**ПРОГРАММНЫЙ ПРОДУКТ**

**«ХОСТЕЛ»**

**РУКОВОДСТВО ПРОГРАММИСТА**

2022

Оглавление

[**Введение** 3](#_Toc119346717)

[**Описание экранных форм приложения** 4](#_Toc119346718)

[**Приложение А Листинг программы** 11](#_Toc119346719)

**Введение**

Руководство предназначено для программистов, работающих с программным продуктом "Хостел". В руководстве содержатся сведения, необходимые для пользования и дальнейшего развития приложения.

**Описание экранных форм приложения**

Код программы представлен в Приложении А.

Первая страница приложения включает компоненты:

* Label – текстовое поле;
* MaskEdit – поле для ввода даты заезда, даты выезда;
* Button – кнопка для перехода к следующим страницам.

Функции Главной станицы:

* - Переход к окнам Регистрации и Адмнистратор;
* - Ввод необходимых данных для поиска подходящей комнаты.

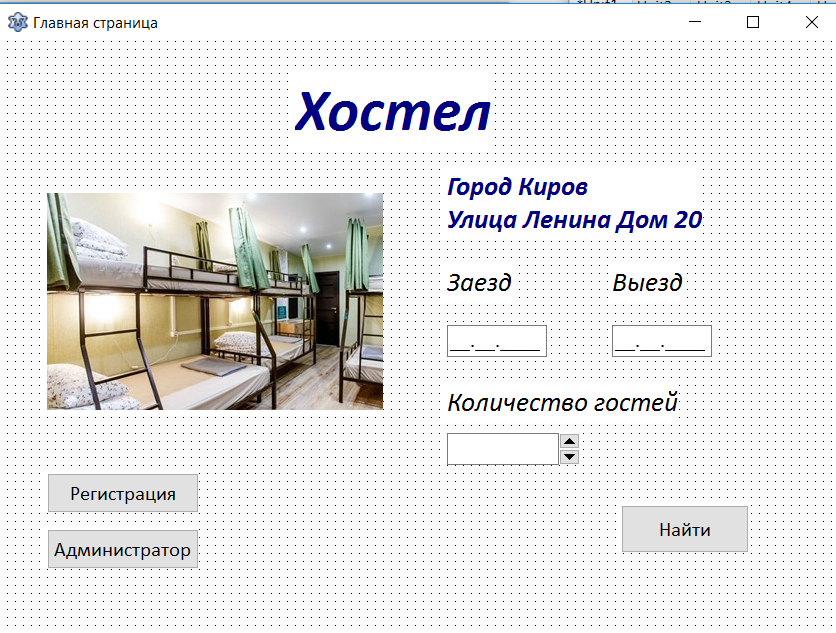


Рисунок 1 – Главная страница

На 2 странице происходит внесение данных клиента, которые при нажатии кнопки «Сохранить» фиксируются в базе данных. При нажатии на кнопку «Главная страница» клиент может вернуться на главную страницу.

Компоненты:

- Обеспечивающие связь формы Lazarus и базы данных Access: TDataSource, TODBCConection, SQLTransaction и SQLQuery.

* Label – текстовое поле;
* Button – кнопка для перехода к следующим страницам.

Процедура занесения данных таблицы в базу описана в Приложении 1 – Unit2.

