**Лабораторная работа №\_\_\_**

**Цель работы** - Приобретение практических навыков, применения методов модульного проектирования, средств свободной кроссплатформенной среды разработки программ Lazarus, методики системного тестирования при реализации модульного программного обеспечения в рамках международного стандарта ISO / IEC 12207

Вариант №11

**Задание**:

1. Реализовать программное средство (графическое приложение) решения задачи 1.1, используя стандартные модули обрабо

тки символьных и строчных данных.

2. реализовать программное средство (графическое приложение) решения задачи 1.2, используя стандартные модули математических вычислений

Задание 1.1

Пользователем вводится украинское слово или знак пунктуации, в единственный компонент вывода, вывести следующую информацию:

* При нажатии кнопки "Количество символов" - количество введенных пользователем букв, если пользователем введено слово и сообщение «Введен знак пунктуации», если введено знак пунктуации;
* По нажатию кнопки "Наличие слова" - текущую дату, если пользователем введен знак пунктуации; иначе - сообщение, есть ли в введённое слово в следующей части стихотворения Анатолия Тарана

Хай щастить у Вашій хаті, мамо,

Всім, хто переступить Ваш поріг:

Добрим людям, і птахам так само,

І котові, що в теплі приліг;

Хай щастить кожненькій деревині,

Що до хати віти притуля,

Хай щастить малесенькій травині

Й вітерцю, що приліта здаля.

Анатолій Таран

Задача 1.2 -

Пользователь вводит числовые значения х, у, z. Вычислить и вывести Q ".

1. Порядок выполнения задачи №1.1
2. Получив у преподавателя условия задачи 1.1
3. Проанализировал требования к программному средству
4. Спроектировал архитектуру модульного программного средства
5. Детально спроектировал отдельные элементы
6. Разработал тестовый набор из пяти негативных случаев, сохранил его под названием **Artifact\_TEST\_SUITE\_lab\_poisk\_t1**, заполнил необходимые поля,
7. Описал тестовые случаи.
8. В Lazarus IDE создал проект визуального приложения, присвоил имя **poisk** и сохранил его с расширением **lpi**
9. На основе результатов спроектированного программного средства решения задачи
10. Получил следующие результаты, графическое приложение имеет вид «Windows» окна с расположенными на нем компонентами, кнопок управления, поля ввода, поля вывода, и для наглядности поле с сохраненным в нем стиха.
11. При помощи стандартных функций и процедур получил конечный исходный код.
12. Скомпилировал исходный код получил графическое приложение с расширением «.exe» и сохранил приложение под названием **poisk.exe**
13. Соответственно стандарту ISO/IEC 12207:2008 было мною же осуществлено системное тестирование программного средства при помощи ранее созданного тестового набора.

Результаты тестирования будут приложены к данному отчету.

Листинг задачи 1.1

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, StdCtrls,DateUtils,Crt;

type

upper= set of 'А'..'Я';

low = set of 'а'..'я';

{ TForm1 }

TForm1 = class(TForm)

Button1: TButton;

Button2: TButton;

Edit1: TEdit;

Memo1: TMemo;

Memo2: TMemo;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ private declarations }

public

{ public declarations }

end;

const

big:upper=['А'..'Я'];

sm:low =['а'..'я'];

var

Form1: TForm1;

k,i,h: integer;

g:string;

a,b,c,d,x:real;

implementation

{$R \*.lfm}

{ TForm1 }

procedure TForm1.Button1Click(Sender: TObject);

begin

if Edit1.Text<> ' ' then

begin

Memo1.Clear;

g:=UTF8ToAnsi(Edit1.Text);

k:=length(g);

for i:=1 to k do

if ('A'<Edit1.Text[i])or(Edit1.Text[i]<'Z') or ('a'<Edit1.Text[i])or(Edit1.Text[i]<'z') then

if ('А'<Edit1.Text[i])or(Edit1.Text[i]<'Я') or ('а'<Edit1.Text[i])or(Edit1.Text[i]<'я') then

begin

if Pos(Edit1.Text,Memo2.Text)<>0 then

begin

Memo1.Lines[0]:= 'Yes';

Memo1.Lines[1]:= 'Entered word is in verse';

end

else

begin

Memo1.Lines.Add(DateTimeToStr(date));

Memo1.Lines[1]:= 'Entered Incorrect value';

break;

end;

end;

end

else

ShowMessage('Please enter your request ');

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

if Edit1.Text<> ' ' then

begin

Memo1.Clear;

a:=0;

b:=0;

c:=0;

d:=0;

x:=0;

h:=2;

for i:=1 to length(Edit1.Text) do

if ('A'<Edit1.Text[i] ) and ( Edit1.Text[i]<'Z') then

a:=a+1

else

if ('a'<Edit1.Text[i]) and ( Edit1.Text[i]<'z') then

b:=b+1

else

if Edit1.Text[i] in big then

c:=c+1

else

if Edit1.Text[i] in sm then

d:=d+1

else

begin

ShowMessage('Error! Incorrect value');

break;

h:=0;

end ;

if h>0 then

begin

x:=a+b+c/2+d/2;

Memo1.Lines[0]:=FloatToStr(x);

end

end

else

ShowMessage('Please enter your request ');

end;

end.

