unit myclasses+;

{$mode objfpc}{$H+}{$M+}

interface

uses

Classes, SysUtils, Unit1, Graphics, TAGraph, TASeries, TALegend, Dialogs;

type

TController = class;

TController2 = class;

TStorage = class;

TStorage2 = class;

TBorder = class(TForm1)

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure ComboBox1Change(Sender: TObject);

private

FCon: TController;

FCon2: TController2;

function GetCon: Tcontroller;

procedure SetCon (Value:Tcontroller);

function GetCon2: Tcontroller2;

procedure SetCon2 (Value:Tcontroller2);

procedure PlotResult;

property Con: Tcontroller read GetCon write SetCon;

property Con2: Tcontroller2 read GetCon2 write SetCon2;

public

procedure ConnectController(var aController: TController);

procedure ConnectController2(var aController2: TController2);

end;

// if assigned (Fcon) then Con(self);

TController = class

private

FBor: TBorder;

FStor: TStorage;

function GetBor: TBorder;

procedure SetBor (Value:TBorder);

function GetStor: TStorage;

procedure SetStor (Value:TStorage);

procedure CalcParam( var cp1, cp2, cp3, cp4, cp5, mup, ip, nup, Tnup:real);

property Bor: TBorder read GetBor write SetBor;

property Stor: TStorage read GetStor write SetStor;

public

procedure ConnectBorder(var aBorder: TBorder);

procedure ConnectStorage(var aStorage: TStorage);

end;

TController2 = class

private

FBor: TBorder;

FStor2: TStorage2;

function GetBor: TBorder;

procedure SetBor (Value:TBorder);

function GetStor2: TStorage2;

procedure SetStor2 (Value:TStorage2);

procedure Table(var cp1,cp2,cp3,cp4,cp5,mup,ip,nup :real);

property Bor: TBorder read GetBor write SetBor;

property Stor2: TStorage2 read GetStor2 write SetStor2;

public

procedure ConnectBorder(var aBorder: TBorder);

procedure ConnectStorage2(var aStorage2: TStorage2);

end;

TStorage = class

private

FCon: TController;

function GetCon: Tcontroller;

procedure SetCon (Value:Tcontroller);

procedure SaveData(var css1,css2,css3,css4,css5,muvss,ivss,nuvss,Tnuss :real);

property Con: Tcontroller read GetCon write SetCon;

public

procedure ConnectController(var aController: TController);

end;

TStorage2 = class

private

teta: array [1 .. 10000] of real;

FCon2: TController2;

function GetCon2: Tcontroller2;

procedure SetCon2 (Value:Tcontroller2);

property Con2: Tcontroller2 read GetCon2 write SetCon2;

procedure SaveMassiv;

public

procedure ConnectController2(var aController2: TController2);

end;

var i,j,k,l,m,n,o,p:integer;

implementation

procedure TBorder.ConnectController(var aController: TController); //Подключение контроллера к бордеру

begin

i:=i+1;

if i = 1 then Self.Con := aController;

if i <> 1 then ShowMessage('Контроллер уже подключен');

end;

//

procedure TBorder.ConnectController2(var aController2: TController2); //Подключение контроллера2 к бордеру

begin

j:=j+1;

if j = 1 then Self.Con2 := aController2;

if j <> 1 then ShowMessage('Контроллер2 уже подключен');

end;

//

procedure TController.ConnectBorder(var aBorder: TBorder); //Подключение бордера к контроллеру

begin

k:=k+1;

if k = 1 then Self.Bor := aBorder;

if k <> 1 then ShowMessage('Бордер уже подключен');

end;

//

procedure TController2.ConnectBorder(var aBorder: TBorder); //Подключение бордера к контроллеру2

begin

l:=l+1;

if l = 1 then Self.Bor := aBorder;

if l <> 1 then ShowMessage('Бордер уже подключен');

end;

//

procedure TController.ConnectStorage(var aStorage: TStorage); //Подключение сторэджа к контроллеру

begin

m:=m+1;

if m = 1 then Self.Stor := aStorage;

if m <> 1 then ShowMessage('Сторэдж уже подключен');

end;

