ЛАБОРАТОРНАЯ РАБОТА №6 (1 ЧАСТЬ)

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ

Рассмотрим процесс создания простейшей информационной системы. В качестве примера возьмем реализацию электронной записной книжки.

Наша записная книжка позволит:

- Вносить новые данные:
  - Создавать задачи с описанием и сроком выполнения;
  - Назначать исполнителя для решения конкретной задачи;
- Определять и выводить список всех невыполненных дел;
- Выводить все задачи в указанном интервале времени;
- Формировать письма исполнителю:
  - Указание выполнить дело;
  - Запрос о причинах задержки сроков выполнения.

Хранимые данные можно разместить в двух таблицах – «Дела» и «Исполнители».

Таблица «Дела»:

Название	Описание	Исполнитель	Срок исполнения (дата, время)	Отметка о выполнении

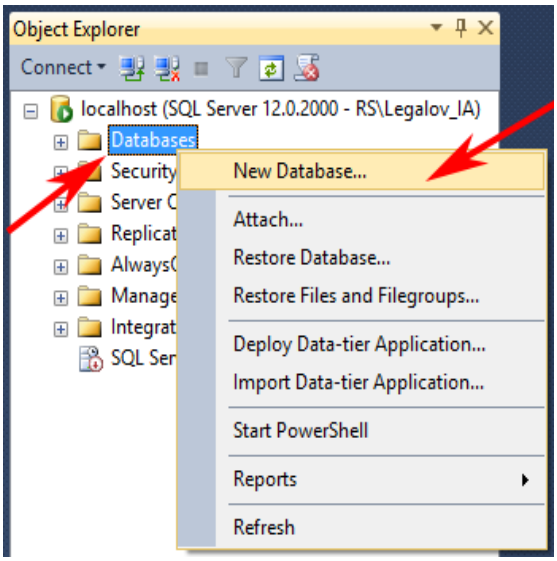
Таблица «Исполнители»:

Ф.И.О.	Название отдела	Телефон	E-Mail	Адрес

В первой части практической работы мы создадим новую базу данных и настроим параметры таблиц.

СОЗДАНИЕ БАЗЫ ДАННЫХ

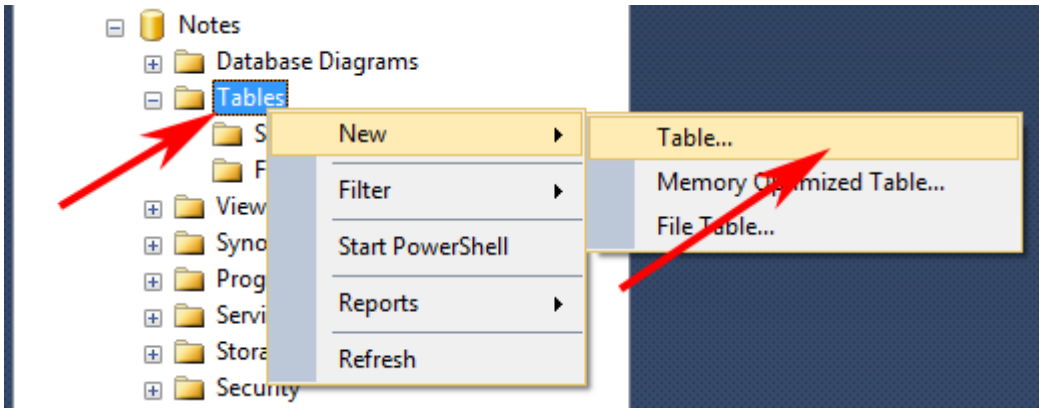
Начнем с создания новой пустой базы данных. В среде **SQL Server Management Studio** раскройте дерево панели **Object Explorer** и щелкните правой кнопкой мыши на узле **Databases**. Выберите пункт **New Database**:



В поле **Database name** введите название для нашей базы данных – **Notes**. Остальные параметры отвечают за различные настройки производительности и правил доступа, поэтому их значения можно не изменять. Нажмите кнопку **ОК**.

