**Задание 1. Создание программной системы в Turbo Delphi:**

Выполнить объектную декомпозицию, разработать формы интерфейса, диаграмму состояний интерфейса, диаграммы классов интерфейсной и предметной областей, диаграмму последовательностей одной из реализуемых операций. Разработать, протестировать и отладить программу в среде Turbo Delphi.

Файл содержит сведения о элементах: номер элемента по таблице Менделеева, название элемента, атомный вес, проводимость (проводник, изолятор, полупроводник). Программа должна в интерактивном режиме формировать файл, добавлять и удалять данные, а также воспринимать каждый из перечисленных запросов и давать на него ответ.

1. Найти названия и атомные веса всех полупроводников.

2. Вывести сведения о проводниках в порядке возрастания их атомных весов.

1. Определить названия всех изоляторов, атомные веса которых не превышают заданного значения.
2. Построить график зависимости атомного веса элемента от его номера.

**Текст программы:**

unit Mz;

interface

uses

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

Dialogs, StdCtrls, Addit, Del, SemiCon, Cond, Ins, Graph, All, XpMan;

type

TForm1 = class(TForm)

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Button6: TButton;

Button7: TButton;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure Button7Click(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

begin

form2.visible:=true;

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

form3.visible:=true;

end;

procedure TForm1.Button3Click(Sender: TObject);

begin

form4.visible:=true;

end;

procedure TForm1.Button4Click(Sender: TObject);

begin

form5.visible:=true;

end;

procedure TForm1.Button5Click(Sender: TObject);

begin

form6.visible:=true;

end;

procedure TForm1.Button6Click(Sender: TObject);

begin

form7.Visible:=true;

end;

procedure TForm1.Button7Click(Sender: TObject);

begin

form8.visible:=true;

end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);

begin

closefile(men);

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

assignfile(men,'YAY.pas');

reset(men);

end;

end.

unit Addit;

interface

uses

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

Dialogs, StdCtrls, Del;

type

TForm2 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Edit5: TEdit;

Button1: TButton;

Button2: TButton;

Error: TMemo;

procedure Button1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure ErrorClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

{$R \*.dfm}

var adel:elem;

procedure TForm2.Button1Click(Sender: TObject);

var key:boolean;

bun:elem;

begin

key:=false;

if (edit5.Text<>'') and (edit2.Text<>'') and (edit4.Text<>'') then

begin

seek(men,0);

while (not eof(men)) and (key=false) do

begin

read(men,bun);

if (bun.num=strtoint(edit5.text)) then

key:=true;

end;

if (key=true) then

begin

error.Visible:=true;

button1.Enabled:=false;

end

else

begin

adel.name:=edit2.Text;

adel.num:=strtoint(edit5.Text);

adel.wei:=StrToFloat(edit4.Text);

adel.con:=edit3.Text;

write(men,adel);

edit5.Text:='';

edit2.Text:='';

edit4.Text:='';

end;

end;

end;

procedure TForm2.Button2Click(Sender: TObject);

begin

edit3.ReadOnly:=false;

if (edit3.text='Conductor') then

edit3.Text:='Semiconductor'

else if (edit3.Text='Semiconductor') then

edit3.Text:='Insulator'

else edit3.Text:='Conductor';

edit3.ReadOnly:=true;

end;

procedure TForm2.ErrorClick(Sender: TObject);

begin

error.Visible:=false;

button1.Enabled:=true;

end;

procedure TForm2.FormCreate(Sender: TObject);

begin

edit3.ReadOnly:=true;

end;

end.

unit Del;

interface

uses

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

Dialogs, StdCtrls;

type

TForm3 = class(TForm)

Label2: TLabel;

Edit2: TEdit;

Button1: TButton;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

type elem = record

name:string[15];

num:integer;

wei:real;

con:string[15];

end;

var

Form3: TForm3;

men: file of elem;

implementation

{$R \*.dfm}

procedure TForm3.Button1Click(Sender: TObject);

var key:boolean;

sel:elem;

pde:integer;

begin

seek(men,0);

key:=false;

if (edit2.Text<>'') then

begin

while (not eof(men)) and (key=false) do

begin

read(men,sel);

if (sel.name=edit2.Text) then

key:=true;

end;

if (key=true) then

begin

pde:=filepos(men)-1;

while (not eof(men)) do

begin

read(men,sel);

seek(men,pde);

write(men,sel);

pde:=filepos(men);

seek(men,filepos(men)+1);

end;

seek(men,filepos(men)-1);

truncate(men);

edit2.Text:='';

end;

end;

end;

end.

