Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 1

з навчальної дисципліни “Модульне програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

НА ОСНОВІ МОДУЛІВ FPC

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

[https://github.com/odorenskyi/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи КІ-19

Ковальов В.О.

ПЕРЕВІРИВ

старший викладач кафедри кібербезпеки та програмного забезпечення

Поліщук Л. І.

Кропивницький – 2020

Тема: реалізація програмного забезпечення на основі модулів FPC.

Мета: набуття ґрунтовних вмінь і практичних навичок застосування методології модульного проектування програмного забезпечення, міжнародного стандарту ISO/IEC 12207, підходів до тестування програмних засобів на системному рівні, реалізації графічних додатків з застосуванням модулів FPC для оброблення текстових даних і математичних обчислень в кросплатформовому Lazarus IDE.

Завдання:

1. Реалізувати програмне забезпечення (графічний додаток) розв’язування задачі 1.1 на основі модулів FPC для оброблення символьних і рядкових даних.

2. Реалізувати програмне забезпечення (графічний додаток) розв’язування задачі 1.2 на основі модулів FPC для математичних обчислень.

**Варіант 12:**

*Задача 1.1:*

Користувач вводить слово. Вивести в єдиний компонент виведення:

* За натисненням кнопки «Випадкова літера» - випадкову літеру української абетки від Д до Ц включно;
* За натисненням кнопки «Дата» - поточну дату у форматі «ДД.ММ», якщо за локальним годинником ОС наразі вечір, інакше – у форматі «місяць\_прописом РРРР року»;
* За натисненням кнопки «Перевірити» - повідомлення, чи є введене користувачем слово у наступному вірше Ліни Костенко:

Мені відкрилась істина печальна:

Життя зникає, як ріка Почайна.

Через віки, а то й через роки,

Ріка вже стане спогадом ріки.

І тільки верби знатимуть старі:

Киян хрестили в ній, а не в Дніпрі.

*Задача 1.2:*

За введеним користувачем значеннями x,y,z обчислити й вивести Q:

Код lab1\_task1\_Kovalov:

unit Task1\_Unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls, Menus, Info\_task1, Valid\_task1;

type

{ TForm1 }

TForm1 = class(TForm)

Label2: TLabel;

MainMenu1: TMainMenu;

MenuItem1: TMenuItem;

RandomS: TButton;

Data: TButton;

Check: TButton;

Edit1: TEdit;

Label1: TLabel;

procedure CheckClick(Sender: TObject);

procedure DataClick(Sender: TObject);

procedure MenuItem1Click(Sender: TObject);

procedure RandomSClick(Sender: TObject);

private

public

end;

var

Form1: TForm1;

implementation

{$R \*.lfm}

{ TForm1 }

procedure TForm1.RandomSClick(Sender: TObject);

begin

randomize;

case random(22)+1 of

1 : Label1.Caption := 'Випадкова літера: Д';

2 : Label1.Caption := 'Випадкова літера: Е';

3 : Label1.Caption := 'Випадкова літера: Є';

4 : Label1.Caption := 'Випадкова літера: Ж';

5 : Label1.Caption := 'Випадкова літера: З';

6 : Label1.Caption := 'Випадкова літера: И';

7 : Label1.Caption := 'Випадкова літера: І';

8 : Label1.Caption := 'Випадкова літера: Ї';

9 : Label1.Caption := 'Випадкова літера: Й';

10: Label1.Caption := 'Випадкова літера: К';

11: Label1.Caption := 'Випадкова літера: Л';

12: Label1.Caption := 'Випадкова літера: М';

13: Label1.Caption := 'Випадкова літера: Н';

14: Label1.Caption := 'Випадкова літера: О';

15: Label1.Caption := 'Випадкова літера: П';

16: Label1.Caption := 'Випадкова літера: Р';

17: Label1.Caption := 'Випадкова літера: С';

18: Label1.Caption := 'Випадкова літера: Т';

