Структура проекта:

Login.pas– Файл содержащий в себе форму и логику авторизации пользователей в системе

Main.pas– Файл содержащий главное окно навигации в приложении. Содержит форму и логику переходов на другие формы.

KreditProductList.pas– Файл содержащий табличный вариант отображения описания существующих кредитный продуктов.

KreditDogovorList.pas– Файл, содержащий экранную форму и логику табличного отображения информации о кредитных договорах.

CreditDogovorEdit.pas– Файл, содержащий форму для редактирования и просмотра информации в карточном виде о создаваемых или уже существующих кредитных договорах.

PaymentEdit.pas– Файл, содержащий форму для создания, редактирования и просмотра информации в карточном виде о платежах.

ClientList.pas– Файл, содержащий экранную форму и логику табличного отображения информации о клиентах кредитной организации.

ClientEdit.pas– Файл, содержащий экранную форму и логику табличного отображения информации о клиентах кредитной организации.

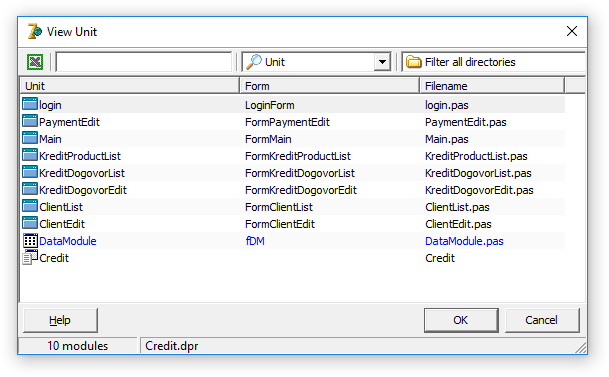


Рисунок. 3.2.1. Структура модулей

Рисунок. 3.2.2. Структура взаимосвязей модулей

Единственной не визуальной формой в разработанном программном обеспечении является DataModule. Она представляет собой форму-компонент для размещения на нем компонентов,которые могу использоваться одновременно другими формами. В DataModuleобычно размещаются компоненты для связи с базой данных. Рассмотрим подробнее этот компонент.

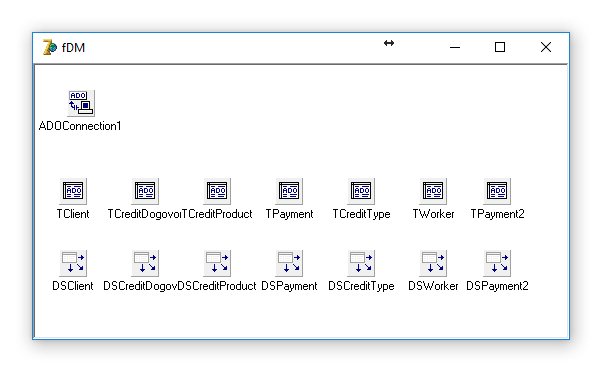


Рисунок. 3.2.2. Data Module

Основным компонентом для соединения с СУБД MSSQL 2008 в DelphiявляетсяADOConnection. Свойство компонента строка соединения формируется автоматически. В качестве поставщика данных используется «MicrosoftOLEDBProviderforSQLServer». В наименовании сервера указывается именованный экземпляр, созданный при установке MSSQLSQLServer, либо сервер, используемый по умолчания на локальной машине – (local). Авторизация на сервере выполняется средствами операционной системы. Название базы данных на сервере «CreditOrganisation».

После установления соединения с сервером базы данных необходимо настроить компоненты для отображения информации получаемой и базы данных.

Компонент, который отвечает за получение данных из БД является ADOTable. Она получает из бд полную копию таблицы и хранит ее на всем протяжении работы приложения.

DBGridявляется конечным компонентом для отображение информации из набора данных. Однако для его соединения с набором данных необходима связь. Эту задачу решает компонент DataSource, он является посредником между наборами данных и конечными визуальными компонентами. Все конечные визуальные компоненты содержат свойство DataSource.

Что бы настроить корректное отображение информации в таблице достаточно в свойстве DataSource указать необходимый набор данных.