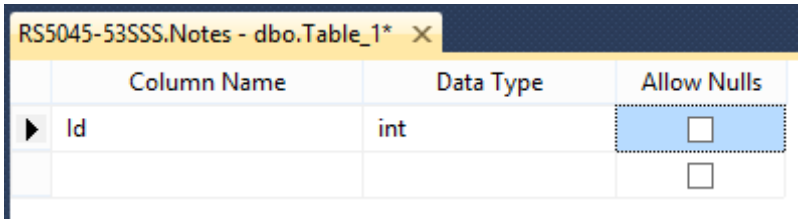
После небольшой паузы наша база отобразится в виде узла на панели **Object Explorer**, найдите внутри этого узла папку **Tables**. Этот раздел предназначен для хранения таблиц и сейчас он пуст. Исправим ситуацию, создав две таблицы – **Tasks** и **Persons**.

Щелкните правой кнопкой мыши на узле **Tables** и выберите пункты меню **New – Table**:

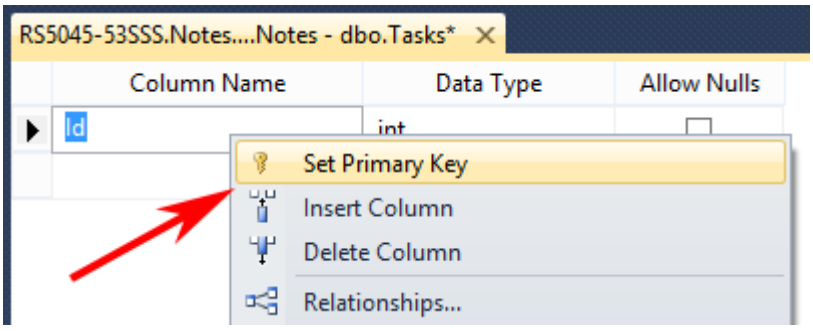


В правой части редактора открылся конструктор таблиц. Здесь можно редактировать столбцы таблицы, изменять их мена и различные свойства. Создадим первый столбец, который будет являться первичным ключом нашей таблицы.

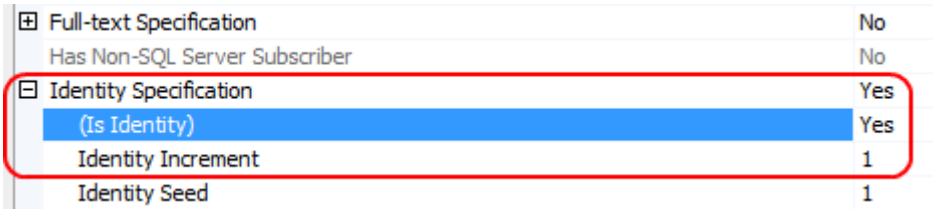
В поле **Column Name** введите имя **Id**, в поле **Data Type** выберите тип данных **int** и снимите галочку с поля **Allow Nulls**.



Щелкните правой кнопкой мыши на любом поле столбца Id и выберите пункт меню **Set Primary Key**:

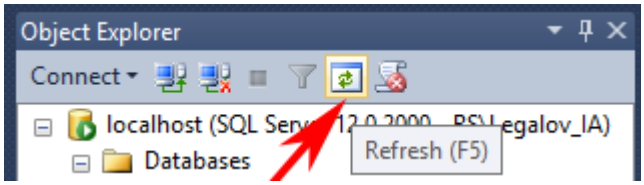


В нижней части окна конструктора расположена область свойств столбца, найдите там раздел **Identity Specification** и установите значение **Is Identity** в **Yes** для активации режима автоматического приращения значения.



Сохраните таблицу, нажав комбинацию клавиш **Ctrl+S**. Во время сохранения откроется диалоговое окно, где будет предложено ввести имя для таблицы, введите **Tasks**.