19: Label1.Caption := 'Випадкова літера: У';

20: Label1.Caption := 'Випадкова літера: Ф';

21: Label1.Caption := 'Випадкова літера: Х';

22: Label1.Caption := 'Випадкова літера: Ц';

end;

end;

procedure TForm1.CheckClick(Sender: TObject);

var

input : string;

begin

if edit1.text = '' then

begin

label1.caption := 'Рядок пустий.';

exit;

end;

input := edit1.text;

if Slovo(input) = false then begin label1.caption := 'Введено некоректне слово.'; exit; end;

if input = 'Мені' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'відкрилась' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'істина' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'печальна' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'життя' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'зникає' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'як' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'ріка' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'Почайна' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'Через' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'віки' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'а' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'то' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'й' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'через' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'роки' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'вже' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'стане' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'спогадом' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'ріки' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'І' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'тільки' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'верби' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'знатимуть' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'старі' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'киян' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'хрестили' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'в' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'ній' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'не' then begin label1.caption := 'Слово присутнє'; exit; end;

if input = 'Дніпрі' then begin label1.caption := 'Слово присутнє'; exit; end;

label1.caption := 'Слово відсутнє';

end;

procedure TForm1.DataClick(Sender: TObject);

var

CurT : TDateTime;

begin

CurT := Now;

if (StrToInt(FormatDateTime('hh', CurT)) > 17) and (StrToInt(FormatDateTime('hh', CurT)) < 23) then

Label1.Caption := 'Поточна дата - ' + FormatDateTime('dd.mm', CurT)

else

Label1.Caption := FormatDateTime('mmmm, yyyy', CurT) + ' року';

end;

procedure TForm1.MenuItem1Click(Sender: TObject);

begin

Form2.Show;

end;

end.

Валідатор до завдання 1:

unit Valid\_task1;

{$mode objfpc}{$H+}

interface

function Slovo (input:string) : boolean;

implementation

function Slovo (input:string) : boolean;

var

Letters : string;

i : byte;

begin

Letters := 'AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZzАаБбВвГгДдЕеЁёЖжЗзИиЙйКкЛлМмНнОоПпРрСсТтУуФфХхЦцЧчШшЩщЪъЫыЬьЭэЮюЯяЄєЇїІіҐґ';

for i := 1 to length(input) do

begin

if pos(input[i],Letters) = 0 then

begin Slovo := false; exit; end;

Slovo := true;

end;

end;

end.

Код lab1\_task2\_Kovalov:

unit task2\_unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls, Menus, info\_task2, Valid\_task2;

type

{ TForm1 }

TForm1 = class(TForm)

Button1: TButton;

MainMenu1: TMainMenu;

MenuItem1: TMenuItem;

x\_input: TEdit;

y\_input: TEdit;

z\_input: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

procedure Button1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure MenuItem1Click(Sender: TObject);

private

public

end;

const p = 3.14159;

var

Form1: TForm1;

implementation

{$R \*.lfm}

{ TForm1 }

procedure TForm1.Button1Click(Sender: TObject);

var

str : string[255];

a, b, c, q : extended;

x,y,z : extended;

begin

// Proverka

if (x\_input.text = '') or (y\_input.text = '') or (z\_input.text = '') then

begin

label1.caption := 'Одне з полів є порожнім.';

exit;

end;

str := x\_input.text;

str := Znaks(str);

if lettersin(str) = false then begin Label1.Caption := 'Помилка. Введене некоректне число'; exit; end;

x := StrToFloat(str);

str := y\_input.text;

str := Znaks(str);

if lettersin(str) = false then begin Label1.Caption := 'Помилка. Введене некоректне число.'; exit; end;

y := StrToFloat(str);

str := z\_input.text;

str := Znaks(str);

if lettersin(str) = false then begin Label1.Caption := 'Помилка. Введене некоректне число.'; exit; end;

z := StrToFloat(str);

if interval(x,y,z) = false then begin showmessage('Діапазон допустимих чисел обмежений. Кожне з введених чисел має бути від 1 до 707 включно, а також число Х не може бути рівним 5.'); exit; end;

a := abs(5-x);

b := pi\*(z+y);

c := b/a;

a := exp(abs(z))\*ln(12\*x);

b := a-c;

c := sqrt(b);

q := 3\*sin(c);

label1.caption := 'Результат: ' + FloatToStr(q);

