САБИТОВ Б.Р., АЛМАСБЕКОВА З.
Кыргызский Национальный Университет им.Ж.Баласагына
УДК.004.42

РАЗРАБОТКА ТОНКОГО КЛИЕНТА ПРИЛОЖЕНИЯ В СРЕДЕ EMBARCADERORADSTUDIOXE2 . ПРОГРАММИРОВАНИЕ МОДУЛЯ ЧИТАТЕЛЯ

Создается сетевая информационная система на базе технологий EmbarcaderoRADStudio, MSSQLSERVER 2008 R2 и системы проектирования базы данных ER/Studio . Подробно изучено особенности построения сетевых приложений в данных системах.

EMBARCADERORADSTUDIOXE2 ЧӨЙРӨСҮНДӨ ИЧКЕ КЛИЕНТ ТИРКЕМЕСИН ТУРГУЗУУ. ОКУУЧУНУН МОДУЛУН ПРОГРАММАЛОО.

Embarcadero RAD Studio, MSSQLSERVER 2008 R2технологияларынын жана берилиштер базасын проектилоочу ER/Studio системасынын негизинде тармактык маалымат системасы тургузулат.

DEVELOPMENTOFATHINCLIENTAPPLICATIONINENVIRONMENTEMBARCADERORADSTUDIOXE2. PROGRAMMING MODULE READER

Create a network-based information system technologies Embarcadero RAD Studio, MS SQL SERVER 2008 R2 and System Database Design ER / Studio. Studied in detail the characteristics of building networked applications in these systems.

Работа является продолжением работы [1]. Она предназначен для читателей библиотеки ,которые смогут с помощью него просматривать свою статистику по абонементу, а также узнавать какие книги поступили и есть в настоящее время в библиотеке.

Итак создаем новое приложение в среде EmbarcaderoRADStudioXE2, форму называем ReaderModuleForm, В ее свойство Caption пишем "Модуль читателя". Сохраняем форму под именемUReaderModule, а проект под именем ReaderMOdUle_proj.

Размещаем на форме три компонента Buttonc текстом "Информация", "Поиск в библиотеке" и "Выход" (Рис. 1).