unit All;

interface

uses

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

Dialogs, Del, StdCtrls;

type

TForm8 = class(TForm)

procedure FormShow(Sender: TObject);

procedure FormHide(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form8: TForm8;

implementation

{$R \*.dfm}

var res:array of tlabel;

siz:integer;

procedure TForm8.FormHide(Sender: TObject);

var i:integer;

begin

for i := 0 to siz-1 do

begin

res[i].caption:='';

res[i].Free;

end;

end;

procedure TForm8.FormShow(Sender: TObject);

var sor:array of elem;

i,x:integer;

begin

seek(men,0);

siz:=filesize(men);

setlength(sor,siz);

setlength(res,siz);

x:=10;

for i := 0 to siz-1 do

begin

read(men,sor[i]);

res[i]:=tlabel.Create(form8);

with res[i] do

begin

parent:=form8;

top:=x;

left:=10;

caption:='Number: '+inttostr(sor[i].num)+' | Name: '+sor[i].name;

caption:=caption+' | Conduction: '+(sor[i].con);

caption:=caption+' | Atomic weight: '+floattostr(sor[i].wei);

end;

x:=x+20;

end;

end;

end.

unit Cond;

interface

uses

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

Dialogs, StdCtrls, Del;

type

TForm5 = class(TForm)

procedure FormShow(Sender: TObject);

procedure FormHide(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form5: TForm5;

implementation

{$R \*.dfm}

var res:array of tlabel;

procedure TForm5.FormHide(Sender: TObject);

var i:integer;

begin

for i := 0 to filesize(men)-1 do

if (res[i].caption<>'') then

res[i].caption:='';

end;

procedure TForm5.FormShow(Sender: TObject);

var sor:array of elem;

i,j,x:integer;

m:boolean;

b:elem;

begin

seek(men,0);

setlength(sor,filesize(men));

for i := 0 to filesize(men)-1 do

read(men,sor[i]);

i:=0;

m:=true;

while (i<=filesize(men)-2) and (m=true) do

begin

m:=false;

for j := 0 to filesize(men)-i-2 do

if (sor[j].wei>sor[j+1].wei) then

begin

b:=sor[j];

sor[j]:=sor[j+1];

sor[j+1]:=b;

m:=true;

end;

end;

setlength(res,filesize(men));

x:=10;

for i := 0 to filesize(men)-1 do

if (sor[i].con='Conductor') then

begin

res[i]:=tlabel.Create(form5);

with res[i] do

begin

parent:=form5;

Top:=x;

Left:=10;

caption:='Name: '+sor[i].name+' | Number: '+inttostr(sor[i].num);

caption:=caption+' | Atomic weight: '+floattostr(sor[i].wei);

end;

x:=x+20;

end;

end;

end.

unit SemiCon;

interface

uses

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

Dialogs, StdCtrls, Del;

type

TForm4 = class(TForm)

procedure FormShow(Sender: TObject);

procedure FormHide(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form4: TForm4;

implementation

{$R \*.dfm}

var res:array of tlabel;

procedure TForm4.FormHide(Sender: TObject);

var i:integer;

begin

for i := 0 to filesize(men)-1 do

if (res[i].caption<>'') then

res[i].caption:='';

end;

procedure TForm4.FormShow(Sender: TObject);

var cur:elem;

i,x:integer;

begin

setlength(res,filesize(men));

seek(men,0);

i:=0;

x:=10;

while (not eof(men)) do

begin

read(men,cur);

if (cur.con='Semiconductor') then

begin

res[i]:=tlabel.Create(form4);

with res[i] do

begin

parent:=form4;

Top:=x;

Left:=10;

caption:='Name: '+cur.name+' | Atomic weight: '+floattostr(cur.wei);

end;

x:=x+20;

inc(i);

end;

end;

end;

end.

unit Ins;

interface

uses

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

Dialogs, Del, StdCtrls;

type

TForm6 = class(TForm)

Edit1: TEdit;

Button1: TButton;