//

procedure TStorage.ConnectController(var aController: TController); //Подключение контроллера к сторэдж

begin

n:=n+1;

if n = 1 then Self.Con := aController;

if n <> 1 then ShowMessage('Контроллер уже подключен');

end;

procedure TController2.ConnectStorage2(var aStorage2: TStorage2); //Подключение сторэджа2 к контроллеру2

begin

o:=o+1;

if o = 1 then Self.Stor2 := aStorage2;

if o <> 1 then ShowMessage('Сторэдж2 уже подключен');

end;

//

procedure TStorage2.ConnectController2(var aController2: TController2); //Подключение контроллера2 к сторэдж2

begin

p:=p+1;

if p = 1 then Self.Con2 := aController2 ;

if p <> 1 then ShowMessage('Контроллер2 уже подключен');

end;

//

function TBorder.GetCon:Tcontroller;

begin

result:=FCon;

end;

procedure TBorder.SetCon(Value :Tcontroller);

begin

FCon:=Value;

end;

function TBorder.GetCon2:Tcontroller2;

begin

result:=FCon2;

end;

procedure TBorder.SetCon2(Value :Tcontroller2);

begin

FCon2:=Value;

end;

function TController.GetBor:TBorder;

begin

result:=FBor;

end;

procedure TController.SetBor(Value :TBorder);

begin

FBor:=Value;

end;

function TController.GetStor:TStorage;

begin

result:=FStor;

end;

procedure TController.SetStor(Value :TStorage);

begin

FStor:=Value;

end;

function TController2.GetBor:TBorder;

begin

result:=FBor;

end;

procedure TController2.SetBor(Value :TBorder);

begin

FBor:=Value;

end;

function TController2.GetStor2:TStorage2;

begin

result:=FStor2;

end;

procedure TController2.SetStor2(Value :TStorage2);

begin

FStor2:=Value;

end;

function TStorage.GetCon:Tcontroller;

begin

result:=FCon;

end;

procedure TStorage.SetCon(Value :Tcontroller);

begin

FCon:=Value;

end;

function TStorage2.GetCon2:Tcontroller2;

begin

result:=FCon2;

end;

procedure TStorage2.SetCon2(Value :Tcontroller2);

begin

FCon2:=Value;

end;

procedure TController.CalcParam( var cp1, cp2, cp3, cp4, cp5, mup, ip, nup, Tnup : real);

var

OmegaA, TA, Tv, Kcc, ash, bsh, Kcsh, TAsh: real;

begin

OmegaA := sqrt(cp1 \* cp4 + cp2);

TA := 1 / OmegaA;

Tv := 1 / cp4;

Kcc := cp3 \* cp4 / (cp1 \* cp4 + cp2);

ash := 2 \* (cp1 + cp5 - cp4) / cp3;

bsh := ((cp1 + cp4 + cp5) \* (cp1 + cp4 + cp5) - 4 \* (cp1 \* cp4 + cp2)) /

(cp3 \* cp3);

mup := abs(-ash / 2 + sqrt(ash \* ash / 4 - bsh));

Kcsh := Kcc / (1 + mup \* Kcc);

TAsh := TA / sqrt(1 + mup \* Kcc);

// Расчет передаточных чисел АП

ip := 1 / (Kcsh \* Tv);

nup := 0.09 / (Kcsh \* Tv \* Tv);

Tnup := ip / nup;

end;

//

procedure TController2.Table(var cp1,cp2,cp3,cp4,cp5,mup,ip,nup :real);

var

i: integer;

t1, t2, n, h: real;

al, alpr, wz, wzpr, x1, x1pr, x2, x2pr: real;

begin

t2 := 10;

t1 := 0;

n := 1000; //кол-во шагов

h := (t2 - t1) / n; //шаг по времени

al := pi / 180;

Stor2.teta[1] := pi / 180;

wz := 0; //угловая скорость по крену

x1 := 0;

x2 := -1;

for i := 2 to 10000 do //интегрирование численным методом и получение

begin //значений графика переходного процесса

wzpr := wz;

alpr := al;

x1pr := x1;

x2pr := x2;

wz := wzpr + h \* (-wzpr \* (Cp1 + cp5 + cp3 \* mup) - alpr \* (cp2 - Cp4 \* Cp5)