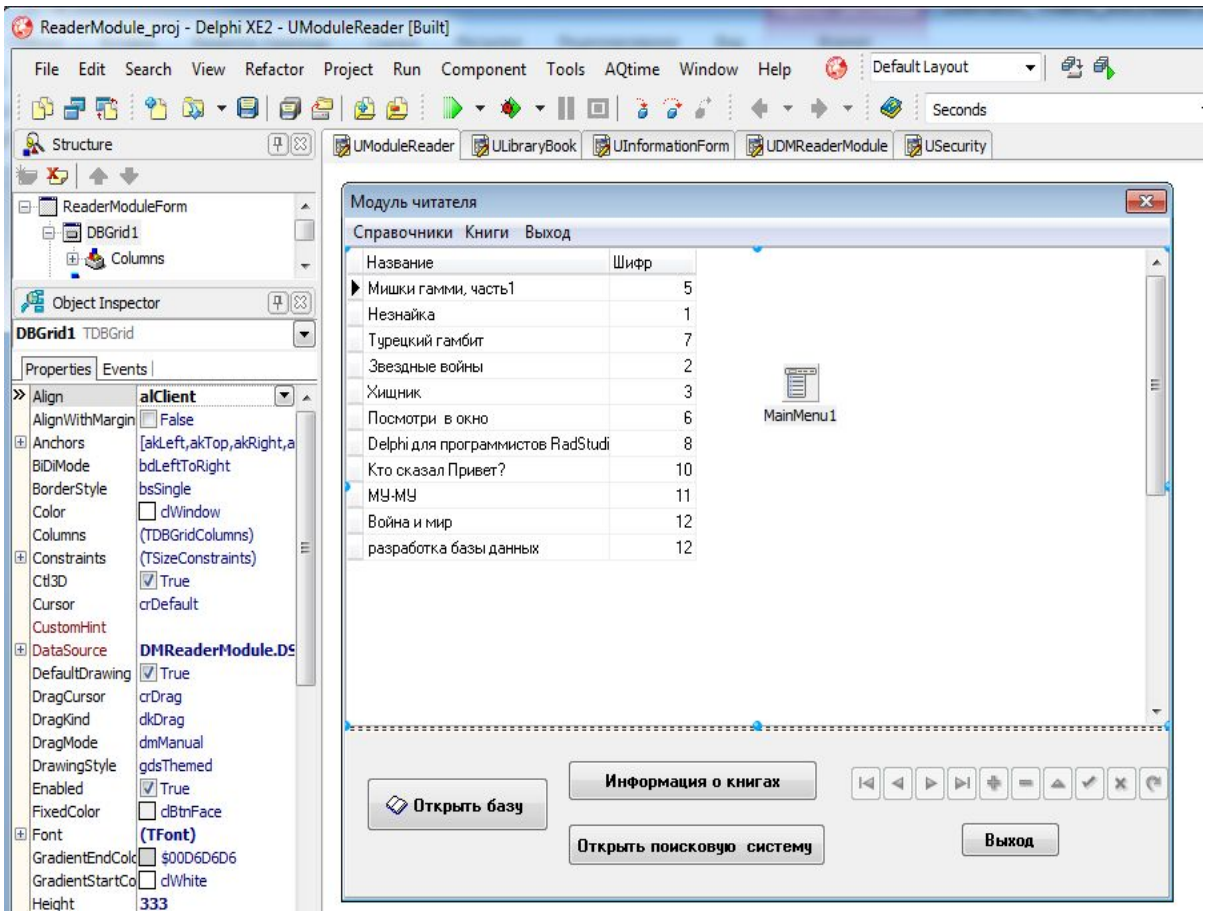

Рис. 1. Вид форм Модуль читателя в среде EmbarcaderoRADStudioXE2.