Новая таблица может отобразиться на панели **Object Explorer** с задержкой. Чтобы убедиться, что она была создана, нажмите кнопку **Refresh**:

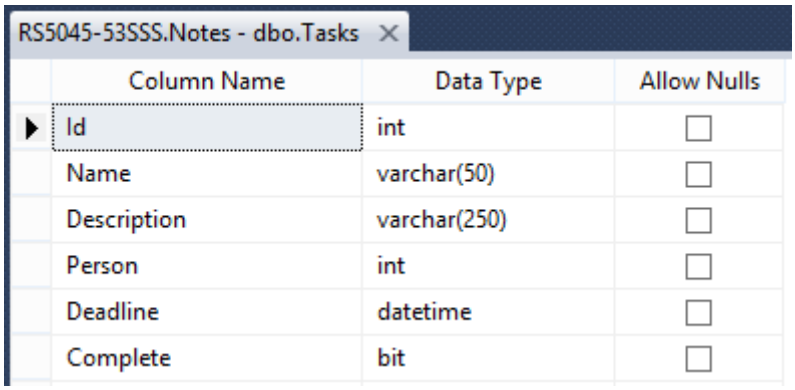


Новая таблица будет находиться в папке **Databases\Notes\Tables** под именем **dbo.Tasks**.

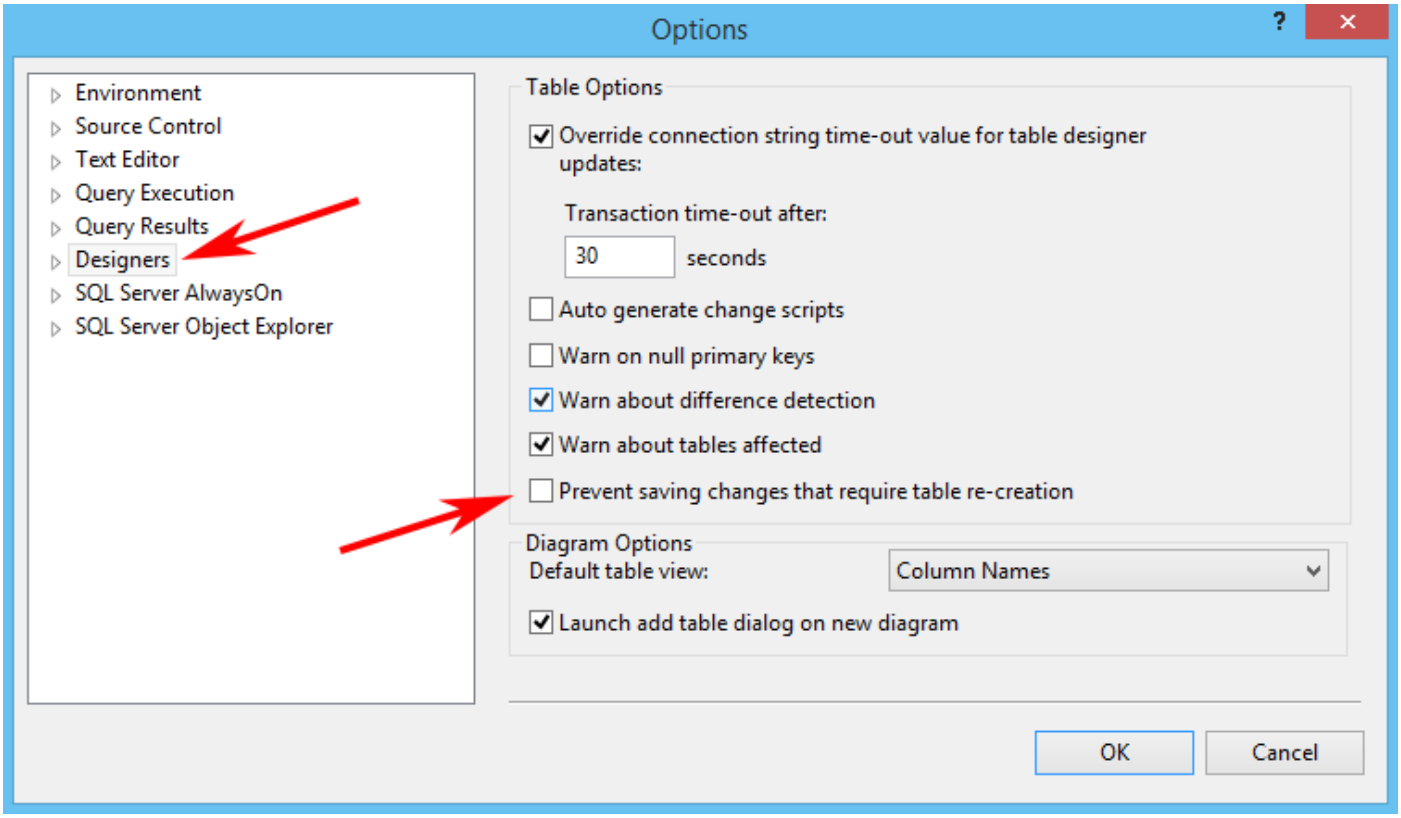
Аналогичным образом добавим в таблицу **Tasks** новые столбцы:

