Министерство образования и науки Российской Федерации

Государственное образовательное учреждение высшего профессионального образования

«Комсомольский – на – Амуре государственный технический университет »

Институт ИКПМТО

Кафедра Технологии машиностроения

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

Расчетная методика проектирования программного комплекса

Студент группы 5МС-1 З.С. Асотова

2006

**Содержание**

Введение

1. Постановка и анализ задачи

1.1 Потребность в идентификации распределения погрешностей

1.2 Оптимальное число интервалов группирования экспериментальных данных

2. Проектирование программного комплекса

3. Кодирование программы

3.1 Модуль Math

3.2 Модуль 1

3.3 Модуль 2

3.4 Модуль 3

3.5 Модуль 4

Заключение

Приложение А

Приложение Б

Приложение В

Приложение Г

Приложение Д

Список использованных источников

**Введение**

Информационные технологии занимают уникальное положение и стали неотъемлемой частью нашей повседневной деятельности. Информационные системы - область науки и техники, без знания которой невозможно дальнейшее развитие современного общества. В отличие от других научно-технических достижений средства вычислительной техники и информатики применяются практически во всех сферах интеллектуальной деятельности человека, способствуя прогрессу в технике и технологии. Конечно, освоение уже имеющихся и хорошо себя зарекомендовавших программных систем и основанных на них технологий - важная и необходимая задача. Нужно создавать необходимые условия для развития в нашей стране исследований в области информационных технологий и реализации их результатов в практических системах и различных приложениях.

**1. Постановка и анализ задачи**

Целью данной работы является разработка программы на языке программирования Delphi для идентификации формы закона распределения погрешностей экспериментальных данных.

**1.1 Потребность в идентификации распределения погрешностей**

Обоснованное решение многих задач требует идентификации формы распределения погрешностей.

Эта идентификация нужна, т.к. рассеяние многих оценок зависит от формы закона распределения. Для обеспечения одной и той же погрешности в определении оценки при одном законе можно ограничиться достаточно малой выборкой, в то время как при другом – выборка исходных данных должна быть значительно больше. Из этого следует, что знание вида закона распределения необходимо для определения одних параметров закона распределения через другие его параметры. Изменение вида закона распределения погрешностей может служить признаком какого-либо изменения условий проведения измерений.

Экспериментальные данные о разнообразии форм распределения погрешностей измерений накоплены в достаточном количестве. В результате этого факт разнообразия законов распределения погрешностей был признан законодательно. 1 января 1974 г. был введен в действие ГОСТ 8.011 – 72, устанавливающий, что при сообщении размера погрешности результата измерения целесообразно указывать вид распределения. Так же были стандартизованы модели равномерного, трапецеидального, треугольного, нормального и двухмодальных распределений.

Надо заметить, что возможность идентификации формы распределения экспериментальных данных ограничена, прежде всего, малостью объема выборки. При большом объеме выборки, например в несколько тысяч наблюдений, построение гистограммы часто позволяет получить достаточно плавную кривую, которая отражает все характерные особенности наблюдаемого закона. Другими словами, более тонкая идентификация формы распределения возможна лишь при соответствующем увеличении объема выборки экспериментальных данных.

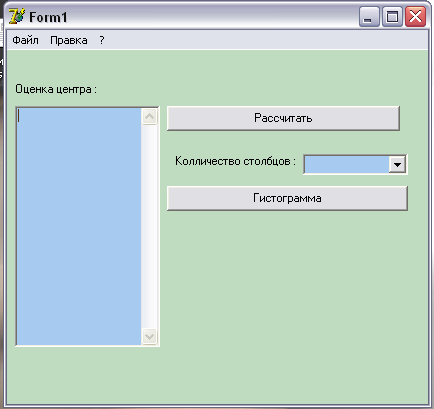
Следовательно, можно сделать вывод о том, что имеет большое значение накопление данных и составление каталога распределения погрешностей различных средств измерений (датчиков, приборов), широко применяемых методов измерений, для того чтобы этими данными можно было затем пользоваться даже при наличии малых серий измерений.

**1.2 Оптимальное число интервалов группирования экспериментальных данных**

Для определения медианы, сгибов, других квантилей, использования критерия согласия Колмогорова-Смирнова или для обнаружения промахов экспериментальные данные необходимо расположить в порядке возрастания, т.е. построить вариационный ряд (упорядоченную выборку).