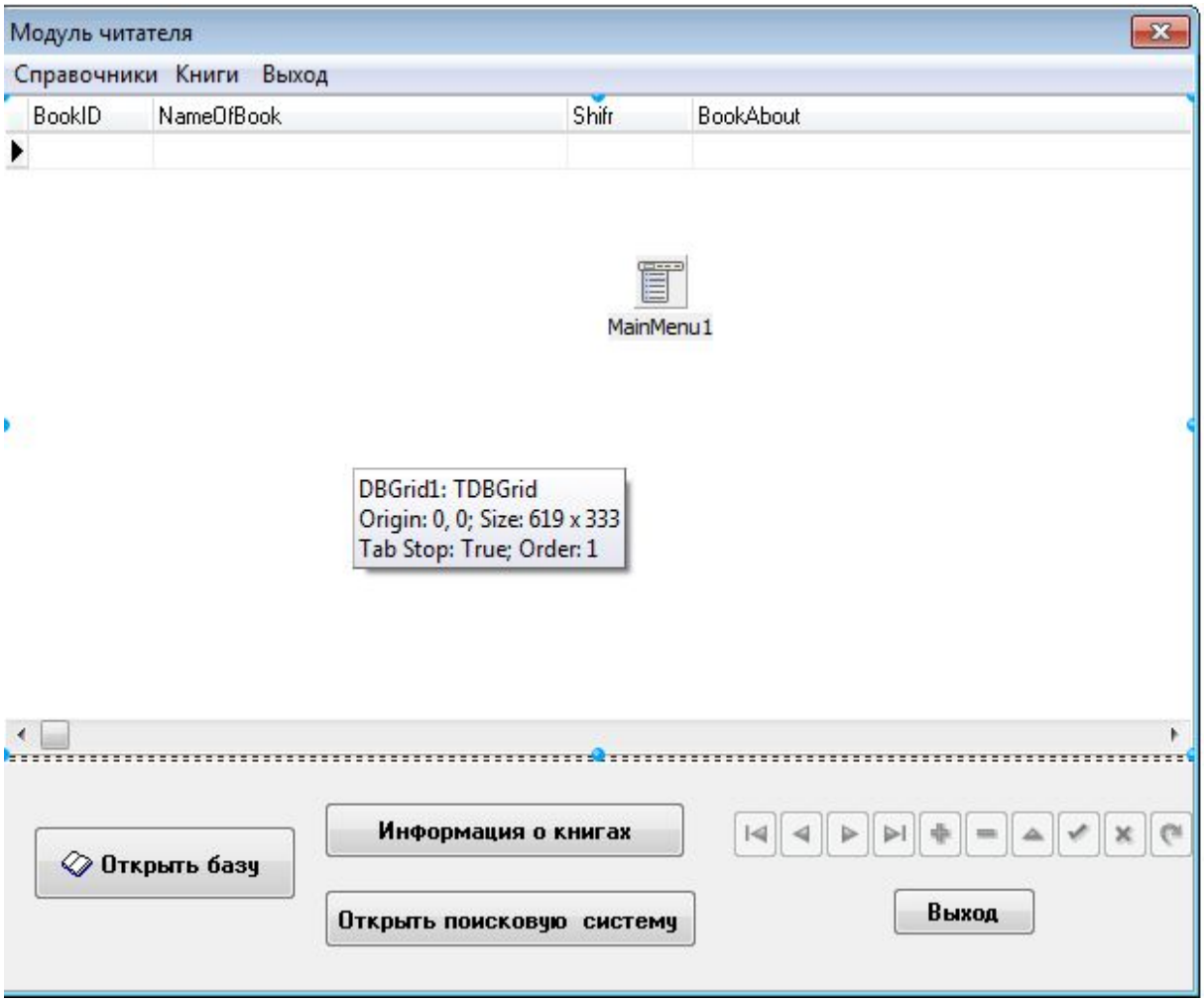

Рис. 2. Вид форм Модуль читателя

Для кнопки Выход пишем код выхода из приложения: Application.Terminate;

Создаем DataModule, называем его DMReaderModule, сохраняем под именем UDMReaderModule. Располагаем на нем компонент IBDatabase, называем его DBLibrary, настраиваем на базу LIBRARY.FDB.

Далее располагаем компонент ADOConnection1, который связываем с DBLibraryИ 2 штуки AdoDataSet. Называем их QBookHistory, QBookInLibrary. Для QBookHistoryB свойстве SQL пишем запрос для выбора всех книг определенного читателя, у которого они находятся на руках:

```
SELECT * FROM BOOK, MOVE
WHERE
(Move.BookKod=Book.BookID)        AND
(Move.AbonementKod=:Par)   AND
(Move.PriznakOfReturn='O')
```

Для QBookInLibraryB свойстве SQL пишем запрос для выбора всех книг, которые находятся в библиотеке:

```
SELECT * FROM BOOK
WHERE OnHand=' 0 ,
```

Располагаем 2 компонентаDataSource, называемихDSQBookHistoryиDSBookInLibrary, связываемихсодноименнымикомпонентамиAdoDataSet.

Нам надо, чтобы пользователь проходил аугентификацию перед входом в программу. Этим и займемся сейчас. Создаем новую форму, называем ее FormSecurity. Сохраняем под именем USecurity, удаляем ее из списка автоматически создаваемых форм.

Подключаем модули UDMReaderModule и UReaderModule. Помещаем на форму 2 штуки Label, 2 штуки Edit, IBQuery, 2 Button.AdoQuery1переименовываем в *QSecuri*ty, настраиваем его на DBLibrary. В свойстве SQL пишем запрос проверки данных, введенных пользователем, для аудентификации:

```
SELECT * FROM READER, ABONEMENT
WHERE(Reader.FIO=:ParFio)
AND
(Abonement.AbonementID=:ParAbonementID)
AND
(Abonement.ReaderKod=Reader._ReaderID)
```

Данный запрос осуществляет проверку правильности введенных данных пользователем; если все верно, то запрос вернет запись, иначе - количество возращенных записей будет нулевым. Необходимо создать объекты-столбцы в редакторе столбцов.

