Министерство образования и науки РФ
ФГБПОУ ВПО Тульский государственный университитет
КАФЕДРА АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

МОДУЛИ И ДИНАМИЧЕСКИ ПОДКЛЮЧАЕМЫЕ БИБЛИОТЕКИ

Лабораторная работа № 6
по курсу «Программирование на ЯВУ»

Вариант № 4

Выполнил: студент группы 220601 _____ Белым А.А.
(подпись)

Проверил: к. ф.-м. н., доцент _____ Сулимова В.В.
(подпись)

Тула 2011

Цель работы

Цель работы заключается в том, чтобы научиться создавать и использовать модули и библиотеки статической и динамической компоновки. Также требуется написать программу с их использованием.

Задание на работу

Составить программу вычисления для заданных значений x, y, z арифметического выражения:

$$w = |\cos x - \cos y|^{(1+2\sin^2 y)} \left[1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}\right]$$

С помощью инспектора объектов изменить цвет формы, шрифт выводимых символов.

Теоретическая справка

Схема алгоритма

Ниже на рисунке 1 представлена схема алгоритма вычисления значения функции.

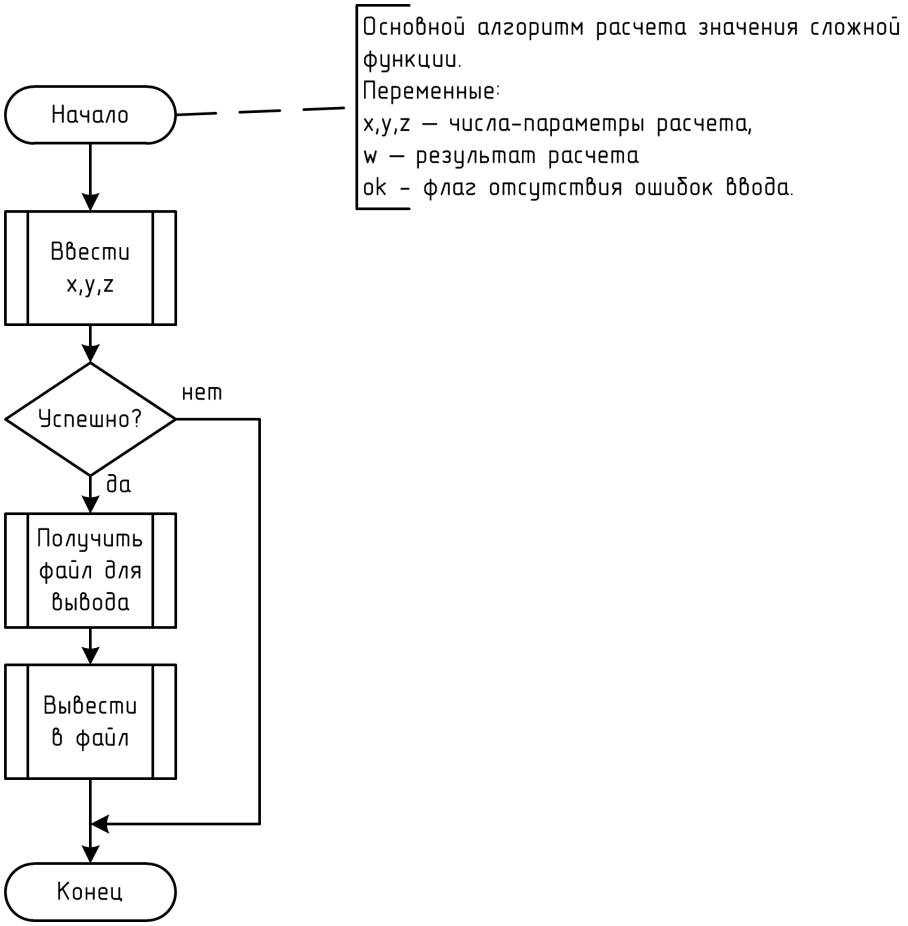

Рисунок 1 - Схема алгоритма вычисления значения функции

На рисунке 2 представлена схема алгоритма ввода параметров расчета значения функции.