Label1: TLabel;

procedure Button1Click(Sender: TObject);

procedure FormHide(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form6: TForm6;

implementation

{$R \*.dfm}

var res:array of tlabel;

procedure TForm6.Button1Click(Sender: TObject);

var i,x:integer;

cur:elem;

begin

if (edit1.Text<>'') then

begin

setlength(res,filesize(men));

for i := 0 to filesize(men)-1 do

if (res[i].Caption<>'') then

begin

res[i].Caption:='';

res[i].Free;

end;

i:=0;

x:=70;

seek(men,0);

while (not eof(men)) do

begin

read(men,cur);

if (cur.con='Insulator') and (cur.wei<=strtofloat(edit1.text)) then

begin

res[i]:=tlabel.create(form6);

with res[i] do

begin

parent:=form6;

top:=x;

left:=10;

caption:='Name: '+cur.name;

end;

x:=x+20;

inc(i);

end;

end;

end;

end;

procedure TForm6.FormHide(Sender: TObject);

var i:integer;

begin

setlength(res,filesize(men));

for i := 0 to filesize(men)-1 do

if (res[i].Caption<>'') then

res[i].Caption:='';

edit1.Text:='';

end;

end.

unit Graph;

interface

uses

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

Dialogs, ExtCtrls, Del, StdCtrls;

type

TForm7 = class(TForm)

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

procedure FormShow(Sender: TObject);

procedure FormHide(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form7: TForm7;

implementation

{$R \*.dfm}

var res1:array of tlabel;

res2:array of tlabel;

mem:integer;

procedure TForm7.FormHide(Sender: TObject);

var i:integer;

begin

image1.Canvas.Brush.Color := ClWhite;

image1.Canvas.FillRect(Canvas.ClipRect);

for i := 0 to mem-1 do

begin

res1[i].Caption:='';

res1[i].Free;

res2[i].Caption:='';

res2[i].Free;

end;

end;

procedure TForm7.FormShow(Sender: TObject);

var sor:array of elem;

i,j:integer;

b:elem;

m:boolean;

begin

mem:=filesize(men);

with image1.Canvas do

begin

pen.Color:=clblack;

moveto(4,4);

lineto(0,0);

lineto(0,299);

lineto(943,299);

lineto(939,295);

end;

seek(men,0);

setlength(sor,mem);

for i := 0 to mem - 1 do

read(men,sor[i]);

i:=0;

m:=true;

while (i<=mem-2) and (m=true) do

begin

m:=false;

for j := 0 to mem-i-2 do

if (sor[j].num>sor[j+1].num) then

begin

b:=sor[j];

sor[j]:=sor[j+1];

sor[j+1]:=b;

m:=true;

end;

end;

if (mem>0) then

begin

setlength(res1,mem);

setlength(res2,mem);

image1.Canvas.pen.color:=clred;

image1.canvas.Pen.Width:=2;

image1.Canvas.moveto(sor[0].num\*8,299-round(sor[0].wei));

for i := 0 to mem-1 do

begin

res1[i]:=tlabel.create(form7);

with res1[i] do

begin

parent:=form7;

top:=325;

left:=sor[i].num\*8+26;

caption:=inttostr(sor[i].num);

end;

res2[i]:=tlabel.create(form7);

with res2[i] do

begin

parent:=form7;

top:=299-round(sor[i].wei)+18;

left:=8;

caption:=inttostr(round(sor[i].wei));

end;

with image1.canvas do

begin

lineto(sor[i].num\*8,299-round(sor[i].wei));

Pen.Color:=clgray;

Pen.Width:=1;

LineTo(sor[i].num\*8,299);

MoveTo(sor[i].num\*8,299-round(sor[i].wei));

lineto(0,299-round(sor[i].wei));

MoveTo(sor[i].num\*8,299-round(sor[i].wei));

Pen.Color:=clred;

Pen.Width:=2;

end;