для кнопки **Вход** пишем код:

```
procedure TFormSecurity.Button1Click(Sender: TObject);
begin
{Задаем значение параметров, для запроса,
первые - логин, второй - пароль}
withQSecurity.Paramsdo
begin
ParamByName('ParFIO').Value:=Edit1.Text;
ParamByName('ParAbonementID').Value:=StrToInt(Edit2.Text);
end;
{Выполняемзапрос}
QSecurity.Open;
{Если количество возращенных данных запросом больше 0
значит, пользователь ввел правильные данные, количество
возвращенных записей хранится в свойстве RecordCount}
if  QSecurity.RecordCount>0 then
begin
{Запоминаемкодпользователя}
id_user:=QSecurityREADERID.Value;
{Запоминаемфамилиюпользователя}
FIO_user:=QSecurityFIO.Value;
{Запоминаемкодпользователя}
id_abonement:=QSecurityABONEMENTID.Value;

{Устанавливаемпризнакпрохожденияаутентификации}
CheckSecurity:=true;
QSecurity.Close;
{Закрываемокноаутентификации}
FormSecurity.Close;
{Выходим из процедуры, т.к. дальше обрабатывать нечего}
exit;
end;
{Если пользователь ввел не правильные данные,
то проинформировать его об этом}
ShowMessage('В доступе отказано');
{Закрываем набор данных}
QSecurity.Close;
end;
```

Для кнопки **Отмена**пишем код закрытия формы FormSecurity: FormSecurity.Close;

в модуле UModuleReader объявляем глобальные переменные:

```
var
ReaderModuleForm: TReaderModuleForm;
CheckSecurity:boolean=false;
{Данныепользователя,код, ФИО, кодабонемента}
id_user:integer=0;
FIO_user:string='';
id_abonement:integer=0;
```

Переменная CheckSecurity будет отвечать за то, прошел ли пользователь аутентификацию или нет.

Создаем новую форму, называем ее InformationForm. В свойство Caption пишем "Информация". Сохраняем ее под именем UlnformationForm, удаляем ее из списка автоматически создаваемых форм. Подключаем к ней модуль UDMReaderModule. DБGrid связываем с DSBookHistory.

Создаемновуюформу, называемееLibraryBookForm, ВсвойствоCaptionпишем "Поисккнигвбиблиотеке". Сохраняем ее под именем ULibraryBook, удаляем ее из списка автоматически создаваемых форм. Подключаем к ней модуль UDMReaderModule.

DБGrid связываем с OSBookInLibrary.

Для компонента Edit в событии OnChange пишем следующий код поиска книги по первым введенным символам:

```
DMReaderModule.QBookInLibrary.
Locate ('NarneOfBook', Editl. Text, [loCaseInsensitive, loPartialKey]) ;
```

НочтобыEmbarcaderoRADStudioXE2 неругаласьнаэтоткод, нужноподключитьмодульDB, потомучтоконстантыloCaseInsensitive, loPartialKeуописаныименновнем. Теперь при вводе в Edit данных с клавиатуры автоматически будет происходить поиск названия книги в наборе данных QBookInLibrary.

Переходим к форме ReaderModuleForm и для кнопки Поиск в библиотеке пишем код вызова окна Поиск в библиотеке

```
procedure TReaderModuleForm.Button2Click(Sender: TObject);
begin
LibraryBookForm:=TLibraryBookForm.Create(self);
DMReaderModule.QBookInLibrary.Open;
LibraryBookForm.ShowModal;
end;
```

Литература:

1. СабитовБ.Р. , АлмазбековаЗ. Разработка тонкого клиента приложения в среде EmbarcaderoRADStudioXE2 .Программирование администратора. Настоящий сборник.
2. .ФароновВ.А Система программированияDelphi 2007 БХВ,СПб.2003г.
3. А.Я. Архангельский Delphi 2007СПб.:Питер, 2006 г
4. ФленовМ. “БиблияDelphi” Москва 2008г.
5. Марко КэнтуDelphi 2009 . Для профессионалов. – М.: SYBEX, ПИТЕР 2010. – 1100 с.