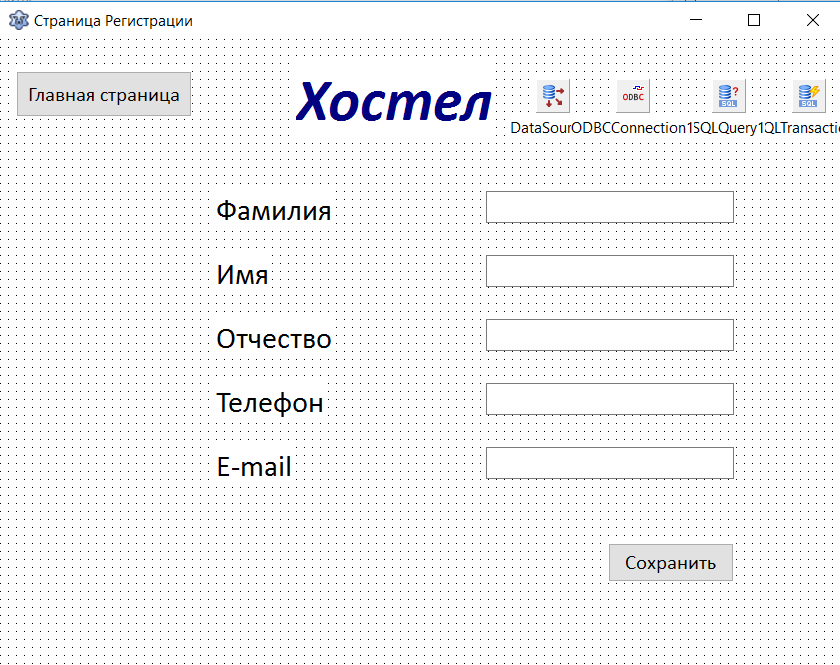


Рисунок 2 – Страница регистрации

Далее пользователь переходит к выбору комнаты. В окне приложения, пользователю необходимо нажать кнопку «Просмотреть» для того, чтобы вывести введенные им данные ранее (Дата заезда/выезда и количество гостей). Для оформления бронирования пользователю нужно ввести фамилию, при условии того, что ранее клиент был зарегистрирован. После выполнения всех операций пользователь нажимает на комнату, которая ему необходима. Выбранная комната будет забронирована. По поводу дальнейших действий (оплаты и наличия мест в выбранном номере) клиенту расскажет оператор.

Компоненты:

* Label – текстовое поле;
* MaskEdit – поле для ввода даты заезда, даты выезда;
* Button – кнопка для перехода к следующим страницам.
* - Обеспечивающие связь формы Lazarus и базы данных Access: TDataSource, TODBCConection, SQLTransaction и SQLQuery.

Процедуры страницы «Выбор комнаты» описаны в Приложении 1 – Unit3.

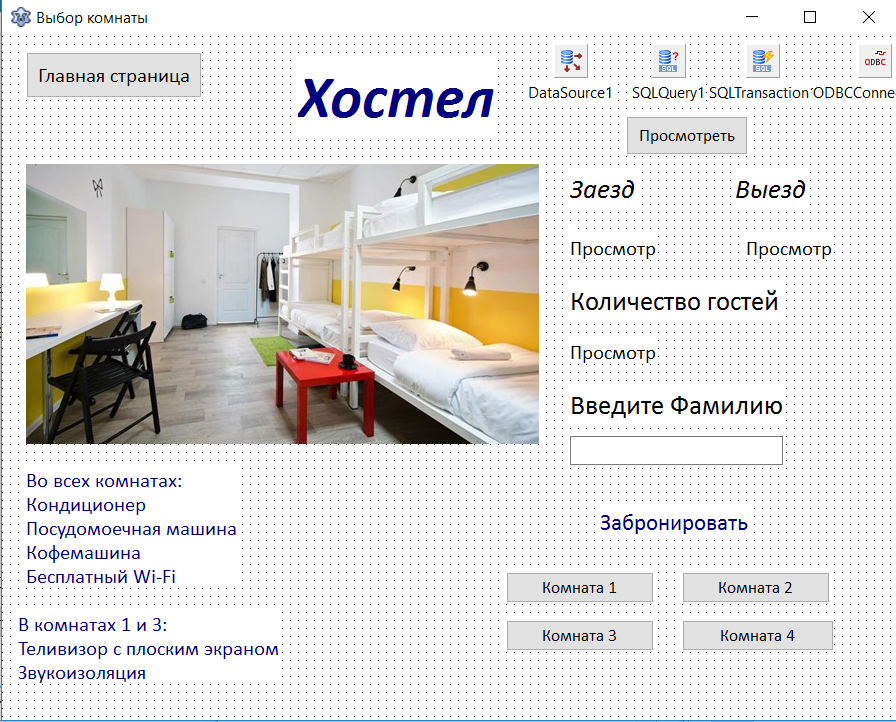


Рисунок 3 – Форма просмотра комнаты

Далее пользователь видит окно, которое подтверждает бронь.