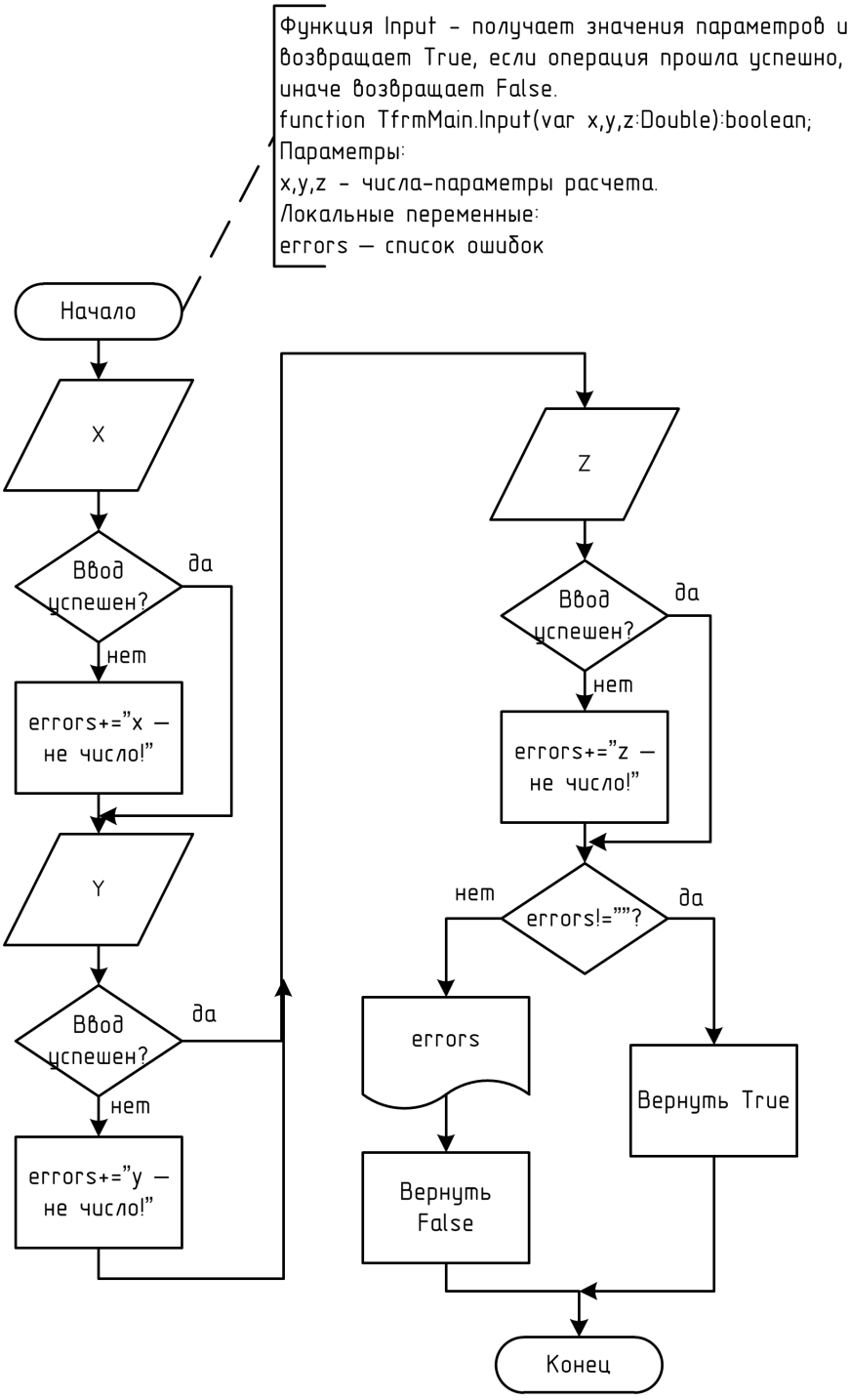

Рисунок 2 - Схема алгоритма ввода параметров функции

На рисунке 3 представлена схема алгоритма ввода расчета значения функции от заданных параметров.

