

## Тема 18. Создание приложения (практика)

### Задание 1. Создание простейшего однотобличного приложения

Каждое приложение, использующее базы данных, обычно имеет несколько компонентов следующих трех типов:

- **Компоненты-наборы данных** (data set) **Table**, **Query** и прочие, непосредственно связывающиеся с базой данных. Это невидимые компоненты, невидимые при работе приложения.
- **Компоненты визуализации и управления данными**, такие как **DBGrid**, **DBText** и т.д.
- **Компоненты - источники данных** (data source), осуществляющие обмен информацией между компонентами первого типа (невизуальными) и второго (визуальными). Таким компонентом является **DataSource**.

Схема взаимодействия компонентов изображена на рис. 18.1:



Рис. № 18.1.

Построим простейшее приложение, работающее с базой данных. Будем использовать созданные ранее таблицы.

1. Создайте новый проект, установите для формы свойства **Caption** и **Name**.

2. В качестве набора данных будем использовать компонент **Table**. Он находится на странице **DataAccess** либо **BDE**, в зависимости от версии **Delphi**. Поместите компонент **Table** на форму. Для связи с необходимой таблицей базы данных служат два свойства:

- **DatabaseName** - псевдоним вашей базы данных,
- **TableName** - таблица из выбранной базы данных (Clients).

3. Поместите на форму компонент **DataSource** (находится на странице **DataAccess**), который будет источником данных для визуальных компонентов.

Установите связь между источником данных и набором данных. Для этого щелкните над свойством **DataSet** и из выпадающего списка выберете **Table1**. Если бы у вас было несколько наборов данных, они все были бы в этом списке.

4. Разместите компонент **DBGrid** (страница **DataControls**). Главное свойство всех компонентов визуализации и управления данными - **DataSource**. Установите это свойство в **DataSource1** (на данный момент у нас других и нет).

5. Для того чтобы таблица открывалась при запуске приложения автоматически, необходимо установить свойство **Active** компонента **Table** в **true**.

6. Создайте кнопку для закрытия приложения и соответствующий обработчик (рис. 18.2).

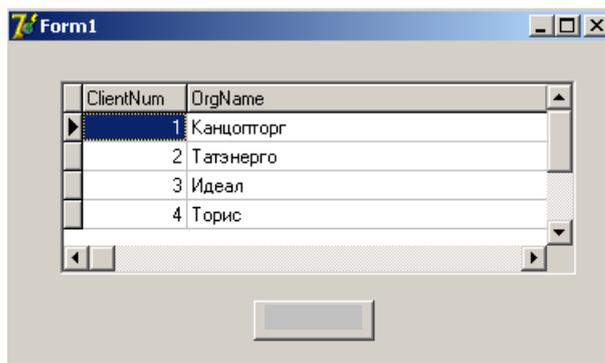


Рис. № 18.2.

Простейшее приложение готово. Запустите его и выполните с помощью сетки: просмотр данных, редактирование значений полей, добавление записей в конце таблицы. Обратите внимание: поля типа **Autoincrement** не доступны для редактирования.

## Задание 2. Развитие программы

1. Добавим еще один компонент, управляющий работой с таблицей: навигатор **DBNavigator** (страница **DataControls**). Компонент имеет ряд кнопок, служащих для управления данными: перемещение по записям, изменение, добавление и удаление. Свяжите навигатор с данными: установите свойство **DataSource** в **DataSource1**.

2. Настроим кнопки навигатора. Пользуясь свойством навигатора **VisibleButtons**, можно убрать любые ненужные в данном приложении кнопки, установив их значение в **false** (список кнопок и описания см. в табл. 3). Оставьте кнопки **nbFirst**, **nbPrior**, **nbNext**, **nbLast**, **nbInsert**, **nbDelete**. Остальные уберите.

Таблица 3. Кнопки навигатора.