**Порядок выполнения задачи №1.2**

1. Получив у преподавателя условия задачи 1.2
2. Проанализировал требования к программному средству
3. Спроектировал архитектуру модульного программного средства
4. Детально спроектировал отдельные элементы
5. Разработал тестовый набор из пяти позитивных случаев, сохранил его под названием **Artifact\_TEST\_SUITE\_lab\_\_calc\_t1.doc**, заполнил необходимые поля,
6. Описал тестовые случаи
7. В Lazarus IDE создал проект визуального приложения присвоил имя **calc** и сохранил его с расширением **.lpi**
8. На основе результатов спроектированного программного средства решения задачи
9. Получил следующие результаты, графическое приложение имеет вид «Windows» окна с расположенными на нем компонентами, кнопок управления, полей ввода, поля вывода.
10. При помощи стандартных функций и процедур получил конечный исходный код.
11. Скомпилировал исходный код получил графическое приложение с расширением «.exe» и сохранил под названием **calc.exe**
12. Соответственно стандарту ISO/IEC 12207:2008 было мною же осуществлено системное тестирование программного средства при помощи ранее созданного тестового набора.
13. Результаты тестирования будут приложены к данному отчету.

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, StdCtrls,Math,Crt;

type

{ TForm1 }

TForm1 = class(TForm)

Button1: TButton;

Button2: TButton;

Button3: TButton;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ private declarations }

public

{ public declarations }

end;

const z=['0','1','2','3','4','5','6','7','8','9',',','-'] ;

var

Form1: TForm1;

i,d,re: integer;

a:string;

r,x,y,w,h,sum:real;

implementation

{$R \*.lfm}

{ TForm1 }

procedure filtr (per:string; var r: real );

begin

for i:=1 to length(per) do

if per[i]in z then

begin

a:=concat(a,per[i]);

d:=d+1;

end

else

begin

d:=0;

break;

end;

if d>0 then

begin

r:=StrToFloat(a);

a:=' ';

end

else

begin

ShowMessage('Введено не корректное значение');

re:=0;

exit;

end;

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

close;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

re:=1;

filtr(Edit1.Text, x);

filtr (Edit2.Text, y);

filtr (Edit3.Text, w);

if re=0 then

exit else

begin

if (length(Edit1.Text)=0) and (length(Edit2.Text)=0)and (length(Edit3.Text)=0) then

begin

ShowMessage('Не введено значение');

exit;

end else

begin

if (x = 0 ) and (y = 0) and (w = 0) then

begin

ShowMessage('Деление на ноль');

exit;

end else

if w<=0 then

begin

ShowMessage('Z должно принимать значения больше 0');

exit ;

end else

h:=(12\*x+(sqr(y)-Pi\*sqrt(w)));

if h<=0 then

begin

ShowMessage('Подкоренное выражение отрицательное') ;

exit;

end

else

begin

sum:=Power(x,w)-Power(y,3)+sqrt(abs((sqr(w))\*(exp(x))))/(12\*x+(sqr(y)-Pi\*sqrt(w)));

Edit4.Text:=FloatToStr(sum);

x:=0; y:=0; w:=0;

end;

end;

end;

end;

end.

Выводы

При выполнении лабораторной работы я приобрел практические навыки, анализа требования к программному продукту, применения модульного программирования, порядок выполнения, правильной постановки задачи, применения методов модульного проектирования, открыл для себя большой диапазон возможностей разработки программных средств на свободной кроссплатформенной среде разработки «Lazarus» при помощи языка FPC.

В кроссплатформенной среде разработки «Lazarus» применяя компоненты, дополнительные элементы программы можно по сравнению с консольным приложением получить более достойную программу, она предоставляет возможность создавать более приятный интерфейс для пользователя, более визуально легче для разработчика.

При решении поставленных задач, сталкивался с многими трудностями и нехваткой знания, но в процессе реализации получил необходимую информацию не только по поставленной задачи, но также получил дополнительно знания, которые я приобрел по мере решения задачи. Ознакомился с методикой системного тестирования, правильного оформления артифактов и тонкостями тестирования которые можно приобрести только практически.

После выполнения лабораторной работы получив ожидаемый результат, появилась уверенность в том, что уже возможно решение более сложных задач.

©