Данные были зафиксированы в базу данных.

Компоненты:

* Label – текстовое поле;
* Button – кнопка для перехода к следующим страницам.

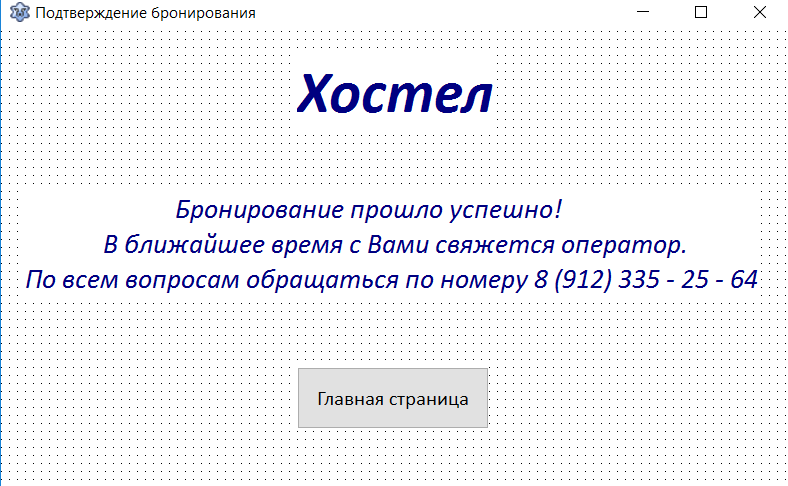


Рисунок 4 – Подтверждение бронирования

Для выполнения дополнительных действий необходимо с «Главной страницы» перейти по кнопке «Администратор».

Сотрудник, переходя в режим администратора, должен ввести логин и пароль, для дальнейшей работы.

Компоненты:

* Label – текстовое поле;
* Button – кнопка для перехода к следующим страницам.

Процедуры страницы «Вход для сотрудников» описаны в Приложении 1 – Unit5.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 5 –Форма входа для сотрудников

После верного ввода логина («admin») и пароля («1234»), пользователь получает доступ к данным о клиентах и комнатах.

При введении некорректного ввода выходит ошибка.

После нажатия кнопки «Просмотр клиентов», администратору выходит таблица, содержащая данные клиентов. Здесь администратор может осуществить поиск клиента по фамилии и отсортировать список, нажав на заголовок столбца фамилий.

Администратор может просмотреть количество зарегистрированных пользователей, нажав кнопку «Просмотр статистики».

Компоненты страницы:

- Обеспечивающие связь формы Lazarus и базы данных Access: TDBGrid, TDataSource, TODBCConection, SQLTransaction и SQLQuery.

* Label – текстовое поле;
* Button – кнопка для перехода к следующим страницам.

Функции кода прописаны в Приложении 1

Процедуры страницы «Просмотр клиентов» описаны в Приложении 1 – Unit6.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 6 – Форма просмотра клиентов

Окно «Статистика» отображает количество зарегистрированных на данный момент пользователей.

Компоненты страницы:

* - Обеспечивающие связь формы Lazarus и базы данных Access: TDBGrid;
* Label – текстовое поле;
* Button – кнопка для перехода к следующим страницам.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 7 – Статистика

Переходя к просмотру комнат, администратор видит базу, содержащую информацию о них. Здесь сотрудник может внести новую запись или удалить существующую.

Компоненты страницы:

- Обеспечивающие связь формы Lazarus и базы данных Access: TDBGrid, TDataSource, TODBCConection, SQLTransaction и SQLQuery.

* Label – текстовое поле;
* Button – кнопка для перехода к следующим страницам.

Процедуры страницы «Просмотр комнат» описаны в Приложении 1 – Unit7.