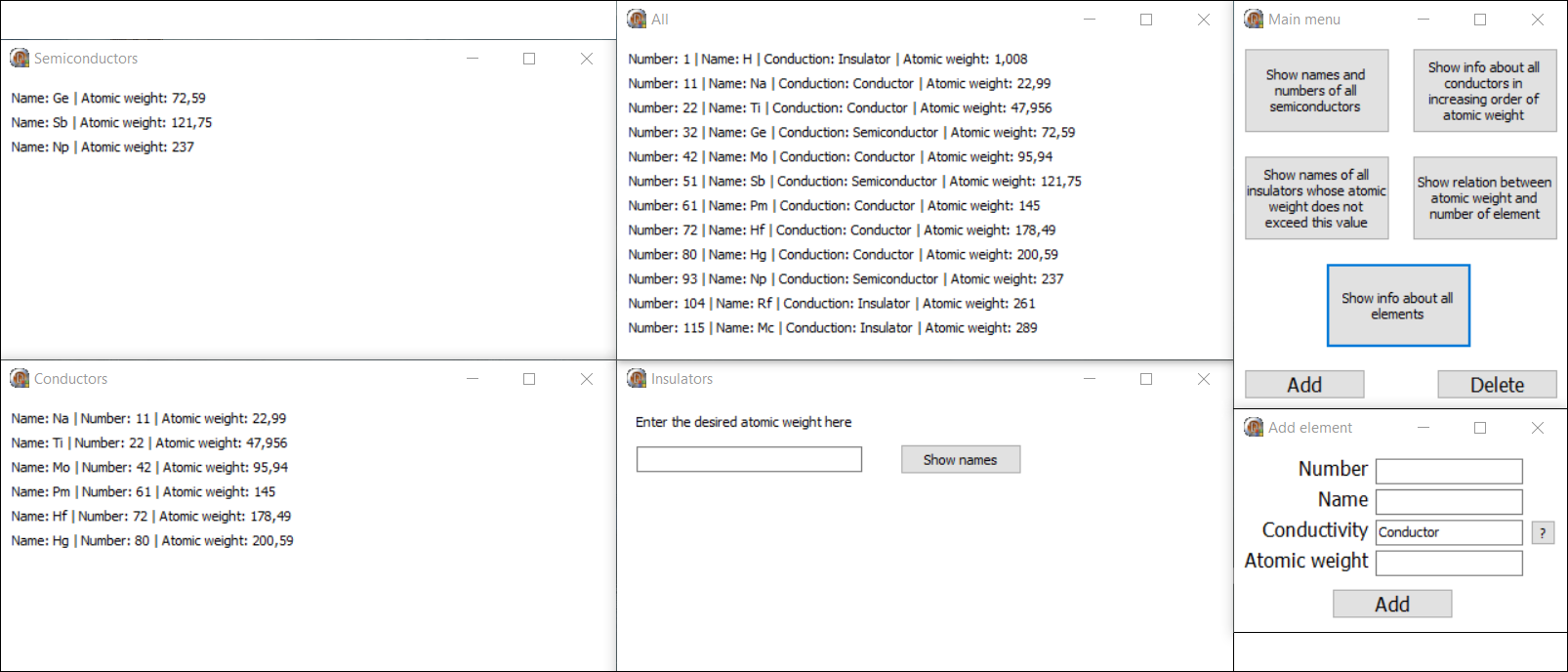
end;

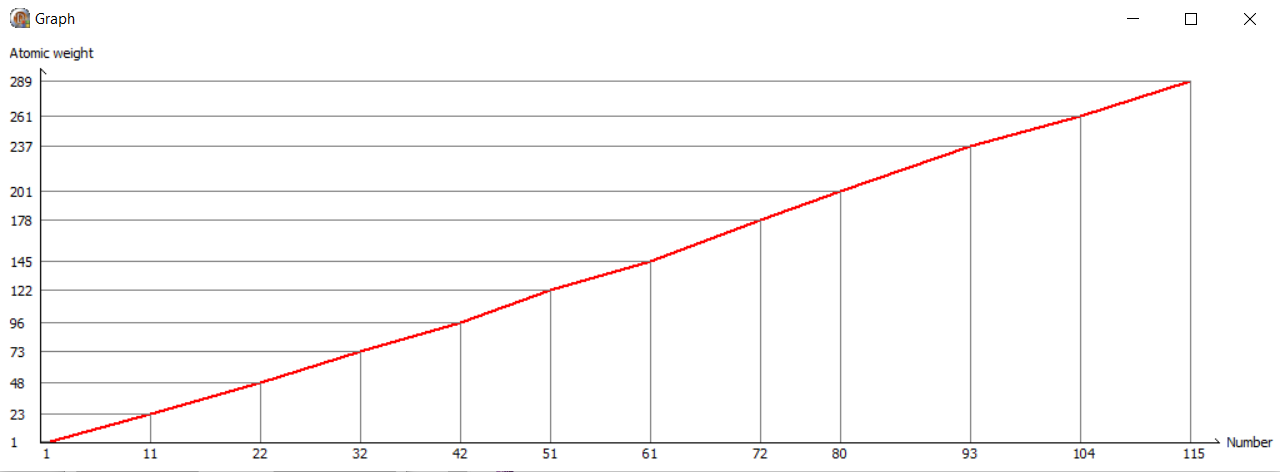
end;

end;