- **Name**, тип данных **int**;
- **Description**, тип данных **varchar**(50);
- **Person**, тип данных **varchar**(250);
- **Deadline**, тип данных **datetime**;
- **Complete**, тип данных **bit**.

Все вводимые данные являются обязательными, поэтому снимите все флаги **Allow Nulls**.



В некоторых случаях настройки среды **SSMS** запрещают добавление столбцов для уже существующей таблицы. Если при сохранении изменений выскакивает сообщение об ошибке, зайдите в меню **Tools – Options** и снимите галочку **Prevent saving changes that require table re-creation** в разделе **Designers**:



Самостоятельно добавьте в базу данных таблицу **Persons**. Используйте следующие параметры столбцов:

- **Id**, тип данных **int**, **первичный ключ, автоматическое приращение**;
- **Name**, тип данных **varchar**(50);
- **Department**, тип данных **varchar**(50);
- **Phone**, тип данных **varchar**(50);
- **Mail**, тип данных **varchar**(50);
- **Address**, тип данных **varchar**(100).

ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСА ИНФОРМАЦИОННОЙ СИСТЕМЫ

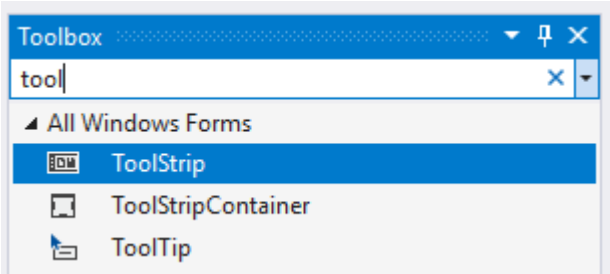
Настройка базы данных завершена, теперь мы можем приступить к написанию программы электронной записной книжки.

Откройте среду **Visual Studio** и создайте новый проект **Windows Forms** с именем **Notes**.

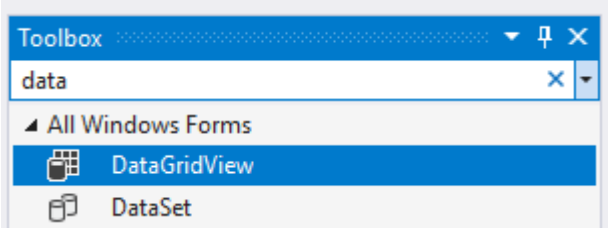
Наша записная книжка будет состоять из двух основных форм и двух дополнительных. Основные формы будут отображать содержимое таблиц, и содержать элементы управления. Две дополнительные формы будут использоваться для ввода данных при создании новых записей в БД.

После создания проекта среда Visual Studio автоматически сгенерирует одну форму с именем **Form1**. Откройте панель свойств и выполните следующие действия:

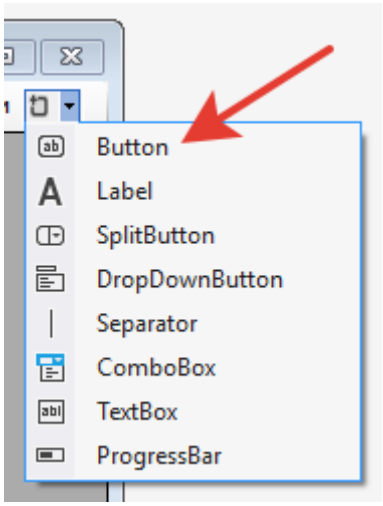
- Измените текст заголовка формы (Text) на "Заметки";
- Измените имя формы (Name) на NotesForm.
- Добавьте на форму элемент управления ToolStrip:



- Добавьте элемент управления DataGridView:



Выберите на форме добавленную панель инструментов ToolStrip, при выделении на панели появляется кнопка создания нового элемента. С ее помощью создайте шесть кнопок:



Откройте свойства первой кнопки и внесите следующие изменения:

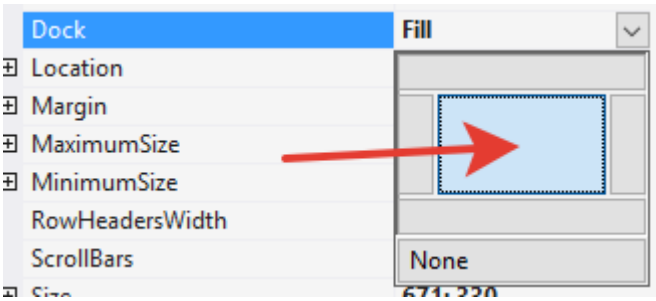
- Установите значение **DisplayStyle** как **Text**;
- Измените надпись на кнопке (**Text**) на "**Добавить**";
- Измените имя самой кнопки (**Name**) на "**AddButton**".

Для остальных кнопок также установите параметр **DisplayStyle** как **Text** и измените оставшиеся параметры, чтобы получить список:

- Конпка "**Удалить**" с именем "**DeleteButton**";
- Конпка "**Отметить** выполнение" с именем "**CompleteButton**";
- Конпка "**Напоминание**" с именем "**RemindButton**";
- Конпка "**Невыполненные**" с именем " **IncompleteButton**";
- Конпка "**Исполнители**" с именем "**PersonsButton**";

Для кнопки "**Невыполненные**" дополнительно установите параметр **CheckOnClick** в значение **true**.

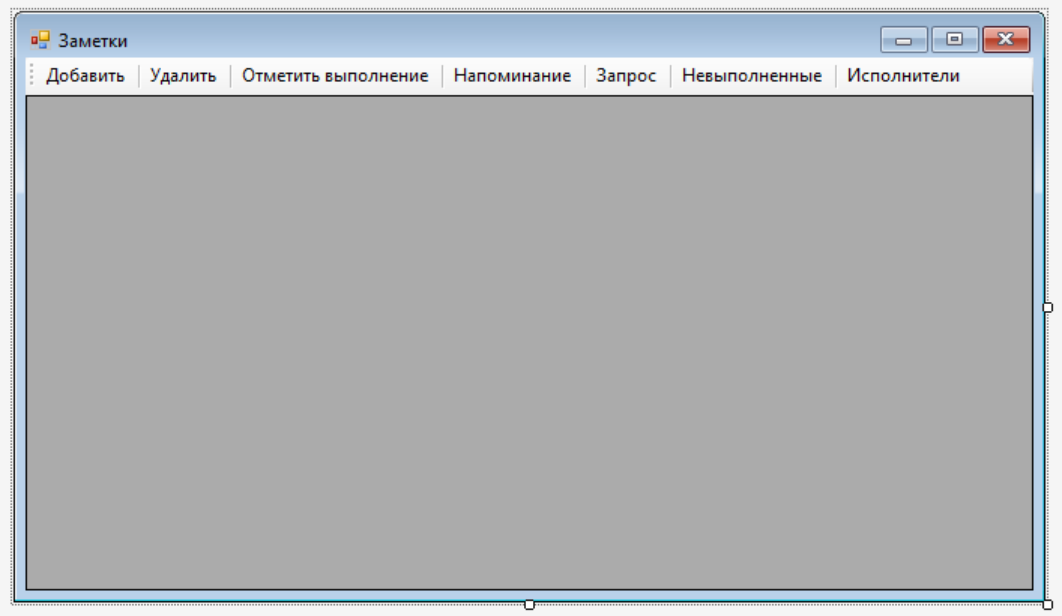
На форме выберите элемент управления **DataGridView** и откройте его свойства. Установите параметр **Dock** на значение **Fill**, для этого щелкните на поле справа и еще раз щелкните на центральную кнопку в открывшемся всплывающем окне:



Измените имя элемента (**Name**) на "**notesGrid**".

Установите параметр **ReadOnly** в значение **true**, параметр **SelectionMode** в значение **FullRowSelect**.