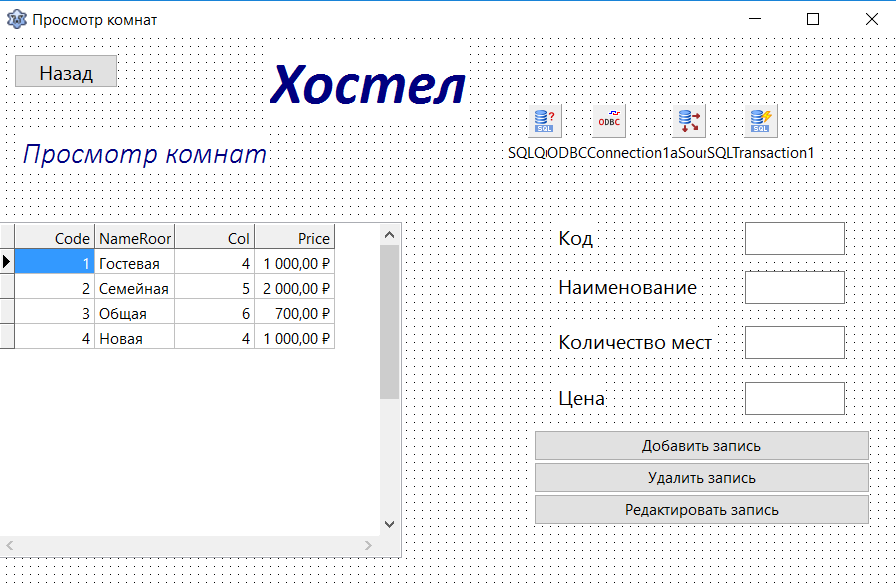


Рисунок 8 – Форма просмотра комнат

**Приложение А  
 Листинг программы**

unit Unit1;

procedure Tform1.Button2Click(Sender: Tobject);

begin

Form2.show;

Form1.Hide;

end;

procedure Tform1.Button3Click(Sender: Tobject);

begin

Form6.Show;

Form1.Hide;

end;

procedure Tform1.Button1Click(Sender: Tobject);

begin

/val(Edit1.Text,s);

Form3.Show;

Form1.Hide;

end;

Unit Unit2;

procedure Tform2.Button1Click(Sender: Tobject);

begin

Form1.Show;

Form2.Hide;

end;

procedure Tform2.Button2Click(Sender: Tobject);

begin

if (edit1.Text<>’’) and (edit2.Text<>’’) and (edit3.Text<>’’) and (edit4.Text<>’’) and (edit5.Text<>’’) then

begin

SQLQuery1.Close; // выключаем компонент

SQLQuery1.SQL.Text := ‘insert into client (Surname,Name,MiddleName,PhoneNumber,Email) values(:p,:u,:r,:t,:z);’; // добавляем sql запрос для добавления данных

SQLQuery1.ParamByName(‘p’).AsString := edit1.Text;// присваиваем записи текстовое значение

SQLQuery1.ParamByName(‘u’).AsString := edit2.Text;

SQLQuery1.ParamByName(‘r’).AsString := edit3.Text;

SQLQuery1.ParamByName(‘t’).AsString := edit4.Text;

SQLQuery1.ParamByName(‘z’).AsString := edit5.Text;

SQLQuery1.ExecSQL; // выполняем запрос

SQLTransaction1.Commit; //подтверждаем изменения в базе

Form1.Show;

Form2.Hide;

end

else

showmessage('Заполнены не все поля! Запись не может быть добавлена в базу данных!');

end;

{

begin

// проверить заполнены ли все поля

if (edit1.Text<>’’) and (edit2.Text<>’’) and (edit3.Text<>’’) and (edit4.Text<>’’) and (edit5.Text<>’’) then

begin

SQLquery1.active:=false;

SQLQuery1.SQL.Text:=’INSERT INTO client (Surname,Name,MiddleName,PhoneNumber,Email) VALUES (“’+edit1.text+’”,”’+edit2.text+’”, “’+edit3.text+’”, ‘+edit4.Text+’,”’+edit5.text+’)’;

SQLQuery1.ExecSQL;

SQLQuery1.close;

SQLQuery1.SQL.Text:=’select \* from сотрудник’;

SQLQuery1.Open;

form2.Close; // закрыть форму ввода новой записи

end

else

showmessage('Заполнены не все поля! Запись не может быть добавлена в базу данных!');

end;}

procedure Tform2.Edit1Change(Sender: Tobject);

begin

end;

end.