Для определения формы распределения просто упорядоченной выборки недостаточно, она должна быть представлена в виде гистограммы, состоящей из m столбцов с определенной протяженностью d соответствующих им интервалов. Принято делать эти интервалы одинаковыми. Существует оптимальное число интервалов группирования, когда ступенчатая огибающая гистограммы наиболее близка к плавной кривой распределения. При группировании данных в слишком большое число мелких интервалов гистограмма будет отличаться от плавной кривой распределения вследствие изрезанности многими всплесками и провалами (некоторые интервалы окажутся пустыми или мало заполненными), т.е. будет иметь «гребенчатый» вид. Но при слишком малом числе m интервалов характерные особенности будут потеряны вследствие слишком крупной ступенчатости. Таким образом, оптимальным числом m интервалов является такое, когда максимальное возможное сглаживание случайных данных сочетается с минимальным искажением от сглаживания самой кривой искомого распределения. Для распределения погрешностей одним из практических признаков приближения к оптимуму может служить в гистограмме провалов. Близким к оптимальному считается наибольшее m, при котором гистограмма еще сохраняет плавный характер.

**2 Проектирование программного комплекса**



В данной работе будет использовано 5 модулей и 4 формы.

Первая форма (mainForm) должна отражать дружественный интерфейс, вариационный ряд чисел, количество этих чисел, оценку центра, количество столбцов и две кнопки. Button1 будет называться «Гистограмма», Button2 – «Рассчитать».

При нажатии кнопки Button2 рассчитываются min и max числа ряда, ширина интервала и количество попаданий чисел в интервалы. Для пользователя выводятся на экран варианты количества столбцов для дальнейшего построения гистограммы. По умолчанию предлагается максимальное количество.

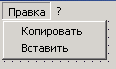
В Delphi имеется два компонента, представляющие меню: MainMenu – главное меню , и PopupMenu – всплывающее меню (страница Standard).



Основное свойство компонентов Items. Его заполнение производится с помощью конструктора меню. Расположим на форме эти компоненты.

1) Двойным щелчком по компоненту MainMenu1 вызовем редактор меню.

2) Задаем свойству Caption значение Файл.

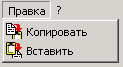


3) Щелчком мышью ниже пункта меню Файл создаем новый пункт Открыть.

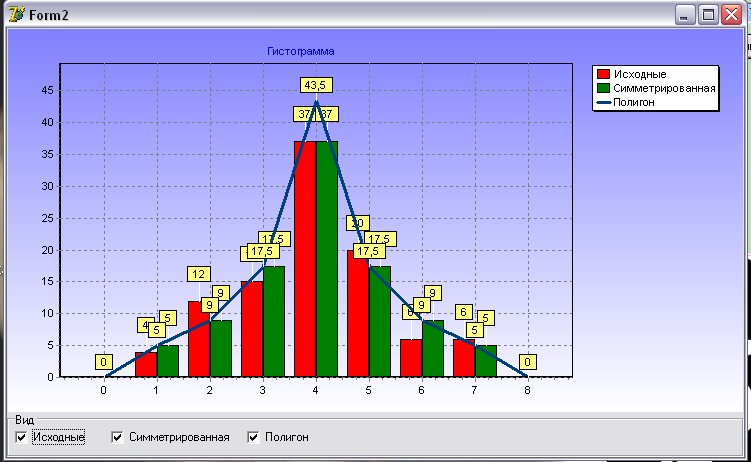
4) Аналогичным способом создаем все пункты меню Файл.

5) Аналогичным способом создаем все пункты меню Правка.

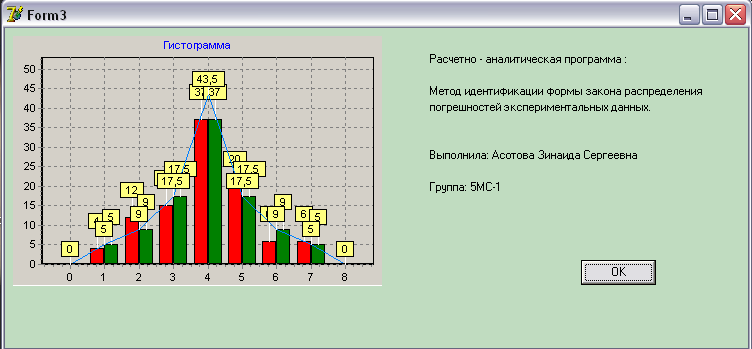
6) Аналогичным способом создаем все пункты меню ?



