ЛЕКЦИЯ №7

**1. РАЗРАБОТКА DLL – БИБЛИОТЕК**

1. Язык программирования Delphi является языком визуального объектно-ориентированного программирования (ООП). В современных программах, совместно с ООП, очень часто используется и модульный принцип создания программ. Наибольшее развитие данная технология получила в форме разработки динамически загружаемых DLL-библиотек. DLL-библиотека представляет собой отдельный файл с расширением «DLL», в котором находятся какие-либо процедуры и функции. Они становятся доступными при подключении к программе DLL-файла.
2. При подключении DLL-библиотеки к программе она загружается в оперативную память компьютера и находится там до завершения программы. Если запускается несколько программ, и всем им нужна одна и таже DLL-библиотека, то она загрузится в память в одном экземпляре, а выгрузится из памяти при закрытии последней (использующей библиотеку) программы.
3. При одновременной работе нескольких программ с одной и той же DLL-библиотекой обеспечивается независимость переменных DLL-модуля для каждой прикладной программы. Таким образом получается, что процедуры и функции загружаются в одном экземпляре, а переменные DLL копируются для каждой прикладной программы отдельно.
4. Прикладная программа на Delphi может подключать DLL-библиотеку двумя способами: динамическим и статическим. При статическом способе DLL-библиотека автоматически подключается при запуске программы и автоматически деактивируется при ее завершении. Если указанный DLL-файл будет не найден, то возникнет ошибка выполнения программы. При динамическом способе DLL-библиотека подключается и отключается в нужное программисту время, при этом обеспечивается контроль над ошибочными ситуациями (например, если DLL-файл не найден).
5. Для создания файла DLL в Delphi необходимо выбрать пункт меню: «File -> New Item -> DLL wizard». Исходный код DLL-библиотеки нужно набрать в открывшимся окне и сохранить как файл с расширением «DPR». Никакие формы и файлы с расширение «PAS» здесь не используются.
6. Для одновременной отладки и создания программы и DLL-библиотеки лучше всего запустить две копии «Delphi» и в них работать отдельно.

**1.1. Пример простейшей библиотеки DLL**

**library demo\_dll;**

function sc(x: real): real export; stdcall; { Функция расчета }

begin

xsin:=sin(x)+ cos(x);

end;

exports sc; { Экспорт процедур и функций для внешних программ }

begin { Секция инициализации библиотеки DLL }

end.

end.

**1.2. Пример использования библиотеки DLL**

unit Demo\_dll;

…

var Form1: TForm1;

function sc(x: real): real; stdcall; external 'demo\_dll.dll'; { Подключение функции из DLL-библиотеки }

implementation

procedure TForm1.Button1Click(Sender: TObject);

begin

showmessage(floattostr(sc(5)));

end;

**1.3. Пример расширенной библиотеки DLL**

library repa;

uses SysUtils, Classes, Windows, Registry, StdCtrls;

procedure Save1(edit1\_text: pchar; mm1: pchar; scrollbar1\_position: integer); export; stdcall;

// Сохранение данных в реестр

var nfile, s: string; t: tregistry; i,k: integer; m1: tstrings;

begin

m1:=TStringList.Create; m1.Text:=mm1;

t:=tRegistry.Create;

t.RootKey:=HKEY\_CURRENT\_USER;

t.OpenKey('\Software\is-302\tsn-12', true);

t.writestring('', 'repa.dll');

t.WriteString('Edit1\_text', edit1\_text);

k:=m1.count;

t.WriteInteger('memo1\_count', k);

for i:=0 to m1.Count-1 do begin

s:=m1[i]; t.Writestring('memo1\_L'+inttostr(i+1), s);

end;

t.WriteInteger('scrollbar1\_position', scrollbar1\_position);

t.closekey; m1.Free;

t.Free;

end;

procedure Load1(var edit1\_text: pchar; var mm1: pchar; var scrollbar1\_position: integer); export; stdcall;

// Считывание данных из реестра

var path\_pr, nfile, s,m: string; t: tregistry; i,k,l: integer; m1,edit: tstrings;

begin

m1:=TStringList.Create;

t:=tRegistry.Create;

t.RootKey:=HKEY\_CURRENT\_USER;

t.OpenKey('\Software\is-302\tsn-12', true);

m1.text:='';

try

edit1\_text:=pchar(t.Readstring('edit1\_text'));

except

edit1\_text:='????';

end;

try

k:=t.readInteger('memo1\_count');

for i:=0 to k-1 do begin

s:= t.readstring('memo1\_L'+inttostr(i+1));

m1.Add(s);

end;

except

end;

mm1:=pchar(m1.text);

try

l:= t.readInteger('scrollbar1\_position'); scrollbar1\_Position:=l;

except

end;

t.closekey; m1.Free; t.Free;

end;

procedure Clear1(var edit1\_text: pchar; var mm1: pchar; var scrollbar1\_position: integer); export; stdcall;

// Очистка данных

begin

edit1\_text:=''; mm1:=''; scrollbar1\_position:=0;

end;

exports Clear1, Save1, Load1;

begin

end.

**1.4. Пример использования расширенной библиотеки DLL**

unit Unit1;

…

var

Form1: TForm1;

procedure Save1(edit1\_text: pchar; mm1: pchar; scrollbar1\_position: integer); stdcall; external 'repa.dll';

procedure Load1(var edit1\_text: pchar; var mm1: pchar; var scrollbar1\_position: integer); stdcall; external 'repa.dll';

procedure Clear1(var edit1\_text: pchar; var mm1: pchar; var scrollbar1\_position: integer); stdcall; external 'repa.dll';

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

// Запись данных

begin

Save1(Pchar(edit1.text), Pchar(memo1.Lines.Text), scrollbar1.Position);

end;

procedure TForm1.Button2Click(Sender: TObject);

// Очистка данных

var e,m: pchar; k: integer;

begin

clear1(e, m, k);

edit1.text:=e; memo1.Lines.Text:=m; scrollbar1.Position:=k;

end;

procedure TForm1.Button3Click(Sender: TObject);

// Загрузка данных

var e,m: pchar; k: integer;

begin

Load1(e, m, k);

edit1.text:=e; memo1.Lines.Text:=m; scrollbar1.Position:=k;

end;

end.