В качестве компонентов для отображения информации из базы данных использовались: DBGrid, DBEdit,DBLookupComboBox. В этих компонентах уже заложена логика отображения информации из базы данных, достаточно только указать им нужный источник данных. Для добавления новых записей в набор данных с данными компонентами используются такие функции как Insert/Addend, для редактирования записи Editи для сохранения изменений Post.

Этих компонентов достаточно для работы с базой данных, однако, когда сложность задач увеличивается и требуется работать одновременно с несколькими таблицами необходимо использовать компоненты ADOQuery.В отличие от ADOTable, запрос не содержит свойств для выбора таблицы, которую он будет отображать. Вместо этого он имеет поле SQL, которое должно содержать текст SQL-запроса к БД для выбора необходимых данных. Основные процедуры:

Модуль «Login»:

Авторизация предназначена для идентификации пользователя в системе. После ввода пользователем логина и пароля и нажатия на кнопку «Войти» процедура описанная ниже передаст в SQLзапрос компонента ADOQuery параметры loginи password. После этого запросы выполняется, и в случае если запрос вернул запись, то авторотационные данные пользователя сохраняются на все время сессии, пока пользователь не завершит приложении либо не зайдет под другой учетной записью. Если пользователь ввел некорректные данные, то после нажатия на кнопку «Войти» он увидит соответственное сообщение.

procedureTLoginForm.LogInButtonClick(Sender: TObject);

begin

ADOQuery1.Close;

ADOQuery1.Parameters.ParamByName('login').Value:=Edit1.Text;

ADOQuery1.Parameters.ParamByName('password').Value:=Edit2.Text;

ADOQuery1.Open;

if (ADOQuery1.RecordCount>0) then

begin

LogUser:=ADOQuery1.Fields[0].AsString+'.'+ADOQuery1.Fields[1].AsString[1]+'.'+ADOQuery1.Fields[2].AsString[1];

FormMain.Show;

hide;

end

else

ShowMessage('Неверныйлогинилипароль');

end;

SQLскрипт для авторизации:

SELECT Фамилия, Имя, Отчество FROM Сотрудники WHERE login = :login and password = :password

Следующей процедурой-обработчиком в данном модулей является обработчик нажатия на клавишу Enter

procedure TLoginForm.Edit2KeyPress(Sender: TObject; var Key: Char);

begin

if (Key = #13) then LogInButtonClick(LogInButton);

end;

Модуль «Main» содержит в себе обработчики переходов на другие формы.Формы которые открываются для редактирования данных содержа в себе код для открытия набора данных на редактирование, например:

procedureTFormMain.CreateLoanAgreement(Sender: TObject);

begin

FormKreditDogovorEdit.Show;

FormKreditDogovorList.DBGrid1.DataSource.DataSet.Insert;

end;

Модуль «KreditDogovorList»

Добавлениеплатежа. В данном обработчике происходит открытие набора данных TPayment2 на добавление новой записи и переход на форму с добавлением нового платежа.

procedureTFormKreditDogovorList.AddPaymentClick(Sender: TObject);

begin

fDM.TPayment2.Insert;

fDM.TPayment2.FieldByName('Кредит').Value:= DBGrid1.DataSource.DataSet.FieldByName('№ договора').Value;

fDM.TPayment2.FieldByName('Сумма').AsFloat:=fDM.TCreditDogovor.FieldByName('размер\_платежа').AsFloat;

FormPaymentEdit.ShowModal;

end;

После перехода на форму Добавления нового платежа автоматически будут заполнены поля номер договора и размер платежа. После того как пользователь введет остальные донные и нажет на сохранить выполниться следующий обработчик:

procedureTFormPaymentEdit.SaveClick(Sender: TObject);

begin

fDM.TPayment2.FieldByName('Дата').AsDateTime:= DateTimePicker1.DateTime;

fDM.TPayment2.Post;

Close;

end;

SQLскрипт для добавления нового договора:

INSERT INTO dbo.LoanAgreement

(ID ,Client\_ID ,Credit\_ID ,Amount ,Term ,DateCreate ,Employee\_ID)

VALUES( 0, 12, 7, 90000.0, 5, 3 );

Работа с остальными модулями организована аналогичным способом, а SQLскрипты для добавления, редактирования и удаления информации приведены в приложении 1