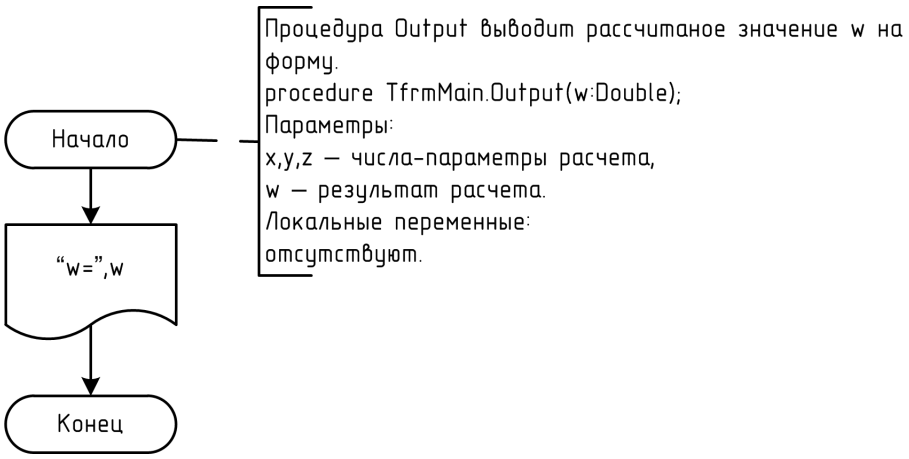

Рисунок 3 - Схема алгоритма расчета значения функции

На рисунке 4 представлена схема алгоритма вывода расчитанного значения функции.

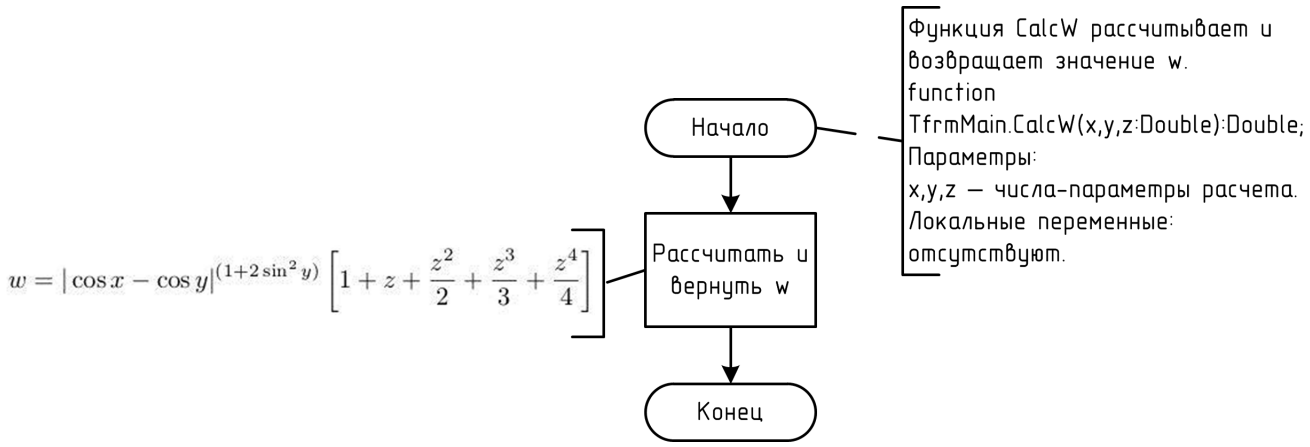

Рисунок 4 - Схема алгоритма вывода значения функции

Инструкция пользователю

Программа позволяет вычислить значение сложной функции трёх вещественных аргументов.

