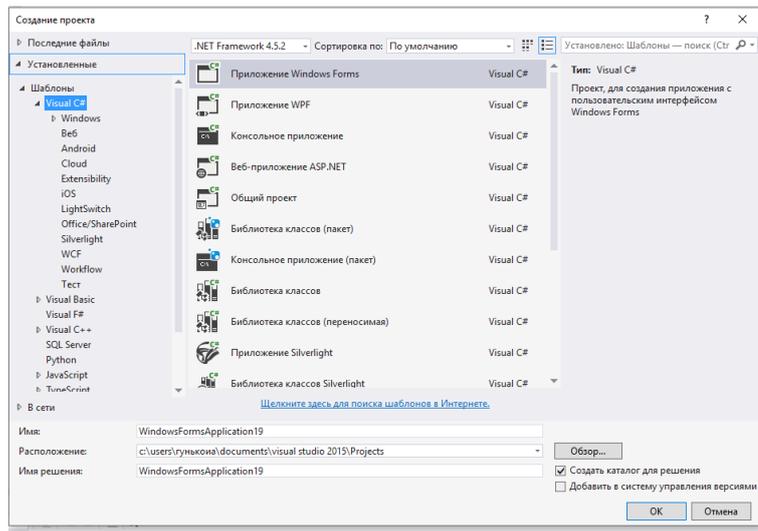
Проверьте, как работает поиск, фильтрация и сортировка записей в таблице, нажимая на соответствующие кнопки. После проверки работы формы для возвращения в среду разработки просто закройте все формы.

Задания

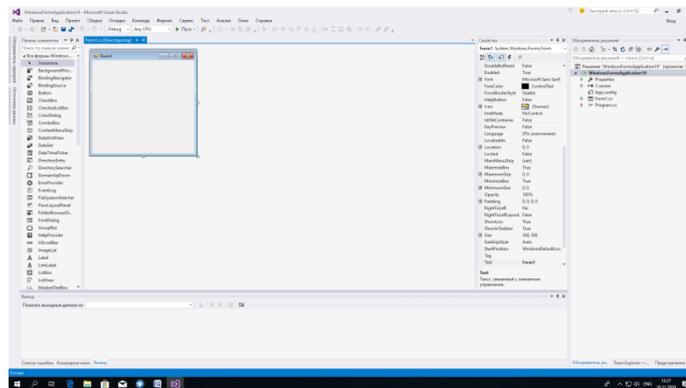
- 1 Изучить теоретические сведения.
- 2 В соответствии с вариантом задания создать кнопочную форму.
- 3 В соответствии с вариантом задания создать табличные формы.

Порядок выполнения работы

Откроем Visual Studio, выберем Создать проект, далее C# и Приложение Windows Forms



Появилось окно создания формы, в левой части инструменты для создания формы, справа свойства выделенного объекта, в данный момент это свойства формы, сразу заполним

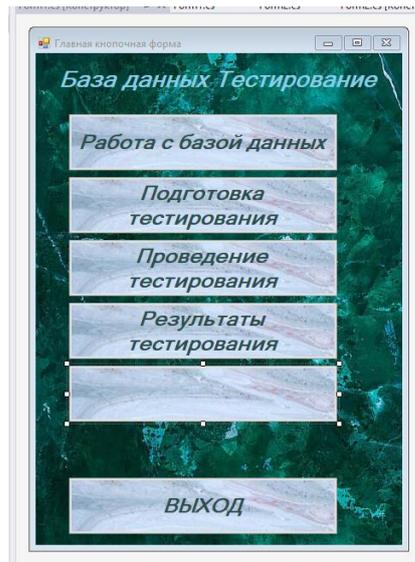


Появилось окно создания формы, в левой части инструменты для создания формы, справа свойства выделенного объекта, в данный момент это свойства формы, сразу заполним заголовок формы в свойстве Text, цвет фона в свойстве BackColor, можно вставить рисунок в свойстве BackgroudImage.

