# Лабораторна РОбота №9

1. Комплексні числа
2. 17 тиждень.
   1. Розробка програми у середовищі Delphi, що виконує розрахунки з комплексними числами, та відображає їх результати і вигляді графіків.

# Загальні вимоги

Розробіть програму, що розв’язує задачу, яку поставлено у вашому варіанті. Програма повинна:

* являти собою діалоговий додаток;
* підтримувати російсько мовний або україно мовний інтерфейс;
* містити уведення початкових даних за допомогою слайдерів в діапазоні від 0,01 до 100;
* відображати відомості про призначення програми;
* відображати відомості про автора програми;
* містити у коді необхідні пояснення;
* зберігати встановлені користувачем значення параметрів у зовнішньому файлі;
* зчитувати встановлені користувачем значення параметрів при останньому запуску програми із зовнішнього файлу та використовувати їх у якості параметрів за замовчуванням;
* програма повинна будувати логарифмічну амплітудно-частотну характеристику ЛАЧХ (залежність логарифму амплітуди від логарифму частоти) у межах ω=[0,0001 , 10000], W=[-40 дб , 80 дб];
* програма повинна будувати логарифмічну фазо-частотну характеристику ЛФЧХ (залежність фази амплітуди від логарифму частоти) у межах ω=[0,0001 , 10000], φ=[-π , π];
* програма повинна будувати логарифмічну амплітудно-фазову частотну характеристику ЛАФЧХ (залежність логарифму комплексної амплітуди частоти в полярних координатах) у межах W=[-160 дб , 160 дб];

|  |  |
| --- | --- |
| 9 |  |

**Програмний код:**

unit UnitMain;

interface

uses

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

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Menus, VclTee.TeeGDIPlus,

Vcl.StdCtrls, Vcl.Imaging.pngimage, Vcl.ExtCtrls, Vcl.ComCtrls,

VCLTee.TeEngine, VCLTee.TeeProcs, VCLTee.Chart, VCLTee.Series;

type

TFormMain = class(TForm)

mnuMain: TMainMenu;

mnuFil: TMenuItem;

mnuExit: TMenuItem;

ChartMain: TChart;

PageControlMain: TPageControl;

TabSheetLACH: TTabSheet;

TabSheetLFCH: TTabSheet;

TabSheetLAFCH: TTabSheet;

TrackBarK: TTrackBar;

TrackBarT1: TTrackBar;

Label1: TLabel;

Label2: TLabel;

ImageMain: TImage;

EditK: TEdit;

EditT1: TEdit;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Series1: TLineSeries;

Series2: TLineSeries;

Label13: TLabel;

EditT2: TEdit;

Label14: TLabel;

Label15: TLabel;

Label16: TLabel;

Label17: TLabel;

Label18: TLabel;

TrackBarT2: TTrackBar;

mnuHelp: TMenuItem;

mnuAbout: TMenuItem;

TrackBarKsi: TTrackBar;

Label19: TLabel;

Label20: TLabel;

Label21: TLabel;

Label22: TLabel;

Label23: TLabel;

EditKsi: TEdit;

Label24: TLabel;

procedure mnuExitClick(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure TrackBarKChange(Sender: TObject);

procedure PageControlMainChange(Sender: TObject);

procedure TrackBarT1Change(Sender: TObject);

procedure TrackBart2Change(Sender: TObject);

procedure mnuAboutClick(Sender: TObject);

procedure TrackBarKsiChange(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

TComplex = class

private

Re:single; //действительная часть

Im:single; //мнимая часть

public

constructor Create(R,I:single);

function Modul:single;

function Angle:single;

end;

var

FormMain: TFormMain;

implementation

{$R \*.dfm}

const

n=500; //количество точек на графике

Omin=0.0001; //минимум...

Omax=10000; //...и максимум изменения частоты

Wmin=0.01; //минимум...