Для работы программе необходимо передать 3 аргумента, от которых будет вычисляться значение функции. Введите их в указанные поля в окне программы, отделяя дробную часть от целой запятой. Возможна запись в экспоненциальной форме - показатель экспоненты указывается после буквы e. После ввода данных нажмите кнопку "Посчитать".

Программа выведет результат в большое текстовое поле внизу окна. Для завершения работы программы нажмите кнопку "Выход".

Инструкция программисту

При разработке программы вычисления значения функции в модуле UnitMain, содержащем основную форму, были написаны следующие процедуры и функции:

1. Процедура btnRunClick - обработчик события шелчка мышки на кнопке btnRun - основная процедура программы.

procedure btnRunClick(Sender: TObject);

Параметры процедуры представлены в таблице 1, локальные переменные - в таблице 2.

Таблица 1 - Параметры процедуры запуска вычислений значения функции

имя	**тип**	**предназначение**
Sender	TObject	объект-возбудитель события

Таблица 2 - Локальные переменные процедуры запуска вычислений значения функции

имя	**тип**	**предназначение**
x,y,z	Double	параметры расчёта значения функции
w	Double	значение функции
ok	boolean	флаг состояния программы

2. Процедура-обработчик события btnExitClick при шелчке мышке по кнопке btnExit завершает приложение.

procedure btnExitClick(Sender: TObject);

Параметры процедуры представлены в таблице 3.

Таблица 3 - Параметры процедуры-обработчика событий шелчка мышки кнопки btnExit

имя	**тип**	**предназначение**
Sender	TObject	объект-возбудитель события

3. Процедура FormCreate добавляет к информационному полю memResult строку "w=".

procedure FormCreate(Sender: TObject);

Параметры процедуры представлены в таблице 4.

Таблица 4 - Параметры процедуры-обработчика событий создания формы frmMain

имя	**тип**	**предназначение**
Sender	TObject	объект-возбудитель события

В библиотеке LabDLL были объявлены следующие функции:

1. Функция ShowForm показывает форму frmMain из модуля UnitMain в модальном режиме.

Возвращает результат модального запуска.

function ShowForm: Integer;

2. Функция Input - получает значения параметров и возвращает True, если операция прошла успешно, иначе возвращает False.

function Input(var x,y,z:Double):boolean;

Параметры-переменные функции представлены в таблице 5, локальные переменные - в таблице 6.

3. Процедура Output выводит рассчитаное значение w на форму.

procedure Output(w:Double);

Таблица 5 - Параметры функции ввода параметров рассчета

имя	**тип**	**предназначение**
x,y,z	Double	параметры расчёта значения функции

Таблица 6 - Локальные переменные функции ввода параметров рассчета

имя	**тип**	**предназначение**
errors	String	список ошибок ввода вывода

Параметры процедуры представлены в таблице 7.

Таблица 7 - Параметры процедуры вывода рассчитанного значения

имя	**тип**	**предназначение**
w	Double	значение функции

4. Функция CalcW рассчитывает и возвращает значение w.

function CalcW(x,y,z:Double):Double;

Параметры функции представлены в таблице 8.

Таблица 8 - Параметры функции получения значения w

имя	**тип**	**предназначение**
x,y,z	Double	параметры расчета w

Кроме того, написаны программы Lab6Static и Lab6Dynamic, подключающие библиотеку LabDLL статически и динамически соответственно, однако в них не содержатся определения функций и типов данных.

