**Занятие № 24**

**Номер учебной группы:** П-16.

**Фамилия, инициалы учащегося:** Пачко Н.Н.

**Дата выполнения работы:** 07.12.2022

**Тема работы:** «Интегрирование программных модулей»

**Ход работы**

**Задание 1**

Используя разработанные модули программного продукта, произвел интегрирование модулей системы.

**Задание 2**

Получил рабочий код программы, исправить ошибки.

**Листинг 1 – Модуль Unit1.pas**

Модуль главного окна программы

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, jpeg, ExtCtrls, Menus, ShellApi;

type

TForm1 = class(TForm)

Image1: TImage;

Button1: TButton;

Button2: TButton;

MainMenu1: TMainMenu;

N1: TMenuItem;

N2: TMenuItem;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure N1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

uses Unit2, Unit6;

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

begin

Form1.Close

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

Form1.Visible:=false;

Form2.Visible:=true;

end;

procedure TForm1.N2Click(Sender: TObject);

begin

Form1.Visible:=false;

Form6.Visible:=true;

end;

procedure TForm1.N1Click(Sender: TObject);

begin

ShellExecute(0,Pchar('Open'),Pchar('spravka.chm'),nil,nil,SW\_SHOW);

end;

end.

**Листинг 2 – Модуль Unit2.pas**

Модуль формы «обработка результатов экзамена»

unit Unit2;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, jpeg, ExtCtrls, StdCtrls;

type

TForm2 = class(TForm)

Image1: TImage;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses Unit1, Unit3, Unit5, Unit4;

{$R \*.dfm}

procedure TForm2.Button1Click(Sender: TObject);

begin

Form2.Visible:=false;

Form1.Showmodal;

end;

procedure TForm2.Button2Click(Sender: TObject);

begin

Form3.Showmodal;

end;

procedure TForm2.Button4Click(Sender: TObject);

begin

Form5.Showmodal;

end;

procedure TForm2.Button3Click(Sender: TObject);

begin

Form4.Showmodal;

end;

end.

**Листинг 3 – Модуль Unit3.pas**

Модуль формы «Математика»

unit Unit3;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, jpeg, ExtCtrls, Grids, DBGrids, DB, ADODB;

type

TForm3 = class(TForm)

Image1: TImage;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Label1: TLabel;

Label2: TLabel;

Edit1: TEdit;

Edit2: TEdit;

ADOConnection1: TADOConnection;

DataSource1: TDataSource;

DBGrid1: TDBGrid;

Edit3: TEdit;

ADOQuery1: TADOQuery;

Label3: TLabel;

procedure Button5Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

implementation

uses Unit2;

{$R \*.dfm}

procedure TForm3.Button5Click(Sender: TObject);

begin

Form3.Close;

end;

procedure TForm3.Button4Click(Sender: TObject);

var

k,sr:real;

i:integer;

begin

k:=0;

for i:=0 to ADOQuery1.RecordCount-1 do

begin

k:=k+ADOQuery1.FieldValues['Отметки'];

ADOQuery1.Next;

end;

sr:=(k/ADOQuery1.RecordCount);

Edit3.Text:=FloatToStr(sr);

end;

procedure TForm3.Button1Click(Sender: TObject);

var lpText,lpCaption : PChar; Tip: integer;

begin

if (Edit1.Text='') or (Edit2.Text='') then

begin

lpText := 'Заполните пропуски.';

lpCaption := 'Есть пустые поля.';

Tip := MB\_OK;

with Application do

MessageBox(lpText, lpCaption, Tip);

end

else

begin

ADOQuery1.Insert;

ADOQuery1['Абитуриенты']:=Edit1.Text;

ADOQuery1['Отметки']:=Edit2.Text;

ADOQuery1.Post;

end;

end;

procedure TForm3.Button2Click(Sender: TObject);

var lpText,lpCaption : PChar; Tip: integer;

begin

if (Edit1.Text='') and (Edit2.Text='') then

begin

lpText := 'Заполните пропуски.';

lpCaption := 'Есть пустые поля.';

Tip := MB\_OK;

with Application do

MessageBox(lpText, lpCaption, Tip);

end

else

begin

ADOQuery1.Edit;

if Edit1.Text<>'' then

ADOQuery1['Абитуриенты']:=Edit1.Text;

if Edit2.Text<>'' then

ADOQuery1['Отметки']:=Edit2.Text;

ADOQuery1.Post;

end;

end;

procedure TForm3.Button3Click(Sender: TObject);

begin

ADOQuery1.delete;

end;

end.

**Листинг 4 – Модуль Unit4.pas**

Модуль формы «Белорусский язык»

unit Unit4;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Grids, DBGrids, DB, ADODB, StdCtrls, jpeg, ExtCtrls;

type

TForm4 = class(TForm)

Image1: TImage;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button5: TButton;

Edit1: TEdit;

Edit2: TEdit;

DBGrid1: TDBGrid;

Button4: TButton;

ADOConnection1: TADOConnection;

ADOQuery1: TADOQuery;

DataSource1: TDataSource;

Image2: TImage;

Edit3: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

procedure Button5Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form4: TForm4;

implementation

{$R \*.dfm}

procedure TForm4.Button5Click(Sender: TObject);

begin

Form4.Close;

end;

procedure TForm4.Button4Click(Sender: TObject);

var

k,sr:real;

i:integer;

begin

k:=0;

for i:=0 to ADOQuery1.RecordCount-1 do

begin

k:=k+ADOQuery1.FieldValues['Отметки'];

ADOQuery1.Next;

end;

sr:=(k/ADOQuery1.RecordCount);