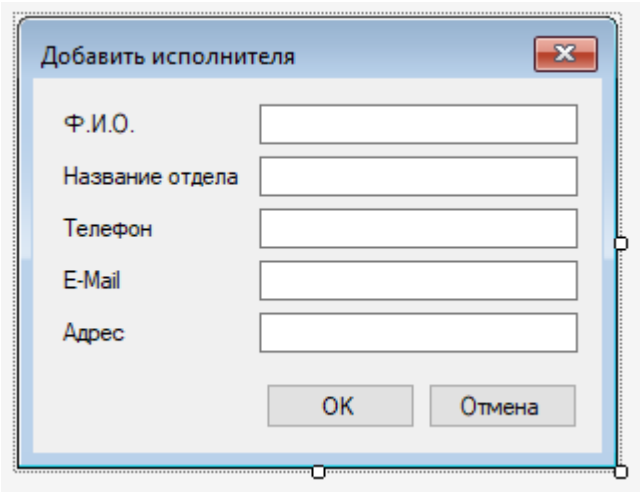
В результате мы должны получить следующую форму:



Самостоятельно создайте форму для отображения работы с исполнителями (**Persons**):

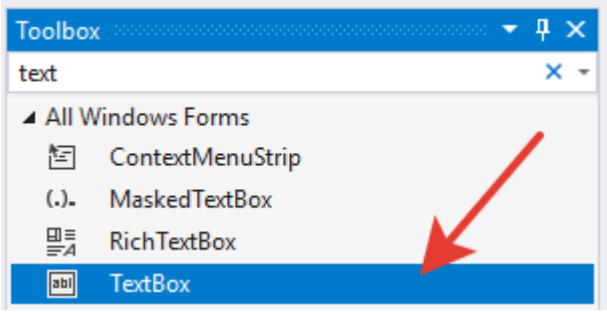
- Укажите заголовок формы как "**Исполнители**" и имя формы как "**PersonsForm**";
- Добавьте кнопки "**Добавить**" с именем "**AddButton**" и "**Удалить**" с именем "**DeleteButton**";
- Настройте элемент управления **DataGridView** и измените его имя на "**personsGrid**".

Теперь нам необходимо создать форму для добавления нового исполнителя. Итоговы результат должен выглядеть следующим образом:



Для этого добавьтев проект новую форму и измените текст ее заголовка (**Text**) на "**Добавить исполнителя**", а имя (**Name**) на "**AddPersonForm**". Дополнительно установите параметр **FormBorderStyle** в значение **FixedDialog** и параметры **MaximizeBox** и **MinimizeBox** в значение **false**.

Добавьте на форму пять элементов управления **TextBox** и расположите их столбиком.



Установите для каждого **TextBox** параметр **Modifiers** в значение **Public**.

Задайте следующие (**Name**) имена элементам **TextBox**:

- NameBox
- DepartmentBox
- PhoneBox
- MailBox
- AddressBox

Добавьте на форму две кнопки "**OK**" и "**Отмена**". Для кнопки "**OK**" задайте параметр **DialogResult** как **OK**, а для кнопки "**Отмена**" как **Cancel**.

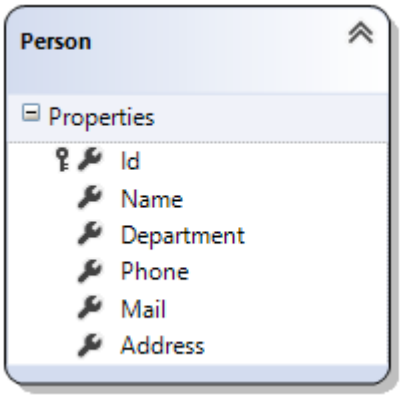
Добавьте на форму пять меток **Label** и расположите их напротив элементов ввода **TextBox**. Измените их текст на:

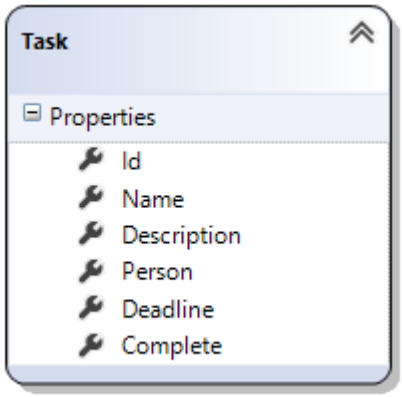
- Ф.И.О.
- Название отдела
- Теефон
- E-Mail
- Адрес