- Cp3 \* nup \* x1pr - Cp3 \* ip \* x2pr);

x1 := x1pr + h \* x2pr;

al := alpr + h \* (wzpr - alpr \* Cp4);

x2 := x2pr + h \* wzpr;

Stor2.teta[i] := x2 + 1;

end;

end;

//

procedure TBorder.PlotResult;

var

j: integer;

begin

Chart1LineSeries1.Clear;

for j := 1 to 10000 do

Chart1LineSeries1.AddXY(0.01 \* (j - 1), Con2.Stor2.teta[j], '', clRed);

end;

procedure TBorder.Button1Click(Sender: TObject);

var cc1, cc2, cc3, cc4, cc5, muv, iv, nuv, Tnu: real;

begin

cc1 := StrToFloat(Edit1.Text);

cc2 := StrToFloat(Edit2.Text);

cc3 := StrToFloat(Edit3.Text);

cc4 := StrToFloat(Edit4.Text);

cc5 := StrToFloat(Edit5.Text);

muv:=0;

iv:=0;

nuv:=0;

Tnu:=0;

Con.CalcParam(cc1,cc2,cc3,cc4,cc5, muv, iv, nuv, Tnu);

Con2.Table(cc1,cc2,cc3,cc4,cc5,muv,iv,nuv);

PlotResult;

Edit6.Text := FloatToStr(muv);

Edit7.Text := FloatToStr(iv);

Edit8.Text := FloatToStr(nuv);

Edit9.Text := FloatToStr(Tnu);

end;

procedure TStorage.SaveData(var css1,css2,css3,css4,css5,muvss,ivss,nuvss,Tnuss :real);

var

File1: TextFile;

str: string;

begin

AssignFile(File1, 'C:\123.txt');

ReWrite(File1);

str := 'Коэффициенты линейной модели ЛА:';

Writeln(File1, str);

str := 'c1 = ' + FloatToStr(css1);

Writeln(File1, str);

str := 'c2 = ' + FloatToStr(css2);

Writeln(File1, str);

str := 'c3 = ' + FloatToStr(css3);

Writeln(File1, str);

str := 'c4 = ' + FloatToStr(css4);

Writeln(File1, str);

str := 'c5 = ' + FloatToStr(css5);

Writeln(File1, str);

str := 'Передаточные числа АП:';

Writeln(File1, str);

str := 'muv = ' + FloatToStr(muvss);

Writeln(File1, str);

str := 'iv = ' + FloatToStr(ivss);

Writeln(File1, str);

str := 'nuv = ' + FloatToStr(nuvss);

Writeln(File1, str);

str := 'Tnu = ' + FloatToStr(Tnuss);

Writeln(File1, str);

Close(File1);

end;

procedure TStorage2.SaveMassiv;

var

File2: TextFile;

str: string;

i: integer;

begin

AssignFile(File2, 'C:\456.txt');

ReWrite(File2);

str := 'Точки графика:';

Writeln(File2, str);

for i := 1 to 10000 do

begin

str := FloatToStr(i);

Write(File2,str);

Write(File2,' ');

str := FloatToStr(teta[i]);

Writeln(File2,str);

end;

Close(File2);

end;

procedure TBorder.Button2Click(Sender: TObject);

var

cs1,cs2,cs3,cs4,cs5,muvs,nuvs,ivs,Tnus : real;

begin

cs1:=StrToFloat(Edit1.Text);

cs2:=StrToFloat(Edit2.Text);

cs3:=StrToFloat(Edit3.Text);

cs4:=StrToFloat(Edit4.Text);

cs5:=StrToFloat(Edit5.Text);

muvs:=StrToFloat(Edit6.Text);

ivs:=StrToFloat(Edit7.Text);

nuvs:=StrToFloat(Edit8.Text);

Tnus:=StrToFloat(Edit9.Text);

Con.Stor.SaveData(cs1,cs2,cs3,cs4,cs5,muvs,ivs,nuvs,Tnus);