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

end;

procedure TForm1.MenuItem1Click(Sender: TObject);

begin

form2.show;

end;

end.

Валідатор до завдання 2:

unit Valid\_task2;

interface

function LettersIn (s : string) : boolean;

function Znaks (s : string) : string;

function Interval (x,y,z : extended) : boolean;

implementation

function LettersIn (s : string) : boolean;

var

Chisla : string;

i : byte;

znak : boolean;

begin

znak:=false;

Chisla := '1234567890,.';

for i := 1 to length(s) do

begin

if pos(s[i],Chisla) = 0 then

begin LettersIn := false; exit; end;

if (s[i] = ',') and (znak = true) then

begin LettersIn := false; exit; end;

if s[i] = ',' then

begin znak := true; end;

end;

if s[1] = ',' then begin LettersIn := false; exit; end;

LettersIn := true;

end;

function Znaks (s : string) : string;

var

i : byte;

begin

for i := 1 to length(s) do

begin

if s[i] = '.' then s[i] := ',';

end;

Znaks := s;

end;

function Interval (x,y,z : extended) : boolean;

begin

if ((x < 1) or (x > 707)) or (x = 5) then

begin Interval := false; exit; end;

if (y < 1) or (y > 707) then

begin Interval := false; exit; end;

if (z < 1) or (z > 707) then

begin Interval := false; exit; end;

Interval := true;

end;

end.

Тестові артефакти обох програмних забезпечень:

 

Висновок:

В ході лабораторної роботи ми здобули знання та вміння користуватися вільним кросплатформовим середовищем розроблення програм Lazarus IDE, а саме – створення програмних продуктів з графічним інтерфейсом. Завдання студента полягало у вивчені взаємодії текстових даних (тобто рядок вводу/виводу) та математичних обчислень (тобто використання IntToStr, StrToFloat, тощо). Також метою виконання роботи було здобування знань, щодо створення та інтегрування власних модулів, яким в ході роботи виступив валідатор. Метою валідатора була перевірка правильності вводу даних, задля недопущення їх до подальших маніпуляцій (адже це може призвести до аварійного завершення програми). Процес валідації даних проходив приблизно наступним чином: дані подавалися до окремого модуля, який, в свою чергу, складався з однієї або декількох функцій. У ці функції подавалися дані, після чого функції перевіряла, чи відповідають вони заданим умовам та знайденим обмеженням. У разі валідності даних – функція повертала позитивний результат, інакше – хибний. У випадку хибного результату – ПЗ виводило користувачу помилку, а процес виконання основної функції зупинявся, що давало змогу запобігти аварійному завершенню. Якщо ж результат був позитивним – продукт продовжував роботи в штатному режимі. Також одним з аспектів роботи з заданим завданням було знаходження неназваних обмежень, як наприклад у роботі з математичною формулою під коренем не може знаходитися від’ємне число. В цілому дана лабораторна робота дала змогу випробовувати себе у написані графічного інтерфейсу, де є набагато більше можливостей, аніж у консольному додатку. Наприклад у графічному додатку меню з вибором можна реалізувати багатьма способами: кнопки, малюнки, тощо, а у консольному доводиться писати окремий елемент коду. Також ми переконалися у необхідності функції створення сторонніх модулів, які потім можна буде включати до наступних проектів такими, як вони написані, або ж трохи відкоригувавши. Це набагато простіше, аніж включати до коду кожного з проектів подібні строки.