Wmax=10000; //...и максимум изменения амплитуды

var

T1,T2,K,ksi:single; //коэфициенты

CurrTab:integer; //номер текущей вкладки

x,y:array[0..n] of single; //массив точек графика

constructor TComplex.Create(r,i:single); //задаем инициализацию комплексного числа

begin

inherited Create;

Re:=r;

Im:=i;

end;

function Add(p,q:TComplex):TComplex; //сложение двух чисел

begin

Result:=TComplex.Create(p.Re+q.Re,p.Im+q.Im);

end;

function Mult(p,q:TComplex):TComplex; //умножение двух чисел

begin

Result:=TComplex.Create(P.Re\*Q.Re-P.Im\*Q.Re,P.Im-Q.Im);

end;

function Divide(p,q:TComplex):TComplex; //деление двух чисел

var m:single; //знаменатель

Re:single; //действительная часть знаменателя

Im:single; //мнимая часть знаменателя

begin

m:=sqr(q.Re)+sqr(q.Im);

if m=0 then

begin

ShowMessage('Возможность деления на ноль не предусмотрена');

Result:=TComplex.Create(1,0);

exit;

end;

Re:=p.Re\*q.Re+p.Im\*q.Im;

Im:=p.Im\*q.Re-p.Re\*q.Im;

Result:=TComplex.Create(Re/m,Im/m);

end;

function TComplex.Modul:single; //модуль комплексного числа

begin

Result:=sqrt(sqr(Re)+sqr(Im));

end;

function TComplex.Angle:single; //угол между векторами

var f:single; //угол

begin

if Re=0 then

if Im<0 then

begin

Result:=-pi/2; exit

end

else

begin

Result:=pi/2; exit

end;

f:=arctan(Im/Re);

if Re<0 then

if Im<0 then

f:=f-pi

else

f:=f+pi;

Result:=f;

end;

function PerFun(p:TComplex):TComplex; //задаем основную функцию

var

chiselnik,znamennik,t2p,sqrt2p,ksi2,ksi2t2p:TComplex;

begin

chiselnik:=Mult(TComplex.Create(K,0),Add(Mult(TComplex.Create(T1,0),p),TComplex.Create(1,0))); //K\*(T1\*p+1)

T2P:=Mult(TComplex.Create(T2,0),p);//T2\*p

sqrT2P:=Mult(T2P,T2P);// (T2\*p)^2

Ksi2:=Mult(TComplex.Create(2,0),TComplex.Create(Ksi,0));// 2\*Ksi

Ksi2T2P:=Mult(Ksi2,T2P);//2\*Ksi\*T2p

znamennik:=Add(Add(sqrt2p,ksi2t2p),TComplex.Create(1,0));//(t2p)^2+2\*Ksi\*T2p+1

Result:=Divide(chiselnik,znamennik);

end;

function Lg(x:single):single; //вычисление lg

begin

Result:=Ln(x)/2.302585;

end;

function Step(x:single):single; //вычисление степени 10

begin

Result:=exp(x\*2.302585);

end;

procedure DrawCoord; //рисуем систему координат

begin

With FormMain.ChartMain do

begin

Series[0].Clear;

case CurrTab of

0:

begin

Series[0].AddXY(Lg(Omin),0,'',clBlack);

Series[0].AddXY(0,0,'',clBlack);

Series[0].AddXY(0,20\*Lg(Wmax),'',clBlack);

Series[0].AddXY(0,20\*Lg(Wmin),'',clBlack); Series[0].AddXY(0,0,'',clBlack); Series[0].AddXY(Lg(Omax),0,'',clBlack);

end;

1:

begin

Series[0].AddXY(Lg(Omin),0,'',clBlack);

Series[0].AddXY(0,0,'',clBlack);

Series[0].AddXY(0,pi,'',clBlack);

Series[0].AddXY(0,-pi,'',clBlack);

Series[0].AddXY(0,0,'',clBlack);

Series[0].AddXY(Lg(Omax),0,'',clBlack);

end;