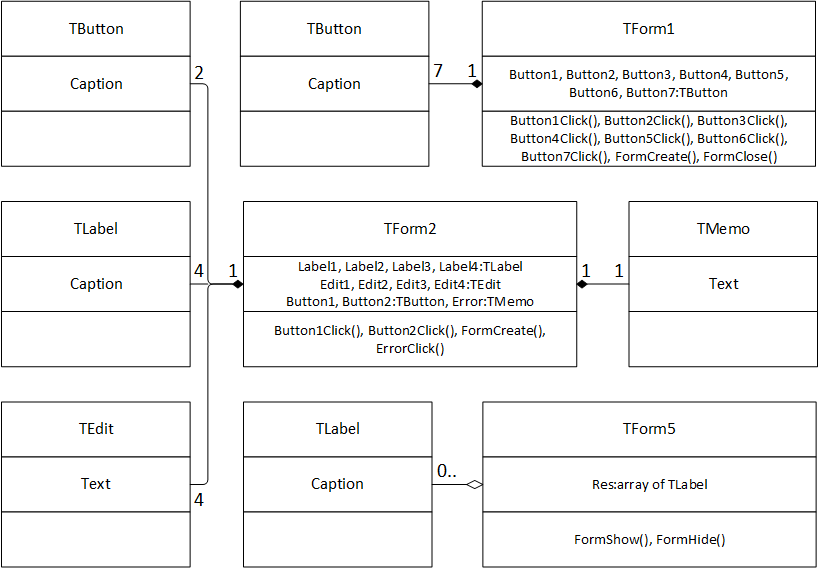
end.