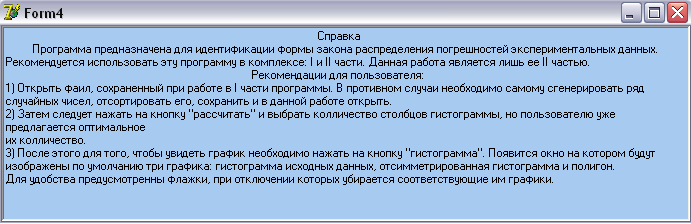
7) Добавляем на форму компонент ImageList1 и заносим в него иконки для пунктов меню. Используя свойство Images компонента MainMenu1, связываем меню с набором иконок. Для некоторых пунктов меню, используя свойство ImageIndex, задаем иконки.



При нажатии кнопки Button1 открывается вторая форма (grafic), которая отображает графики (гистограмма исходных данных, симметрированная гистограмма и полигон). Для удобства внизу формы расположены флажки (CheckBox1, CheckBox2, CheckBox3), а они в свою очередь на панели (компонент GroupBox1). Эти флажки можно как включать так и отключать. По умолчанию, при открытии формы, они активированы.

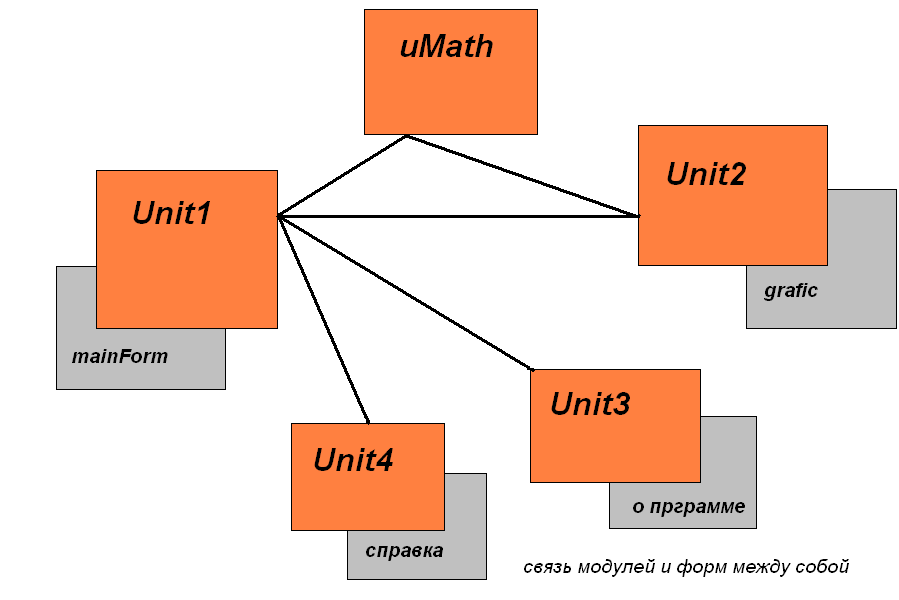


Открывая пункт меню «О программе» появляется третья форма (о программе). На нее помещен компонент Image1 (данная картинка отображает суть программы), кнопка «ОК» (компонент Button1) и краткие данные.



Открывая пункт меню «Справка» появляется четвертая форма (справка). На которую помещен компонент Memo1, в котором отображено как пользоваться данной программой.

**3. Кодирование программы**



**3.1 Модуль Math**

В данном модуле мы описываем все свои математические действия. Здесь мы рассчитываем ширину интервала, количество интервалов, max, min.

Код модуля приведен в приложении А.

**3.2 Модуль 1**

Первый модуль связан непосредственно с первой формой, о которой говорилось выше. Здесь мы описываем дружественный интерфейс, вывод вариационного ряда в компонент Memo1. Причем, нужно учесть, что в файле, где сохранены числа первым указана оценка центра Хс, его мы выводим на форму в Label3. Так же мы описываем здесь расчеты, которые программа выполняет при нажатии на кнопку Button2 и Button3. Описана связь первой формы со второй.

Код модуля приведен в приложении Б.