2:

begin

Series[0].AddXY(-40\*Lg(Wmax),0,'',clBlack);

Series[0].AddXY(0,0,'',clBlack);

Series[0].AddXY(0,40\*Lg(Wmax),'',clBlack);

Series[0].AddXY(0,-40\*Lg(Wmax),'',clBlack);

Series[0].AddXY(0,0,'',clBlack);

Series[0].AddXY(40\*Lg(Wmax),0,'',clBlack);

end;

end;

end;

end;

procedure Calc; //заполняем массив точками

var i:integer; //номер точки

stp:single; //шаг на логарифмической шкале

f,m:single; //фаза и амплитуда функции

Om:TComplex; //частота

W:TComplex; //комплексное значение функции

begin

stp:=(Lg(Omax)-Lg(Omin))/n;

case CurrTab of

0:

begin

for i:=0 to n do

Begin

x[i]:=Lg(Omin)+i\*stp;

Om:=TComplex.Create(Step(x[i]),0);

W:=PerFun(Om);

y[i]:=20\*Lg(W.Modul);

end;

end;

1:

begin

for i:=0 to n do

Begin

x[i]:=Lg(Omin)+i\*stp;

Om:=TComplex.Create(Step(x[i]),Step(x[i]));

W:=PerFun(Om);

y[i]:=W.Angle;

end;

end;

2:

begin

for i:=0 to n do

Begin

Om:=TComplex.Create(Step(Lg(Omin)+i\*stp),Step(Lg(Omin)+i\*stp));

W:=PerFun(Om);

m:=20\*Lg(W.Modul);

f:=W.Angle;

x[i]:=m\*cos(f);

y[i]:=m\*sin(f);

end;

end;

end;

end;

procedure DrawGraph; //рисуем график

var i:integer;

begin

With FormMain.ChartMain do

begin

Series[1].Clear;

case CurrTab of

0:

begin

for i:=0 to n do

if(y[i]<20\*Lg(Wmax))and(y[i]>20\*Lg(Wmin))then

Series[1].AddXY(x[i],y[i],'',clBlue);

end;

1:

begin

for i:=0 to n do

Series[1].AddXY(x[i],y[i],'',clBlue);

end;

2:

begin

for i:=0 to n do

if(x[i]<40\*lg(Wmax)) and (x[i]>-40\*lg(Wmax)) and (y[i]<40\*lg(Wmax)) and (y[i]>-40\*lg(Wmax)) then

Series[1].AddXY(x[i],y[i],'',clBlue);

end;

end;

end;

end;

procedure TFormMain.FormShow(Sender: TObject);

begin

T1:=1;

T2:=1;

K:=1;

Ksi:=1;

CurrTab:=0;

DrawCoord;

Calc;

DrawGraph;

end;

procedure TFormMain.mnuAboutClick(Sender: TObject);

begin

Application.MessageBox('Программа для показа графика y=ln(x)'+#13+#13+'Coded by Serge Ivanov'+#13+#13+'-Contacts-'+#13+'Email: serega.iv.1407@gmail.com', 'About Program');//Инфо о программе

end;

procedure TFormMain.mnuExitClick(Sender: TObject);

begin

Close;

end;

procedure TFormMain.PageControlMainChange(Sender: TObject);

begin

CurrTab:=PageControlMain.TabIndex;

DrawCoord;

Calc;

DrawGraph;

end;

procedure TFormMain.TrackBarKChange(Sender: TObject);

begin

K:=Step(-TrackBarK.Position/10);

EditK.Text:=FloatToStrF(K,ffFixed,3,2);

DrawCoord;

Calc;

DrawGraph;

end;

procedure TFormMain.TrackBarKsiChange(Sender: TObject);

begin

ksi:=Step(-TrackBarksi.Position/10);

Editksi.Text:=FloatToStrF(ksi,ffFixed,3,2);

DrawCoord;

Calc;

DrawGraph;

end;

procedure TFormMain.TrackBarT1Change(Sender: TObject);

begin

T1:=Step(-TrackBarT1.Position/10);

EditT1.Text:=FloatToStrF(T1,ffFixed,3,2);

DrawCoord;

Calc;

DrawGraph;

end;

procedure TFormMain.TrackBart2Change(Sender: TObject);

begin

t2:=Step(-TrackBarT2.Position/10);

EditT2.Text:=FloatToStrF(t2,ffFixed,3,2);

DrawCoord;

Calc;

DrawGraph;

end;

end.