	назначение
nbFirst	Перемещение к первой записи
nbPrior	Перемещение к предыдущей записи
nbNext	Перемещение к следующей записи
nbLast	Перемещение к последней записи
nbInsert	Вставить новую запись перед текущей
nbDelete	Удалить текущую запись
nbEdit	Редактировать текущую запись
nbPost	Послать отредактированную информацию в базу данных
nbCancel	Отменить результаты редактирования
nbRefresh	Очистить буфер, связанный с набором данных

3. Включите подсказки. Для этого установите свойство **ShowHint** в **true**. В свойстве **Hints** содержится список строк типа **TStrings**, содержащий тексты всплывающих ярлычков кнопок навигатора. По умолчанию эти тексты английские. Введите вместо них соответствующий набор русских подсказок.

4. Разместите на форме компонент **DBEdit** (страница **DataControls**). Этот компонент дает возможность редактирования значений отдельного поля. Для связи с полем установите свойство **DataSource** в **DataSource1**, а свойство **DataField** - в имя соответствующего поля. Установите компоненты **DBEdit** для каждого поля таблицы.

5. К каждому компоненту **DBEdit** добавьте подпись - компонент **Label**.

6. Установите свойство **ReadOnly** компонента **DBGrid1** в **true** - тем самым будет запрещено редактирование данных непосредственно в самой сетке.

7. Сделаем так, чтобы при перемещении по данным в сетке выделялись не отдельные поля, а вся строка целиком. Для этого в свойстве **Options**

компонента **DBGrid** установите подсвойства **dgRowSelect** и **dgAlwaysShowSelection** в true.

8. Настроим внешний вид сетки. Вызовите редактор колонок (свойство **Columns** компонента **DBGrid** или двойной щелчок левой кнопкой мышки на нем). В появившемся окне нажмите кнопку **Add All Fields** - отобразится список полей таблицы (рис. 18.3).

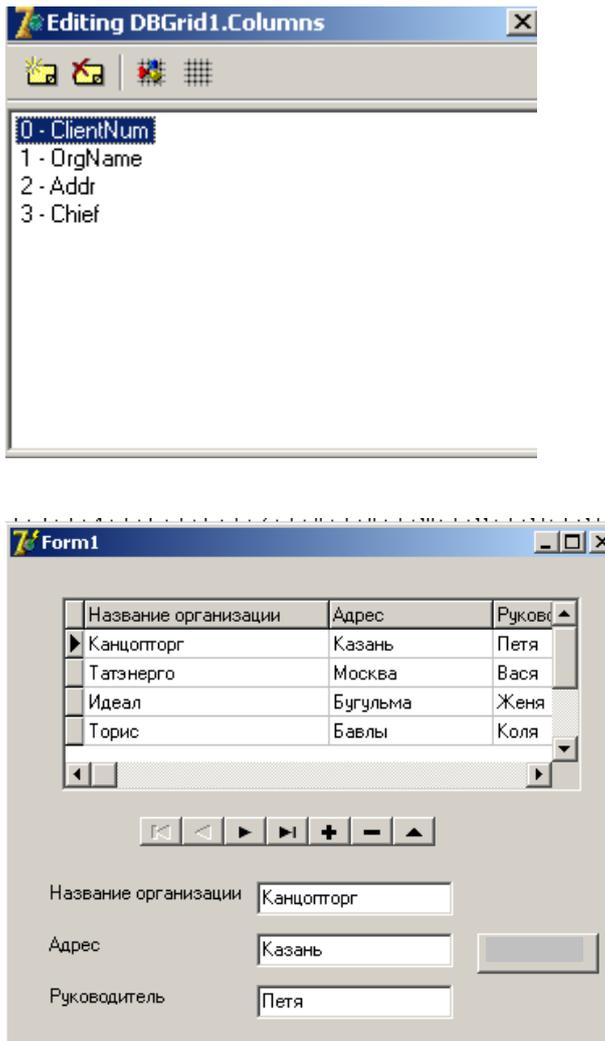


Рис. № 18.3.

Теперь можно настроить последовательность вывода полей на экран, их внешний вид, какие поля отображать, а какие нет. Выберете поле в списке - в инспекторе объектов отобразятся свойства этого поля. Шрифт заголовка столбца меняется с помощью свойства **Font**, свойство **Title.Caption** позволяют изменить текст заголовка. Например, **OrgName** можно заменить на

**Название организации.** Ширина поля меняется с помощью свойства **Width**.

Поле можно скрыть, установив свойство **Visible в false**.

Измените названия всех полей в сетке на русские, скройте поле **ClientNum**.