Unit Unit3;

procedure TForm3.Button1Click(Sender: TObject);

begin

Form1.Show;

Form3.Hide;

end;

procedure TForm3.Button2Click(Sender: TObject);

begin

Form4.Show;

Form3.Hide;

end;

procedure TForm3.Button3Click(Sender: TObject);

begin

Label7.Caption := Form1.MaskEdit1.Text;

Label8.Caption := Form1.MaskEdit2.Text;

Label9.Caption := Form1.Edit1.Text;

end;

procedure TForm3.Button4Click(Sender: TObject);

begin

// проверить заполнены ли все поля

if (edit1.Text<>'') then

begin

SQLQuery1.Close; // выключаем компонент

SQLQuery1.SQL.Text := 'insert into DistributionOfRooms(Room,Client,StartDate,EndDate,People) VALUES (:p,:u,:r,:o,:t)'; // добавляем sql запрос для добавления данных

SQLQuery1.ParamByName('p').AsString := '1';// присваиваем записи текстовое значение

SQLQuery1.ParamByName('u').AsString := Edit1.Text;

SQLQuery1.ParamByName('r').AsString := Label7.Caption;

SQLQuery1.ParamByName('o').AsString := Label8.Caption;

SQLQuery1.ParamByName('t').AsString := Label9.Caption;

SQLQuery1.ExecSQL;

//SQLQuery1.execSQL; // выполняем запрос

SQLTransaction1.Commit; //подтверждаем изменения в базе

Form4.Show;

Form3.Hide;

end

else

showmessage('Заполните поле "Фамилия"');

end;

procedure TForm3.Button5Click(Sender: TObject);

begin

// проверить заполнены ли все поля

if (edit1.Text<>'') then

begin

SQLQuery1.Close; // выключаем компонент

SQLQuery1.SQL.Text := 'insert into DistributionOfRooms(Room,Client) VALUES (:p,:u)'; // добавляем sql запрос для добавления данных

SQLQuery1.ParamByName('p').AsString := '2';// присваиваем записи текстовое значение

SQLQuery1.ParamByName('u').AsString := Edit1.Text;

SQLQuery1.ExecSQL;

//SQLQuery1.execSQL; // выполняем запрос

SQLTransaction1.Commit; //подтверждаем изменения в базе

Form4.Show;

Form3.Hide;

end

else

showmessage('Заполните поле "Фамилия"');

end;

procedure TForm3.Button6Click(Sender: TObject);

begin

// проверить заполнены ли все поля

if (edit1.Text<>'') then

begin

SQLQuery1.Close; // выключаем компонент

SQLQuery1.SQL.Text := 'insert into DistributionOfRooms(Room,Client) VALUES (:p,:u)'; // добавляем sql запрос для добавления данных

SQLQuery1.ParamByName('p').AsString := '3';// присваиваем записи текстовое значение

SQLQuery1.ParamByName('u').AsString := Edit1.Text;

SQLQuery1.ExecSQL;

//SQLQuery1.execSQL; // выполняем запрос

SQLTransaction1.Commit; //подтверждаем изменения в базе

Form4.Show;

Form3.Hide;

end

else

showmessage('Заполните поле "Фамилия"');

end;

procedure TForm3.Button7Click(Sender: TObject);

begin

// проверить заполнены ли все поля

if (edit1.Text<>'') then

begin

SQLQuery1.Close; // выключаем компонент

SQLQuery1.SQL.Text := 'insert into DistributionOfRooms(Room,Client) VALUES (:p,:u)'; // добавляем sql запрос для добавления данных

SQLQuery1.ParamByName('p').AsString := '4';// присваиваем записи текстовое значение

SQLQuery1.ParamByName('u').AsString := Edit1.Text;

SQLQuery1.ExecSQL;