**3.3 Модуль 2**

В данном модуле программа рассчитывает и строит графики. Связан с модулем 1 и модулем Math

Код модуля приведен в приложении В.

**3.4 Модуль 3**

Этот модуль описывает третью форму, связан с первым модулем.

Код модуля приведен в приложении Г.

**3.5 Модуль 4**

Данный модуль прилагается к четвертой форме, на которой располагается справка.

Код модуля приведен в приложении Д.

**Заключение**

Целью данной работы являлась разработка программы на языке программирования Delphi для идентификации формы закона распределения погрешностей экспериментальных данных. Построение полигона, который более наглядно, чем гистограмма, отражает форму распределения, производилось путем соединения прямыми середин верхних оснований каждого столбца гистограммы. За пределами гистограммы, как слева, так и справа следовали пустые интервалы, в которых точки, соответствующие их серединам, лежали на оси абсцисс. Все эти точки, при построении полигона, соединялись между собой отрезками прямых линий, образуя с осью х замкнутую фигуру. При замене гистограммы кривой в виде полигона выполняется автоматически правило нормирования, т.к. от каждого большого столбца гистограммы (рис.1) отсекается и отбрасывается часть площади в виде треугольников, заштрихованных на рис.1 вертикально, а к каждому меньшему столбцу добавляются такие же площади треугольников, заштрихованных горизонтально. В итого общая площадь под кривой полигона остается равной площади исходной гистограммы. Однако это «перемещение площадей» при переходе от гистограммы к полигону происходит всегда «вниз по склону». В итоге площадь центрального столбца в полигоне оказывается меньше площади центрального столбца гистограммы. Чтобы устранить это явление, мы считали центральный столбец гистограммы состоящим из двух равных столбцов. В этом случае вершина кривой полигона выше верхнего основания центрального столбца, как это показано штриховой линией на рис.1, но площадь под кривой полигона в пределах центрального столбца будет равна площади центрального столбца гистограммы.

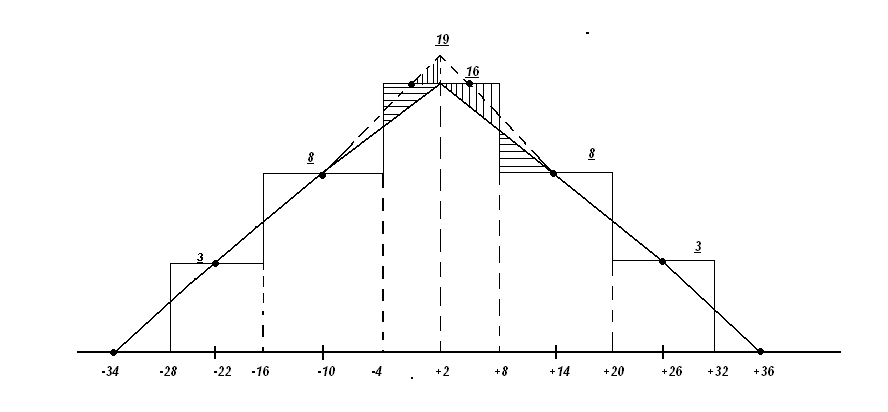


Рис.1

ПРИЛОЖЕНИЕ А

(Модуль Math)

unit uMath;

interface

const

MaxN = 1000;

type

IndexEl = 1..maxN;

TM = array[IndexEl] of Real;

TParam = (lev, prav, kolvo);

var

arrReal : TM; //исходные данные

CountX: Integer; //кол-во исходных чисел

dX: Real; //ширина интервала

MasInt: array [TParam, IndexEl] of Real;

dXmax: IndexEl;

M: integer;

N: integer; //фактрическое кол-во чисел

CountInterv: Integer; //фактическое кол-во интервалов

i: IndexEl;

s: integer;

sa: real;

buf: Real;

imin: IndexEl;

j: IndexEl;

Me: Real;

x: Real;

Mmin: Real;

Mmax: Real;

Xc : Real;//оценка центра

SredInt: Integer;

Max, Min : real;

function fMmin(N: Integer): Integer;

function fMmax(N: Integer): Integer;

function fdX(M: TM; N: Integer; Xc: Real; NInt: Integer): Real;

procedure ChastotaGist;

implementation

uses Math;

function fMmin(N: Integer): Integer;

var

x: Real;

begin

X := 0.55\*power(N,0.4);