Edit3.Text:=FloatToStr(sr);

end;

procedure TForm4.Button3Click(Sender: TObject);

begin

ADOQuery1.delete;

end;

procedure TForm4.Button1Click(Sender: TObject);

var lpText,lpCaption : PChar; Tip: integer;

begin

if (Edit1.Text='') or (Edit2.Text='') then

begin

lpText := 'Заполните пропуски.';

lpCaption := 'Есть пустые поля.';

Tip := MB\_OK;

with Application do

MessageBox(lpText, lpCaption, Tip);

end

else

begin

ADOQuery1.Insert;

ADOQuery1['Абитуриенты']:=Edit1.Text;

ADOQuery1['Отметки']:=Edit2.Text;

ADOQuery1.Post;

end;

end;

procedure TForm4.Button2Click(Sender: TObject);

var lpText,lpCaption : PChar; Tip: integer;

begin

if (Edit1.Text='') and (Edit2.Text='') then

begin

lpText := 'Заполните пропуски.';

lpCaption := 'Есть пустые поля.';

Tip := MB\_OK;

with Application do

MessageBox(lpText, lpCaption, Tip);

end

else

begin

ADOQuery1.Edit;

if Edit1.Text<>'' then

ADOQuery1['Абитуриенты']:=Edit1.Text;

if Edit2.Text<>'' then

ADOQuery1['Отметки']:=Edit2.Text;

ADOQuery1.Post;

end;

end;

end.

**Листинг 5 – Модуль Unit5.pas**

Модуль формы «Русский язык»

unit Unit5;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Grids, DBGrids, DB, ADODB, StdCtrls, jpeg, ExtCtrls;

type

TForm5 = class(TForm)

Image1: TImage;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button5: TButton;

Label1: TLabel;

Label2: TLabel;

Edit1: TEdit;

Edit2: TEdit;

DBGrid1: TDBGrid;

Button4: TButton;

ADOConnection1: TADOConnection;

ADOQuery1: TADOQuery;

DataSource1: TDataSource;

Label3: TLabel;

Edit3: TEdit;

procedure Button5Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form5: TForm5;

implementation

{$R \*.dfm}

procedure TForm5.Button5Click(Sender: TObject);

begin

Form5.Close;

end;

procedure TForm5.Button4Click(Sender: TObject);

var

k,sr:real;

i:integer;

begin

k:=0;

for i:=0 to ADOQuery1.RecordCount-1 do

begin

k:=k+ADOQuery1.FieldValues['Отметки'];

ADOQuery1.Next;

end;

sr:=(k/ADOQuery1.RecordCount);

Edit3.Text:=FloatToStr(sr);

end;

procedure TForm5.Button1Click(Sender: TObject);

var lpText,lpCaption : PChar; Tip: integer;

begin

if (Edit1.Text='') or (Edit2.Text='') then

begin

lpText := 'Заполните пропуски.';

lpCaption := 'Есть пустые поля.';

Tip := MB\_OK;

with Application do

MessageBox(lpText, lpCaption, Tip);

end

else

begin

ADOQuery1.Insert;

ADOQuery1['Абитуриенты']:=Edit1.Text;

ADOQuery1['Отметки']:=Edit2.Text;

ADOQuery1.Post;

end;

end;

procedure TForm5.Button2Click(Sender: TObject);

var lpText,lpCaption : PChar; Tip: integer;

begin

if (Edit1.Text='') and (Edit2.Text='') then

begin

lpText := 'Заполните пропуски.';

lpCaption := 'Есть пустые поля.';

Tip := MB\_OK;

with Application do

MessageBox(lpText, lpCaption, Tip);

end

else

begin

ADOQuery1.Edit;

if Edit1.Text<>'' then

ADOQuery1['Абитуриенты']:=Edit1.Text;

if Edit2.Text<>'' then

ADOQuery1['Отметки']:=Edit2.Text;

ADOQuery1.Post;

end;

end;

procedure TForm5.Button3Click(Sender: TObject);

begin

ADOQuery1.delete;

end;

end.

**Листинг 6 – Модуль Unit6.pas**

Модуль формы «О программе»

unit Unit6;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls;

type

TForm6 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Button1: TButton;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form6: TForm6;

implementation

{$R \*.dfm}

procedure TForm6.Button1Click(Sender: TObject);

begin

Form6.Close;

end;

end.

**Задание 3**

Ответил на контрольные вопросы.

**1. Интегрирование модулей - это?**

Интегрирование модулей — это обмен данными между системами с возможной последующей их обработкой. Смысл интеграции заключается в том, чтобы данные, которые пользователь вводит в одну систему, автоматически переносились в другую.

**2. Опишите основные подходы интегрирования модулей.**

Модульное проектирование является одним из первых подходов к разработке структуры ПС и уже несколько десятилетий сохраняет свои позиции как в качестве классического подхода, так и в качестве основы для современных технологий разработки ПС.  
При разработке модульных ПС могут использоваться методы структурного проектирования или методы объектно-ориентированного проектирования. Их целью является формирование структуры создаваемой программы – ее разделение по некоторым установленным правилам на структурные компоненты (модуляризация) с последующей иерархической организацией данных компонентов. Для различных языков программирования такими компонентами могут быть подпрограммы, внешние модули, объекты и т.п.

**3. Что означает Верификация программного обеспечения?**

Верификацией называется проверка соответствия результатов отдельных этапов разработки программной системы требованиям и ограничениям, сформулированным для них на предыдущих этапах.

**4. Какие вы знаете инструменты интеграции?**

* ArcESB.
* CloverDX.
* Линкс
* Xplenty.
* Интегромат
* DataDeck