//SQLQuery1.execSQL; // выполняем запрос

SQLTransaction1.Commit; //подтверждаем изменения в базе

Form4.Show;

Form3.Hide;

end

else

showmessage('Заполните поле "Фамилия"');

end;

Unit Unit5;

procedure TForm5.Button2Click(Sender: TObject);

var c,k: string;

begin

c:='admin';

k:='1234';

if (Edit1.Text=c) and (Edit2.Text=k) then

begin

Label5.Caption:='Данные введены верно';

Form7.Show;

Form5.Hide;

end

else Label5.Caption:='Введен неверный логин или пароль!';

end;

procedure TForm5.BitBtn1Click(Sender: TObject);

var c,k: string;

begin

c:='admin';

k:='1234';

if (Edit1.Text=c) and (Edit2.Text=k) then

begin

Label5.Caption:='Данные введены верно';

Form6.Show;

Form5.Hide;

end

else Label5.Caption:='Введен неверный логин или пароль!';

end;

Unit Unit6;

procedure TForm6.Button2Click(Sender: TObject);

var s1 :string;

begin

if edit1.Text<>'' then s1:='and(Surname="'+edit1.text+'")';

SQLQuery1.Close;// закрываем датасет

SQLQuery1.SQL.Text := 'select \* from client where Surname = :s1'; // добавляем наш запрос

SQLQuery1.ParamByName('s1').AsString:=edit1.text;// присваиваем требуемый параметр

SQLQuery1.Open;// открываем запрос

end;

procedure TForm6.Button3Click(Sender: TObject);

begin

Form8.Show;

Form6.Hide;

end;

procedure TForm6.DBGrid1TitleClick(Column: TColumn);

begin

SQLQuery1.Close;

SQLQuery1.SQL.Clear;

SQLQuery1.SQL.Add('SELECT \* FROM client ORDER BY 1 ASC');

SQLQuery1.Open;

end;

Unit Unit7;

procedure TForm7.Button3Click(Sender: TObject);

begin

Form5.Show;

Form7.Hide;

end;

procedure TForm7.Button4Click(Sender: TObject);

begin

SQLQuery1.Close; // выключаем компонент

SQLQuery1.SQL.Text := 'update room set Code:=cd, NameRoom:=nr, Col:=cl, Price:=pr, where id:=id';

// добавляем sql запрос для добавления данных

SQLQuery1.ParamByName('cd').AsString :=Edit1.Text;// присваиваем записи текстовое значение

SQLQuery1.ParamByName('nr').AsString :=Edit2.Text;

SQLQuery1.ParamByName('cl').AsString :=Edit3.Text;

SQLQuery1.ParamByName('pr').AsString :=Edit4.Text;

SQLQuery1.ParamByName('id').AsString :='600';

SQLQuery1.execSQL; // выполняем запрос

SQLTransaction1.Commit; //подтверждаем изменения в базе

end;

procedure TForm7.Button1Click(Sender: TObject);

begin

// проверить заполнены ли все поля

if (edit1.Text<>'') and (edit2.Text<>'') and (edit3.Text<>'') and (edit4.Text<>'') then

begin

SQLQuery1.Close; // выключаем компонент

SQLQuery1.SQL.Text := 'insert into room(Code,NameRoom,Col,Price) VALUES (:p,:u,:q,:h)'; // добавляем sql запрос для добавления данных

SQLQuery1.ParamByName('p').AsString := Edit1.Text;// присваиваем записи текстовое значение

SQLQuery1.ParamByName('u').AsString := Edit2.Text;

SQLQuery1.ParamByName('q').AsString := Edit3.Text;

SQLQuery1.ParamByName('h').AsString := Edit4.Text;

SQLQuery1.execSQL; // выполняем запрос

SQLTransaction1.Commit; //подтверждаем изменения в базе

showmessage ('Запись добавлена');

end

else

showmessage('Заполнены не все поля! Запись не может быть добавлена в базу данных!');

end;

procedure TForm7.Button2Click(Sender: TObject);

begin

SQLQuery1.Delete; // удаляем запись

end;.