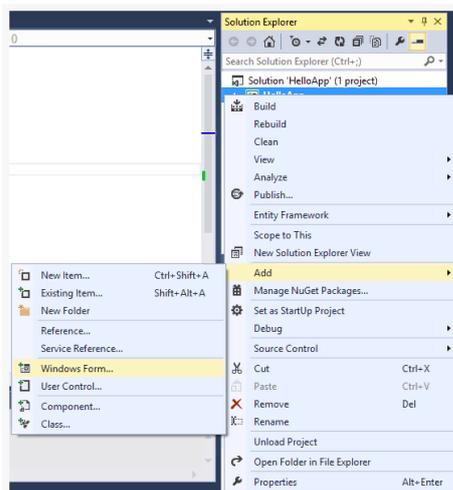
После оформления на форму поместим надпись (Label) и кнопку (Button). Оформим с помощью свойств созданные объекты.



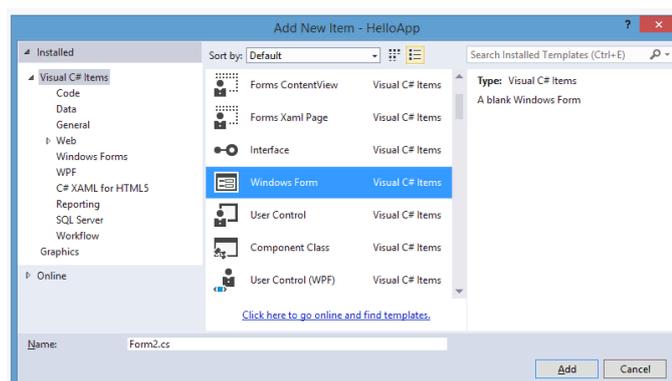
Затем можно скопировать кнопку необходимое количество раз, и вписать действия кнопок:



Чтобы добавить еще одну форму в проект, нажмем на имя проекта в окне Solution Explorer (Обозреватель решений) правой кнопкой мыши и выберем Add(Добавить)->Windows Form...



Дадим новой форме какое-нибудь имя, например, Form2.cs:



Итак, у нас в проект была добавлена вторая форма. Оформим ее и разместим управляющие элементы. Одна из кнопок пусть отвечает за возврат на главную форму



Теперь попробуем осуществить взаимодействие между двумя формами. Допустим, первая форма по нажатию на кнопку будет вызывать вторую форму. Двойным щелчком по кнопке перейдем в файл кода. Итак, мы попадем в обработчик события нажатия кнопки, который создается по умолчанию после двойного щелчка по кнопке:

```
private void button1_Click(object sender, EventArgs e)
{
}
}
```

Теперь добавим в него код вызова второй формы. У нас вторая форма называется Form2 (в свойствах формы раздел Name), поэтому сначала мы создаем объект данного класса, а потом для его отображения на экране вызываем метод Show, функция Hide() скрывает форму 1:

```
private void button1_Click(object sender, EventArgs e)
{
    Form nF2 = new Form2();
    nF2.Show(); // отображаем Form2
    this.Hide(); // скрываем Form1 (this - текущая форма)
}
}
```

Мы отображаем Form2 и скрываем Form1. Не закрываем, а именно скрываем, т.е. экземпляр Form1 продолжает существовать. После окончания работы с формой 2, необходимо ее закрыть и вернуться на главную форму, для чего перейдем ко второй форме и перейдем к коду кнопки «На главную». Итак, изменим файл кода второй формы на следующий:

```
this.Close();  
Form nF1 = Application.OpenForms[0];  
nF1.Show(); // отображаем Form1
```

OpenForms — получает коллекцию открытых форм приложения. В нашем случае мы сразу обратились к главной форме приложения (OpenForms[0]).

Теперь при закрытии Form2 у нас будет вызываться и отображаться та форма, которая породила закрывающуюся.

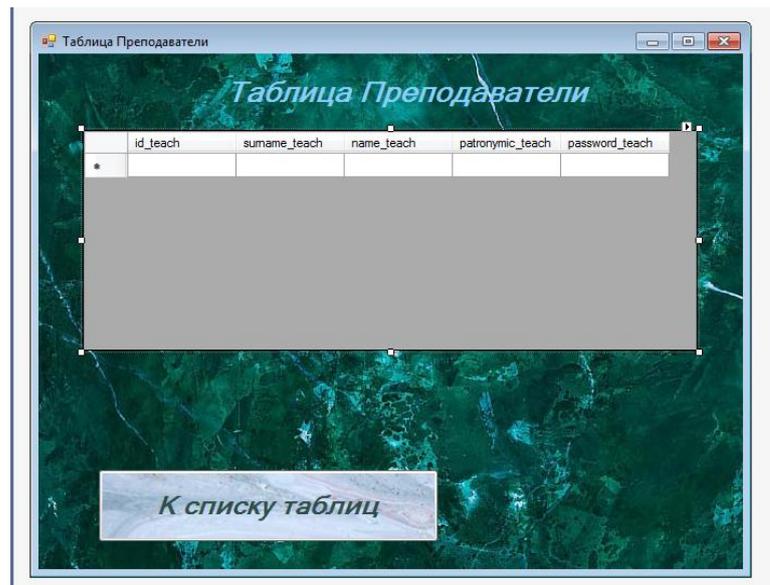
Если нужно, из второй кнопочной формы вызвать третью, то соответствующей кнопке задаем код:

```
Form nF4 = new Form4();  
nF4.Show();  
this.Hide();
```

