

## Тема 21. Программная обработка таблиц (практика)

Основные методы компонента **Table**:

- **Open** - открытие соединения с базой данных;
- **Close** - закрытие соединения с базой данных.

Эти методы аналогичны работе со свойством **Active: true** -установить соединение, **false** - разорвать соединение. Мы сразу устанавливали его в **true** при работе над проектом. Это не очень хороший вариант, поскольку занимается база данных. Если вы работаете не в сети или если вы единственный пользователь этой базы данных, то это, конечно, не имеет особого значения.

- **First**- перемещение к первой записи;
- **Last** - перемещение к последней записи;
- **Next** - перемещение к следующей записи;
- **Prior** - перемещение к предыдущей записи;
- **Edit**- перевод в режим редактирования текущей записи;
- **Insert** - вставка новой пустой записи;
- **Append** -добавление новой пустой записи в конец таблицы;
- **AppendRecord()** - добавляет запись в конец таблицы и заполняет необходимыми значениями;
- **Delete** - удаление текущей записи;
- **Post** - пересылка записи в базу данных.

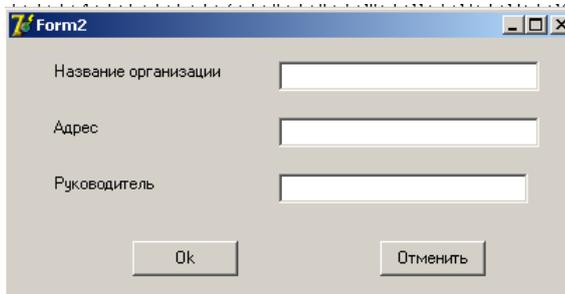
Пока идет редактирование текущей записи, изменения осуществляются в буфере, а не в самой базе данных. Пересылка записи в базу данных производится только при выполнении метода **Post**.

- **EOF** - логическая функция конца таблицы;
- **FieldByName** - доступ к заданным полям текущей записи (альтернативно можно использовать свойство **FieldValues**);

Включим в наше приложение возможность добавления записей.

1. Отключите возможность редактирования непосредственно в **DBGrid** (свойство **ReadOnly**).

2. Создайте новую форму для ввода данных в таблицу **КЛИЕНТЫ**. На этой форме разместите три компонента редактора, подписи к ним, кнопку ввода и отмены ввода.



3. Для кнопки **Отменить** установите **ModalResult** в **mrCancel**, а для кнопки **ОК** - в **mrOK**.

4. Добавьте на главную форму кнопку вызова (**Новая организация**) этой новой формы. Создайте для нее обработчик события примерно со следующим кодом (имена компонентов у вас могут быть другими):

```
if Form2.ShowModal=mrOK then  
begin  
  Table1.Append;  
  Table1['OrgName']:=Form2.Edit1.Text;  
  Table1['Addr']:=Form2.Edit2.Text;  
  Table1['Chief']:=Form2.Edit3.Text;  
  Table1.Post;  
end;
```

Обращение к значению поля текущей записи осуществляется следующим образом: **Table1['имя\_поля']**.

Другие варианты: **Table1.имя\_поля.Value** или  
**Table1.FieldName('имя\_поля').Value;**

Название организации	Адрес	Руководитель
Татэнерго	Москва	Вася
Идеал	Бугульма	Женя
Торис	Бавлы	Коля
Эльдорадо	Дербышки	Мустафа

Дата	Сумма
04.05.1999	6000

5. Создайте кнопку **Изменить** на главной форме для изменения данных организации и введите обработчик с кодом:

```
Form2.Edit1 Text:=Table1 ['OrgName'];  
Form2.Edit2Text:=Table1['Addr'];  
Form2.Edit3Text:=Table1['Chief'];  
if Form2.ShowModal=mrOK then  
begin  
    Table1.Edit;  
    Table1['OrgName']:=Form2.Edit1Text;  
    Table1['Addr']:=Form2.Edit2Text;  
    Table1['Chief']:=Form2.Edit3.Text;  
    Table1.Post;  
end;
```

6. Прodelайте то же самое для таблицы **ДОГОВОРА**: добавьте возможность ввода новых данных (даты и суммы) и изменения данных.

### **Задание. Вычисляемые поля**

Рассмотрим возможность формирования в таблице нового поля, не предусмотренного при ее создании, значение которого вычисляется на основании значений других полей записи. Подобные поля называются **вычисляемыми полями (calculated fields)**.

1. Будем использовать демонстрационную базу данных **DBDemos**, входящую в комплект **Delphi**. Разместите компоненты **Table**, **DataSource**, **DBGrid** и настройте их на таблицу **Country** этой БД (эта таблица содержит данные по некоторым странам мира: столица, площадь, население,...).

2. Вызовите редактор полей **Field Editor**: двойной щелчок левой кнопкой мыши на компоненте **Table** (либо правой с последующим выбором из меню). Затем щелчок правой кнопкой мыши внутри окна **Field Editor** и по команде **Add All Fields** создайте объекты для всех полей таблицы. В принципе список полей вы можете составить самостоятельно, добавляя отдельные поля с помощью команды **Add Fields** или удаляя ненужные командой **Delete**, но нужно иметь в виду, что в дальнейшем вы сможете обращаться только к тем полям, которые есть в этом списке. Выбрав поле-объект, можно настроить его свойства. Например, **Size (размер)** для текстовых полей или **Precision (точность)** для числовых полей, выравнивание внутри колонки **Alignment**, заголовок столбца **DisplayLabel**, ширина колонки **DisplayWidth**. Настройте заголовки всех столбцов на русские названия.

3. Создадим новое поле, отсутствующее в исходной таблице **Country**, в котором будет отображаться плотность населения. Правый щелчок внутри окна **Fields Editor**, команда **New Field** - появится диалоговое окно для создания нового поля. Задать имя поля (например, **Plotnost**), тип поля **Float** и установить переключатель **Calculated**.

4. Закройте **Fields Editor** и установите свойство **Active** компонента **Table** в **true**, после чего в сетке появится пустой пока столбец **Plotnost**.

5. Теперь укажем приложению, как надо вычислять это поле. Для компонента **Table** перейдите на вкладку **Events** и создайте обработчик события **OnCalcFields** с кодом:

**Table1.Plotnost.Value:= Table1.Population.Value / Table1.Area.Value**

6. Еще одна интересная особенность редактора полей: возможность перетаскивать из него поля на форму с помощью мышки. На форме при этом автоматически будут созданы компоненты, отображающие данные каждого поля и снабженные соответствующими метками. Разместите их должным образом.

7. Добавьте навигатор и настройте на таблицу. Установите свойство **Readonly** таблицы в **true**.