ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМ. Н.П. ОГАРЁВА»

Направление: Информатика и вычислительная техника

**ЛАБОРАТОРНАЯ РАБОТА**

по дисциплине

«Программная Инженерия»

ЛР – 02069964 – Программная инженерия – 08 – 19

Лабораторная работа № 8

Выполнили: Проверил:

студенты 441 группы

21.11.2019 \_\_\_\_\_\_\_\_\_\_\_

С. О. Леушкин С.А. Федосин

Д.Д. Абдуллаев

Саранск 2019

# **Лабораторная работа №8.**

# Разработка простейшей динамически присоединяемой библиотеки

### Цель работы.

* научиться самостоятельно создавать динамически присоединяемые библиотеки (DLL).

Lab8.dll

library LAB8;

{ Important note about DLL memory management: ShareMem must be the

first unit in your library's USES clause AND your project's (select

Project-View Source) USES clause if your DLL exports any procedures or

functions that pass strings as parameters or function results. This

applies to all strings passed to and from your DLL--even those that

are nested in records and classes. ShareMem is the interface unit to

the BORLNDMM.DLL shared memory manager, which must be deployed along

with your DLL. To avoid using BORLNDMM.DLL, pass string information

using PChar or ShortString parameters. }

uses

System.SysUtils,

System.Classes,

XString in 'XString.pas';

exports

UpCaseFirstStr index 1,

LTrimStr index 2,

RTrimStr index 3,

StripStr index 4;

{$R \*.res}

begin

end.

XString.pas

unit XString;

interface

Function UpCaseFirstStr(Const S: String): ShortString; register;

Function LTrimStr(Const S: String): ShortString; register;

Function RTrimStr(Const S: String): ShortString; register;

Function StripStr(Const S: String): ShortString; register;

implementation

Function UpCaseFirstStr(Const S: String): ShortString; register;

Var

Index: Byte;

First: Boolean;

begin

Result:= S;

First:= True;

for Index := 1 to Length(S) do

begin

if First then

Result[Index]:= UpCase(Result[Index]);

if Result[Index] = ' ' then

First:= True

else

First:= False;

end;

end;

Function LTrimStr(Const S: String): ShortString; register;

Var

Index,

MaxIndex: Integer;

begin

Index:= 1;

MaxIndex:= Length(S);

while (Index<=MaxIndex) And (S[Index]=' ') do

Inc(Index);

Result:= Copy(S,Index,MaxIndex-Index+1);

end;

Function RTrimStr(Const S: String): ShortString; register;

Var

Index: Integer;

begin

Index:= 1;

while (Index > 0) And (S[Index]=' ') do

Dec(Index);

Result:= Copy(S,1,Index);

end;

Function StripStr(Const S: String): ShortString; register;

Var

Index: Integer;

begin

Result:= '';

for Index := 1 to Length(S) do

if S[Index] <> ' ' then

Result:= Result+S[Index];

end;

end.

Main.pas

unit main;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, libunit, Vcl.StdCtrls;

type

TForm1 = class(TForm)

Edit1: TEdit;

Label1: TLabel;

Edit2: TEdit;

UpCaseFirst: TButton;

LTrim: TButton;

RTrim: TButton;

Strip: TButton;

procedure UpCaseFirstClick(Sender: TObject);

procedure LTrimClick(Sender: TObject);

procedure RTrimClick(Sender: TObject);

procedure StripClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.LTrimClick(Sender: TObject);

begin

Edit2.Text:=LTrimStr(Edit1.Text);

Label1.Caption:=(Sender as TButton).Caption;

end;

procedure TForm1.RTrimClick(Sender: TObject);

begin

Edit2.Text:=RTrimStr(Edit1.Text);

Label1.Caption:=(Sender as TButton).Caption;

end;

procedure TForm1.StripClick(Sender: TObject);

begin

Edit2.Text:=StripStr(Edit1.Text);

Label1.Caption:=(Sender as TButton).Caption;

end;

procedure TForm1.UpCaseFirstClick(Sender: TObject);

begin

Edit2.Text:=UpCaseFirstStr(Edit1.Text);

Label1.Caption:=(Sender as TButton).Caption;

end;

end.

Libunit.pas

unit libunit;

interface

Function UpCaseFirstStr(Const S: String): ShortString; register;

Function LTrimStr(Const S: String): ShortString; register;

Function RTrimStr(Const S: String): ShortString; register;

Function StripStr(Const S: String): ShortString; register;

implementation

Const LibName='lab8.dll';

Function UpCaseFirstStr; external LibName index 1;

Function LTrimStr; external LibName index 2;

Function RTrimStr; external LibName index 3;

Function StripStr; external LibName index 4;

end.

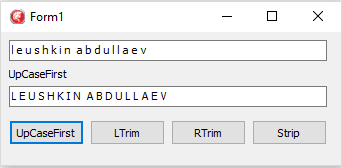


Рисунок 1

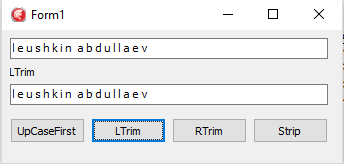


Рисунок 2

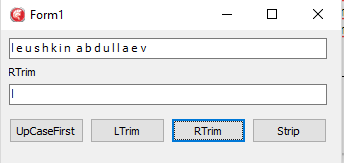


Рисунок 3

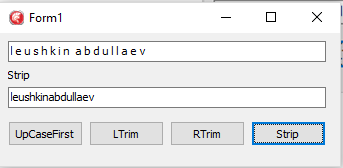


Рисунок 4

### Контрольные вопросы.

1. **Объясните смысл слов index, register, export и external, используемых для создания DLL и применения хранящихся в ней функций.**

Раздел Exports помогает компилятору и компоновщику создать специальный заголовок DLL-модуля, в котором перечисляются имена подпрограмм и адреса их точек входа. Можно изменить умалчиваемую индексацию и явно указать индекс подпрограммы, добавив за ее именем в списке Exports слово index и целое число без знака в диапазоне от 0 до 32767.

Для использования подпрограмм из DLL необходимо описать их как внешние, добавив за словом External имя библиотеки в апострофах.

Директива register используется по умолчанию. Поэтому нет необходимости добавлять ключевое слов register после объявления функции. Вызов такого типа называется быстрым (fast call). В нем используются три расширенных регистра процессора, в которые помещаются переменные длиной не более 32-х разрядов и указатели. Остальные параметры помещаются в стек слева направо. После использования стек очищается вызываемой процедурой.

1. **Почему в программе используется тип ShortString, а не String.**

Для того чтобы была верная кодировка, так как String по умолчанию принимает тип AnsiString. Поэтому нам приходится его переопределять на ShortString.