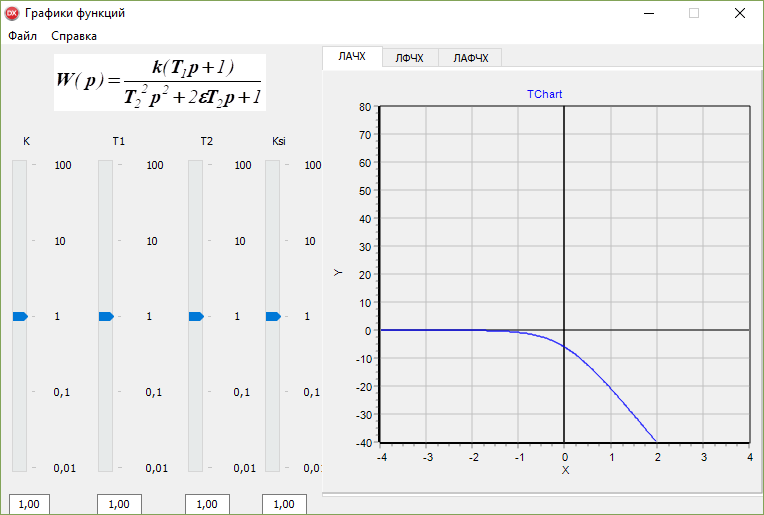


Рис 1. Головне вікно программи

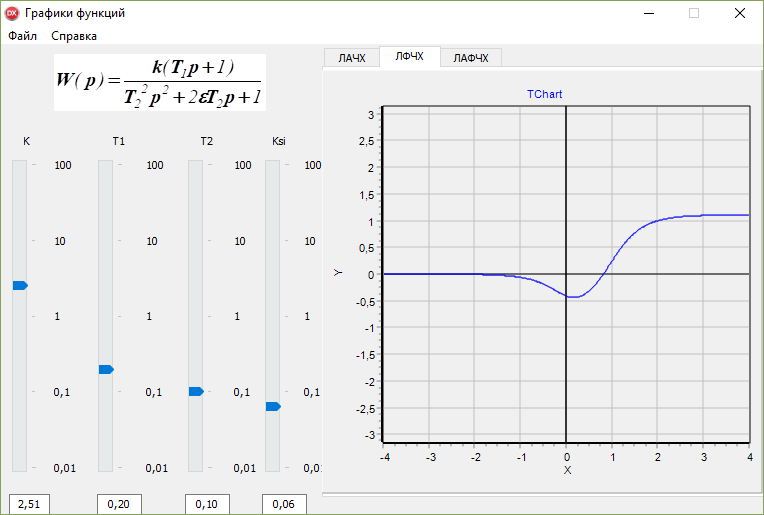


Рис 2. Программа в роботі

**Висновок** : на цій лабораторній роботі я створив програму застосувавши об’єктно-орієнтовну бібліотеку для розробки програмного забезпечення (VCL) Delphi.

|  |  |
| --- | --- |
| **Тип** | **Свойства** |
| TForm | Name, Caption, Height, Width, Position, BorderStyle, BorderIcons. |
| TMainMenu | Name Items. |
| TMenuItem | Name, Caption |
| TLabel | Caption, Top, Left |
| TEdit | Top, Left, Height, Width, Text, ReadOnly |
| TTrackBar | Top, Left, Name, Min, Max, Position |
| TChart | Name, Top, Left, Height, Width, ShowInLegend |