Result := round(X);

end;

function fMmax(N: Integer): Integer;

var

x: Real;

begin

X := 1.25\*power(N,0.4);

Result := round(X);

end;

function fdX(M: TM; N: Integer; Xc: Real; NInt: Integer): Real;

var

i: Integer;

x, R: Real;

nGist, NumInt: integer;

Gist: array [1..5] of Integer;

begin

Mmax := M[1];

Mmin := M[1];

for i := 2 to N do

begin

if M[i] > Mmax then Mmax := M[i];

if M[i] < Mmin then Mmin := M[i];

end;

if abs(MMax - Xc) > abs(MMin - Xc) then

x := MMax

else

x := MMin;

Result := (2 \* abs(x - Xc)) / NInt;

end;

procedure ChastotaGist;

var

i, NumInt: Integer;

begin

for i := 1 to CountInterv do

MasInt[kolvo, i] := 0;

for i := 1 to N do

begin

NumInt := Trunc((arrReal[i] - MMin)/dx) + 1;

MasInt[kolvo, NumInt] := MasInt[kolvo, NumInt] + 1;

end;

end;

end.

**Приложение Б** (Модуль 1)

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ImgList, Menus, StdCtrls, ExtCtrls, ExtDlgs, ComCtrls, Math;

type

TForm1 = class(TForm)

MainMenu1: TMainMenu;

PopupMenu1: TPopupMenu;

N1: TMenuItem;

N2: TMenuItem;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N6: TMenuItem;

N7: TMenuItem;

N8: TMenuItem;

N9: TMenuItem;

N10: TMenuItem;

N11: TMenuItem;

ImageList1: TImageList;

N12: TMenuItem;

N13: TMenuItem;

N14: TMenuItem;

Memo1: TMemo;

OpenDialog1: TOpenDialog;

SaveDialog1: TSaveDialog;

N15: TMenuItem;

Button1: TButton;

Button2: TButton;

ComboBox1: TComboBox;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

N16: TMenuItem;

procedure N6Click(Sender: TObject);

procedure N11Click(Sender: TObject);

procedure N4Click(Sender: TObject);

procedure N3Click(Sender: TObject);

procedure N15Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure N16Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

function TextSaved: Boolean;

end;

var

Form1: TForm1;

implementation

uses Unit2, uMath, Unit3, Unit4;

{$R \*.dfm}

function TForm1.TextSaved: Boolean;

begin

Result := True;

if Memo1.Modified then

case MessageDlg('Save current document?', mtConfirmation, mbYesNoCancel, 0) of

mrYes:

begin

N3Click(Self);

Result := not Memo1.Modified;

end;

mrCancel: Result := False;

end;

end;

procedure TForm1.N6Click(Sender: TObject);

begin

if TextSaved then

begin

Memo1.Lines.Clear;

Caption := 'Text Editor';

SaveDialog1.FileName := '';

end;

end;

procedure TForm1.N11Click(Sender: TObject);

begin

Form3.Show;

end;

procedure TForm1.N4Click(Sender: TObject);

begin

with SaveDialog1 do

if Execute then

begin

Memo1.Lines.SaveToFile(FileName);

Memo1.Modified := False;

// - сохранение текста в файле

Caption := 'Text Editor - ' + ExtractFileName(FileName);

end;

end;

procedure TForm1.N3Click(Sender: TObject);

begin

with SaveDialog1 do

if FileName = '' then

N4Click(Self)

else

begin

Memo1.Lines.SaveToFile(FileName);

Memo1.Modified := False;

end;

end;

procedure TForm1.N15Click(Sender: TObject);

begin

Close;

end;

procedure TForm1.N2Click(Sender: TObject);

var

i:Integer;

f: System.Text;

sX: string;

X: Real;

begin

with OpenDialog1 do

if Execute then

begin

AssignFile(f, OpenDialog1.FileName);

Reset(f);

ReadLn(f, sX);

X := StrToFloat(sX);

Xc := X;

i := 0;

while not EOF(f) do

begin

i := i + 1;

ReadLn(f, sX);

X := StrToFloat(sX);

arrReal[i] := X;

end;

N := i;

CloseFile(f);

Label2.Caption := FloatToStr(Xc);