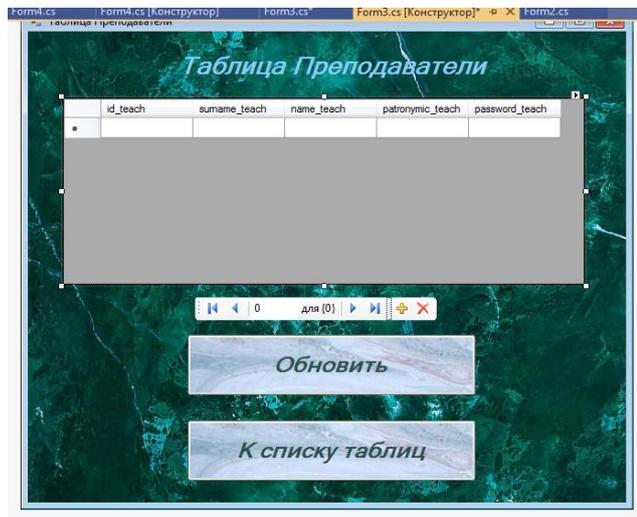
А для возврата код:

```
this.Close();  
Form nF2 = Application.OpenForms[1];  
nF2.Show();
```

Создадим форму, кроме кнопки возврата разместим на ней надпись (label), таблицу, подключим к ней таблицу базы данных:



Вместе с таблицей расположим на форме навигатор (BindingNavigator) и кнопку Обновить.



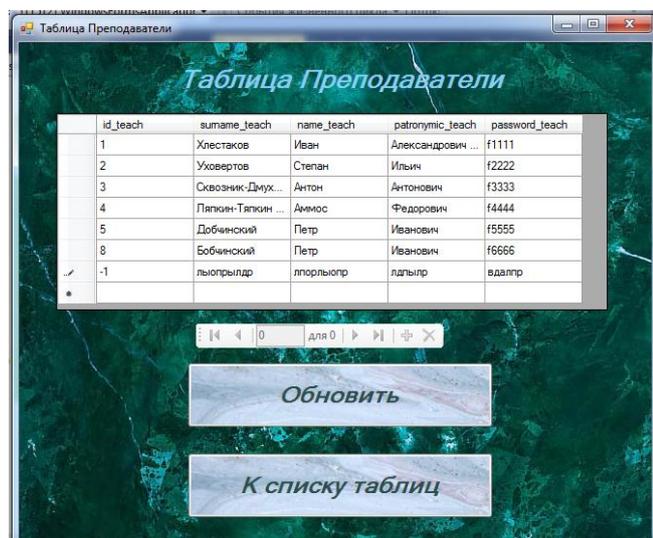
Для того, чтобы изменения в таблице сохранялись в базе данных в кнопку Обновить запишем следующий код:

```
this.<имя таблицы>TableAdapter.Update(this.<имя базы данных>DataSet);
```

В моем примере таблица называется teachers, а база данных testing:

```
this.teachersTableAdapter.Update(this.testingDataSet);
```

После запуска программы на форме появится таблица базы данных, в которую можно вносить изменения, добавлять и удалять записи, если нажать на кнопку Обновить, все изменения сохранятся в таблице базы данных.



Содержание отчета

- 1 Название работы
- 2 Цель работы
- 3 Перечень технических средств обучения

- 4 Порядок выполнения работы
- 5 Вывод

Варианты заданий

Варианты заданий представлены в практической работе № 39

Используемая литература

- Г.Н.Федорова Основы проектирования баз данных. М.: Академия, 2020
- Г.Н.Федорова Разработка, администрирование и защита баз данных. М.: Академия, 2018
- <https://victorz.ru/20160519140>