Con2.Stor2.SaveMassiv;

end;

procedure TBorder.Button4Click(Sender: TObject);

begin

Close;

end;

procedure TBorder.ComboBox1Change(Sender: TObject);

var i:integer;

begin

i:=Combobox1.itemindex;

case i of

0: begin Edit1.Text:=FloattoStr(0.762);

Edit2.Text:=FloattoStr(15.45);

Edit3.Text:=FloattoStr(2.83);

Edit4.Text:=FloattoStr(0.841);

Edit5.Text:=FloattoStr(0.309);

end;

1: begin Edit1.Text:=FloattoStr(0.715);

Edit2.Text:=FloattoStr(14.51);

Edit3.Text:=FloattoStr(2.49);

Edit4.Text:=FloattoStr(1.32);

Edit5.Text:=FloattoStr(0.290);

end;

2: begin Edit1.Text:=FloattoStr(0.655);

Edit2.Text:=FloattoStr(13.75);

Edit3.Text:=FloattoStr(3.73);

Edit4.Text:=FloattoStr(0.872);

Edit5.Text:=FloattoStr(0.275);

end;

3: begin Edit1.Text:=FloattoStr(0.879);

Edit2.Text:=FloattoStr(31.55);

Edit3.Text:=FloattoStr(6.26);

Edit4.Text:=FloattoStr(1.41);

Edit5.Text:=FloattoStr(0.356);

end;

4: begin Edit1.Text:=FloattoStr(0.502);

Edit2.Text:=FloattoStr(10.05);

Edit3.Text:=FloattoStr(3.31);

Edit4.Text:=FloattoStr(0.812);

Edit5.Text:=FloattoStr(0.201);

end;

5: begin Edit1.Text:=FloattoStr(0.549);

Edit2.Text:=FloattoStr(12.25);

Edit3.Text:=FloattoStr(3.86);

Edit4.Text:=FloattoStr(0.860);

Edit5.Text:=FloattoStr(0.220);

end;

6: begin Edit1.Text:=FloattoStr(0.502);

Edit2.Text:=FloattoStr(2.645);

Edit3.Text:=FloattoStr(1.22);

Edit4.Text:=FloattoStr(0.801);

Edit5.Text:=FloattoStr(0.192);

end;

7: begin Edit1.Text:=FloattoStr(0.393);

Edit2.Text:=FloattoStr(2.278);

Edit3.Text:=FloattoStr(0.748);

Edit4.Text:=FloattoStr(1.05);

Edit5.Text:=FloattoStr(0.150);

end;

end;

end;

end.

unit Unit1;

{$mode objfpc}{$H+}{$M+}

interface

uses

Classes, SysUtils, FileUtil, TAGraph, TASeries, Forms, Controls, Graphics,

Dialogs, StdCtrls , TALegend;

type

{ TForm1}

TForm1 = class(TForm)

Button1:TButton;

Button2:TButton;

Button3: TButton;

Button4:TButton;

Chart1: TChart;

Chart1LineSeries1: TLineSeries;

ComboBox1: TComboBox;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Edit5: TEdit;

Edit6: TEdit;

Edit7: TEdit;

Edit8: TEdit;

Edit9: TEdit;

Label1: TLabel;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

SaveDialog1: TSaveDialog;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure ComboBox1Change(Sender: TObject);

end;

implementation

{$R \*.lfm}

{ TForm1 }

procedure TForm1.ComboBox1Change(Sender: TObject);

begin

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

end;

procedure TForm1.Button4Click(Sender: TObject);

begin

end;

end.

unit Unit2;

{$mode objfpc}{$H+}{$M+}

interface

uses

Classes, SysUtils, Unit1, Graphics, TAGraph, TASeries, TALegend;

type

TStorage = class;

Tcontroller = class;

TStorage = class

private

FCon: TController;

procedure SaveData(var css1,css2,css3,css4,css5,muvss,ivss,nuvss,Tnuss :real);

property Con: Tcontroller read FCon write FCon;

public

procedure ConnectController(var aController: TController);

end;

Tcontroller = class

private

r:real;

end;

implementation

end.