Label4.Caption := format('Кол-во: %d',[N]);

Memo1.Lines.Clear;

for i:=1 to N do

Memo1.Lines.Add(format('%f',[arrReal[i]]));

Caption := 'Text Editor - ' + ExtractFileName(FileName);

end;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

Form2.Raschet;

Form2.Show;

end;

procedure TForm1.Button2Click(Sender: TObject);

var

i, min, max: Integer;

begin

min := fMmin(N);

max := fMmax(N);

ComboBox1.Clear;

for i := min to max do

if Odd(i) then

ComboBox1.Items.Add(format('%d',[i]));

ComboBox1.ItemIndex := ComboBox1.Items.Count - 1;

end;

procedure TForm1.N16Click(Sender: TObject);

begin

Form4.Show;

end;

end.

**Приложение В** (Модуль 2)

unit Unit2;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, TeEngine, Series, ExtCtrls, TeeProcs, Chart, Math, StdCtrls;

const

MaxN = 1000;

type

TForm2 = class(TForm)

Chart1: TChart;

Series1: TBarSeries;

Series2: TBarSeries;

Series3: TLineSeries;

GroupBox1: TGroupBox;

CheckBox1: TCheckBox;

CheckBox2: TCheckBox;

CheckBox3: TCheckBox;

procedure CheckBox1Click(Sender: TObject);

procedure CheckBox2Click(Sender: TObject);

procedure CheckBox3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

procedure Raschet;

end;

var

Form2: TForm2;

implementation

uses Unit1, uMath;

{$R \*.dfm}

{ TForm2 }

procedure TForm2.Raschet;

var

sum: Real;

i : integer;

a, Centr : Real;

begin

CountInterv:= StrToInt(Form1.ComboBox1.Items[Form1.ComboBox1.ItemIndex]);

SredInt := (CountInterv div 2) + 1;

dX := fdX(arrReal, N, Xc, CountInterv);

ChastotaGist;

Chart1.Series[0].Clear;

Chart1.Series[1].Clear;

Chart1.Series[2].Clear;

//исходные

for i := 1 to CountInterv do

Chart1.Series[0].AddXY(i,MasInt[kolvo, i]);

//симмметрированные

for i := SredInt+1 to CountInterv do

begin

sum := MasInt[kolvo, i] + MasInt[kolvo, CountInterv+1-i];

MasInt[kolvo, i] := sum /2;

MasInt[kolvo, CountInterv+1-i] := sum /2;

end;

for i := 1 to CountInterv do

Chart1.Series[1].AddXY(i,MasInt[kolvo, i]);

//полигон

a := abs(MasInt[kolvo, SredInt] - MasInt[kolvo, SredInt - 1]);

a := a / 3 \* 4;

Centr := MasInt[kolvo, SredInt - 1] + a;

Chart1.Series[2].Clear;

Chart1.Series[2].AddXY(0,0);

for i := 1 to CountInterv do

begin

if i <> SredInt then

Chart1.Series[2].AddXY(i,MasInt[kolvo, i])

else

Chart1.Series[2].AddXY(i,Centr);

end;

Chart1.Series[2].AddXY(CountInterv+1,0);

end;

procedure TForm2.CheckBox1Click(Sender: TObject);

begin

Chart1.Series[0].Active := CheckBox1.Checked;

end;

procedure TForm2.CheckBox2Click(Sender: TObject);

begin

Chart1.Series[1].Active := CheckBox2.Checked;

end;

procedure TForm2.CheckBox3Click(Sender: TObject);

begin

Chart1.Series[2].Active := CheckBox3.Checked;

end;

end.

**Приложение** **Г** (Модуль 3)

unit Unit3;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ExtCtrls, StdCtrls;

type

TForm3 = class(TForm)

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Button1: TButton;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

implementation

{$R \*.dfm}

procedure TForm3.Button1Click(Sender: TObject);

begin

Close;

end;

end.

**Приложение Д** (Модуль 4)

unit Unit4;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls;

type

TForm4 = class(TForm)

Memo1: TMemo;

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form4: TForm4;

implementation

{$R \*.dfm}

end.

**Список использованных источников**

1. Новицкий П.В. Оценка погрешностей результатов измерений / П.В. Новицкий, И.А Зограф. - 2-е изд., перераб. – Л.: Энерго – Атомиздат,1991. – 304 с.