Текст программы

Ниже представлен текст модуля на языке Delphi 7, в котором содержится форма программы, реализующей расчёт значений функции и имеющей графический интерфейс.

```
unit UnitMain;
interface

uses

  Classes, Controls, Forms,
  StdCtrls, jpeg, ExtCtrls;

type
  TfrmMain = class(TForm)
    lblInfo: TLabel;//Информация о программе
    grpParams: TGroupBox;//Группа параметров
    btnRun: TButton;//Кнопка запуска
    btnExit: TButton;//Кнопка выхода
    edtX: TLabeledEdit;//Ввод параметра X
    edtY: TLabeledEdit;//Ввод параметра Y
    edtZ: TLabeledEdit;//Ввод параметра Z
    imgFormula: TImage;//Формула
    memResult: TMemo;//Результат расчета
    //Обработчик нажатия кнопки запуска
    procedure btnRunClick(Sender: TObject);
    //Обработчик нажатия кнопки выхода
    procedure btnExitClick(Sender: TObject);
    //Обработчик события создания формы
    procedure FormCreate(Sender: TObject);

  private
    { Private declarations }

  public
    { Public declarations }


  end;
    //Ввод данных
    function Input(var x,y,z:Double):boolean;  external 'LabDLL';
    //Вывод данных
    procedure Output(w:Double);   external 'LabDLL';
    //Расчет функции
    function CalcW(x,y,z:Double):Double; external 'LabDLL';
var
  frmMain: TfrmMain;

implementation

{$R *.dfm}
(*
Процедура btnRunClick – обработчик события шелчка мышки на кнопке
btnRun – основная процедура программы.
```

```
Параметры:
Sender:TObject — объект—возбудитель события
Локальные переменные:
x,y,z:Double—параметры расчёта значения функции
w   : Double —значение функции
*)
procedure TfrmMain.btnRunClick(Sender: TObject);
var x,y,z,w:Double;ok:boolean;
begin
  ok:=Input(x,y,z);
  if ok then begin
    w:=CalcW(x,y,z);
    Output(w);
  end;
end;
(*
Процедура—обработчик события btnExitClick при шелчке мышке по кнопке
btnExit завершает приложение.
Параметры:
Sender:TObject — объект—возбудитель события
*)
procedure TfrmMain.btnExitClick(Sender: TObject);
begin
Application.Terminate;
end;
(*
Процедура FormCreate добавляет к информационному полю memResult
строку "w=".
Параметры:
Sender:TObject—объект—возбудитель события
*)
procedure TfrmMain.FormCreate(Sender: TObject);
begin
  memResult.Text:=memResult.Text+'w=';
end;
end.
```

Далее представлен текст динамической библиотеки, которая использует вышеприведенную форму.

```
Library LabDLL;
Uses Forms,sysutils,classes,math,Dialogs,UnitMain in 'UnitMain.pas';

(*
Функция ShowForm показывает форму frmMain
из модуля UnitMain  в модальном режиме.
Возвращает результат модального запуска.
*)
function ShowForm: Integer;
begin
frmMain := TfrmMain.Create(Application);
Result := frmMain.ShowModal;
frmMain.Free;
end;

(*
```

```
 Функция CalcW рассчитывает и возвращает значение $w$.
Параметры:
x,y,z:Double—параметры расчета w
*)
function CalcW(x,y,z:Double):Double;
begin
  CalcW:=Power(Abs(cos(x)—cos(y)),1+2*sqr(sin(y)))*
    (1+z+sqr(z)/2+z*sqr(z)/3+sqr(z)*sqr(z)/4);
end;

(*
Процедура Output выводит рассчитаное значение $w$ на форму.
Параметры:
w   : Double — значение функции
*)
procedure Output(w:Double);
begin
  with frmMain do begin
    memResult.Text:=memResult.Text+FloatToStr(w)+#13#10+'w=';
    memResult.SelLength:=length(memResult.Text);
  end;
end;

(*
Функция Input — получает значения параметров и возвращает True,
если операция прошла успешно, иначе возвращает False.
Параметры:
x,y,z:Double—параметры расчёта значения функции
Локальные переменные функции ввода параметров рассчета
errors:String—список ошибок ввода вывода
*)
function Input(var x,y,z:Double):boolean;
var errors:string;
begin
  with frmMain do begin
  errors:='';
  if not TryStrToFloat(edtX.Text,x) then
    errors:=#10#13+'x — не вещественное число!';
  if not TryStrToFloat(edtY.Text,y) then
    errors:=errors+#10#13+'y — не вещественное число!';
  if not TryStrToFloat(edtZ.Text,z) then
    errors:=errors+#10#13+'z — не вещественное число!';
  if (errors<>'') then begin
          Input:=False;MessageDlg('Ошибки: '+errors,mtError,[mbOK],0)
  end else Input:=True;
  end;
end;

exports  CalcW,Input,Output,ShowForm;
begin

end.
```