ЛАБОРАТОРНАЯ РАБОТА №6 (2 ЧАСТЬ)

РЕАЛИЗАЦИЯ БИЗНЕС-ЛОГИКИ ПРИЛОЖЕНИЯ

По аналогии с 3й частью лабораторной работы №1 добавьте в проект новый элемент **LINQ to SQL Classes** с именем "**NotesDatabase**". Создайте подключение к БД **Notes** и перенесите в конструктор отношений таблицы **Persons** и **Tasks**.





Откройте код формы "**Исполнители**" (**PersonsForm**) и добавьте следующий код в конец конструктора формы:

```
public PersonsForm() {
    InitializeComponent();

    personsGrid.DataSource = from p in db.Persons select p;
}
```

Этот код читает из базы данных содержимое таблицы Persons и выводит его в таблице **DataGridView**.

Откройте конструктор формы "**Исполнители**" (**PersonsForm**) и дважды щелкните на кнопке "**Добавить**". Будет сгенерирован новый обработчик события нажатия на кнопку. Вставьте в него следующий код:

```
var form = new AddPersonForm();
if (form.ShowDialog() == DialogResult.OK) {
    var p = new Person();
    p.Name = form.NameBox.Text;
    p.Department = form.DepartmentBox.Text;
    p.Phone = form.PhoneBox.Text;
    p.Mail = form.MailBox.Text;
    p.Address = form.AddressBox.Text;

    db.Persons.InsertOnSubmit(p);
    db.SubmitChanges();
}
personsGrid.DataSource = from p in db.Persons select p;
```

Здесь мы создаем новое диалоговое окно по добавлению исполнителя и отображаем его пользователю:

```
var form = new AddPersonForm();
if (form.ShowDialog() == DialogResult.OK) {
```

В случае если пользователь нажал кнопку ОК, мы создаем новый объект исполнителя и заполняем его данными из формы:

```
    var p = new Person();
    p.Name = form.NameBox.Text;
    p.Department = form.DepartmentBox.Text;
    p.Phone = form.PhoneBox.Text;
    p.Mail = form.MailBox.Text;
    p.Address = form.AddressBox.Text;
```

После этого добавляем нового исполнителя в базу данных:

```
db.Persons.InsertOnSubmit(p);
db.SubmitChanges();
```

Затем обновляем содержимое таблиц:

```
personsGrid.DataSource = from p in db.Persons select p;
```

Еще раз откройте конструктор формы "**Исполнители**" (**PersonsForm**) и дважды щелкните на кнопке "**Удалить**".

Вставьте следующий код в сгенерированный обработчик:

```
var ids = new List<int>();
var selected = personsGrid.SelectedRows;
foreach (DataGridViewRow row in selected) {
    ids.Add((int)row.Cells["Id"].Value);
}

var query = from p in db.Persons
            where ids.Contains(p.Id)
            select p;
foreach (var q in query) {
    db.Persons.DeleteOnSubmit(q);
}
db.SubmitChanges();

personsGrid.DataSource = from p in db.Persons select p;
```

Для удаления записи из таблицы сначала мы определить, что именно было выделено в окне формы:

```
var ids = new List<int>();
var selected = personsGrid.SelectedRows;
foreach (DataGridViewRow row in selected) {
    ids.Add((int)row.Cells["Id"].Value);
}
```

Мы получаем список выделенных строк **SelectedRows**, проходим по нему циклом **foreach** и получаем целочисленное значение ячеек столбца **Id**. Это значение мы сохраняем в список **ids** и в дальнейшем используем для определения удаляемых записей БД:

```
var query = from p in db.Persons
            where ids.Contains(p.Id)
            select p;
```

Удаляем записи из базы:

```
foreach (var q in query) {
    db.Persons.DeleteOnSubmit(q);
}
db.SubmitChanges();
```

Обновляем содержимое формы:

```
personsGrid.DataSource = from p in db.Persons select p;
```

Самостоятельно реализуйте обработку нажатия на кнопки "**Добавить**" и "**Удалить**" на форме "**Заметки**" (**NotesForm**).