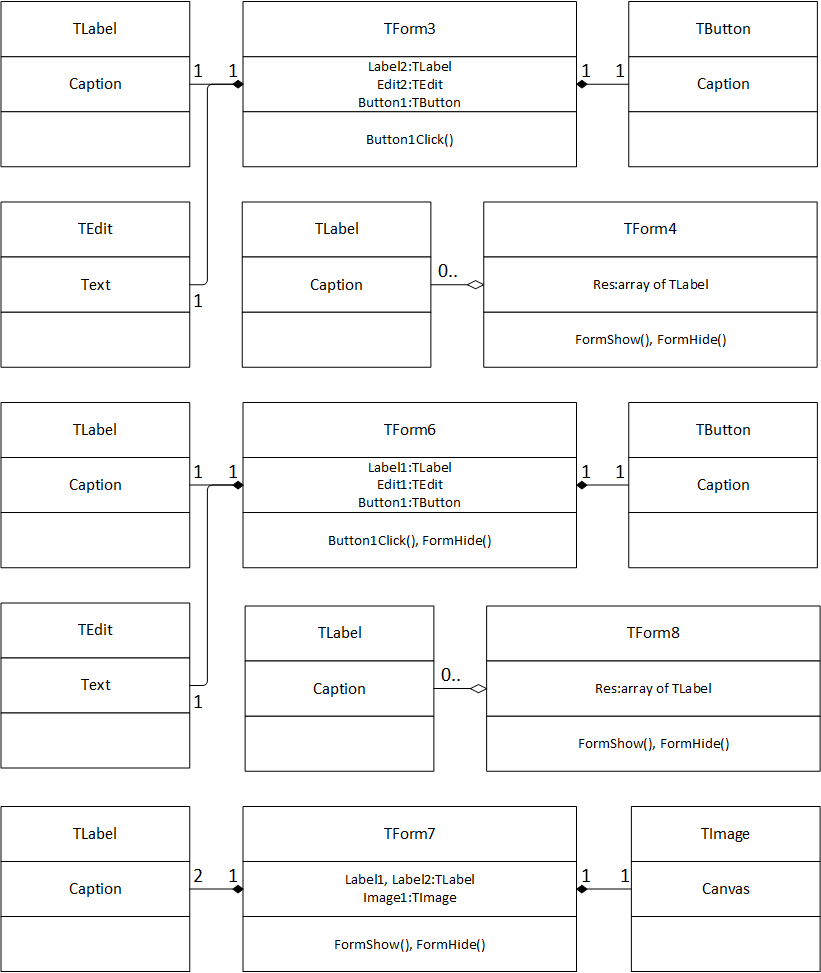
**Скриншоты:**

****

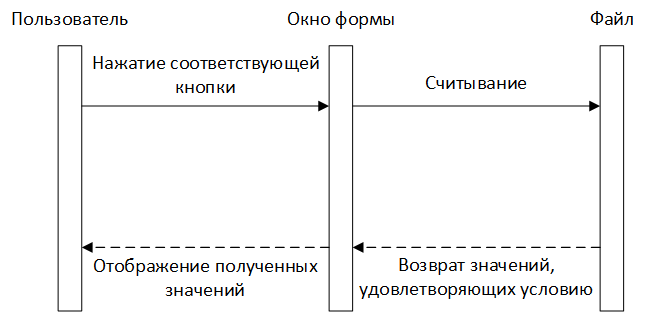
****

**Диаграмма классов:**

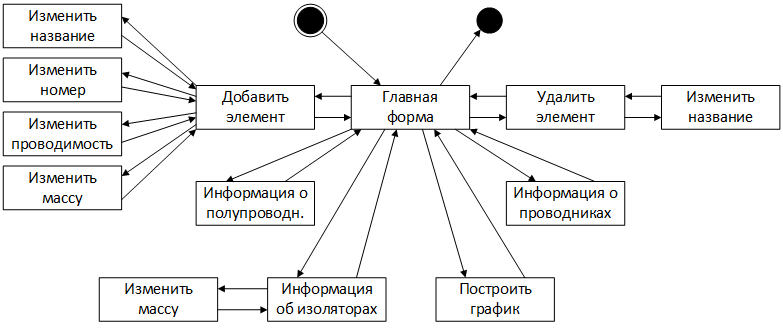
****

****

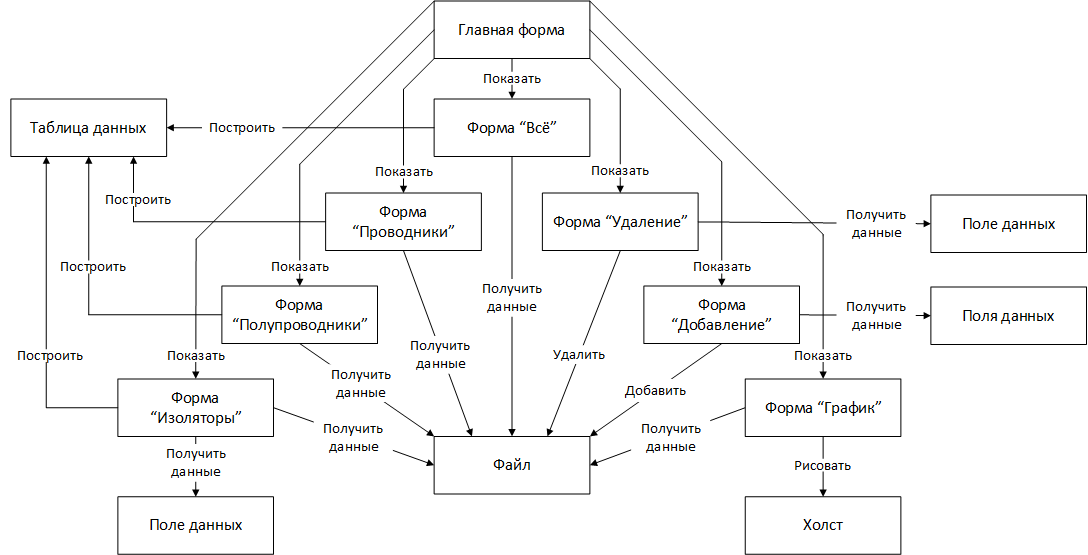
**Диаграмма последовательностей операции нахождения названий и атомных масс всех полупроводников:**

****

**Диаграмма состояний интерфейса:**

****

**Объектная декомпозиция:**

****

**Выводы:**

Delphi предлагает широкий спектр инструментов для создания графических интерфейсов.