Ниже приведен текст приложения, использующего динамическую библиотеку при статическом связывании.

```
program Lab6Static;

uses
  Forms;

function ShowForm: Integer; external 'LabDLL';
{$R *.res}

begin
  Application.Initialize;
  ShowForm;
end.
```

Далее приведен текст приложения, использующего динамическую библиотеку при динамическом связывании.

```
program Lab6Dynamic;

uses
  Windows,Forms,Dialogs;

{$R *.res}
var Handle:LongWord;
    ShowForm:function:Integer;
begin
  Application.Initialize;
  Handle := LoadLibrary('LabDLL.dll');
  if Handle = 0 then
  begin
    MessageDlg('Не найдена библиотека LabDLL.DLL',mtError,[mbOk],0) ;
    Halt
  end;
  @ShowForm:=GetProcAddress(Handle,'ShowForm');
  if @ShowForm=NIL then
    MessageDlg('Не найдена библиотека LabDLL.DLL',mtError,[mbOk],0)
  else ShowForm;
  FreeLibrary(Handle);
end.
```

Тестовый пример

Ниже на рисунке 5 представлен общий вид окна программы.

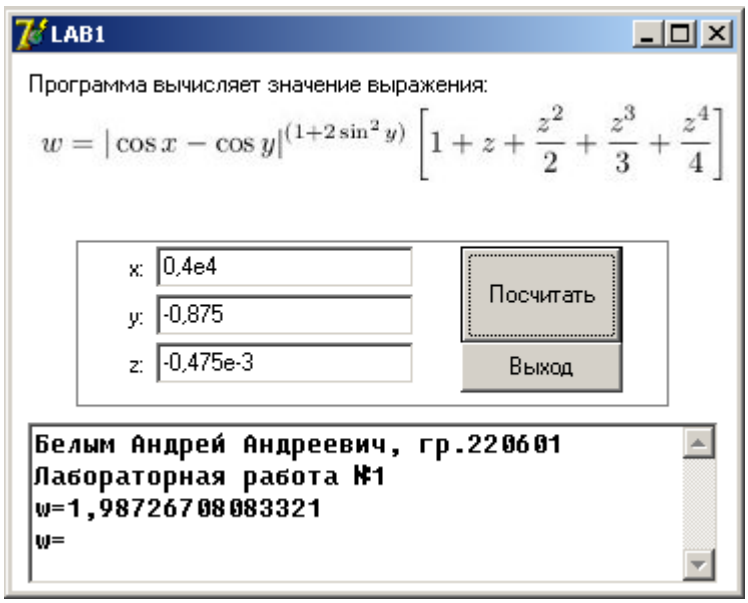

Рисунок 5 - Внешний вид программы

Вывод

В этой лабораторной работе я научился использовать библиотеки DLL при написании программ на Delphi. Динамические библиотеки позволяют использовать в собственных скомпилированный машинный код(без доступа к исходным текстам), причем не имеет значения исходный яхык написания библиотеки, а также позволяет подключать код не только во время компиляции, но и во время выполнения программы. Эти свойства делают динамические библиотеки чрезвычайно мощным инструментам, который позволяет и снизить трудозатраты, и сделать приложение более функциональным и гибким.