|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**Отчет**

|  |  |
| --- | --- |
| **по домашней работе №** | **1** |

**Название:**

Программирование на Object Pascal с использованием классов

**Дисциплина:** Объектно-ориентированное программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22Б |  |  |  |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  |  |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2020

**Часть 1.1. Графический редактор**

**Задание:**

Разработать иерархию классов. Поместить определение классов в отдельном модуле.

Класс, позволяющий рисовать линию от точки, определенной нажатием правой клавиши мыши, до точки ее отпускания.

Класс, позволяющий рисовать окружность от точки, определенной нажатием левой клавиши мыши, до точки ее отпускания.

Цвет фигур задавать с использованием интерфейсных элементов.

В отчете показать иерархии используемых классов VCL и разработанных классов, граф состояния пользовательского интерфейса и объектную декомпозицию.

**Текст программы:**

unit MainForm;

interface

uses

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

Dialogs, ExtCtrls, StdCtrls, Storage;

type

TForm1 = class(TForm)

Image1: TImage;

ColorBox1: TColorBox;

procedure Image1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure Image1MouseUp(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

begin

with image1.Canvas do

begin

pen.color:=clblack;

rectangle(0,0,272,184);

end;

end;

procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

if (button=mbright) then

figl.sinit(x,y,image1)

else

fige.sinit(x,y,image1);

end;

procedure TForm1.Image1MouseUp(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

if (button=mbright) then

begin

figl.einit(x,y);

figl.drawline(colorbox1.Selected);

end

else

begin

fige.einit(x,y);

fige.drawellipse(colorbox1.Selected);

end

end;

end.

unit Storage;

interface

uses ExtCtrls, Graphics;

type figure=object

xs,ys,xe,ye:integer;

can:timage;

procedure sinit(x,y:integer;ic:timage);

procedure einit(x,y:integer);

end;

line=object(figure)

procedure drawline(col:tcolor);

end;

ellipse=object(figure)

procedure drawellipse(col:tcolor);

end;

var figl:line;

fige:ellipse;

implementation

procedure figure.sinit(x: Integer; y: Integer; ic: TImage);

begin

xs:=x;

ys:=y;

can:=ic;

end;

procedure figure.einit(x: Integer; y: Integer);

begin

xe:=x;

ye:=y;

end;

procedure line.drawline(col: TColor);

begin

with can.Canvas do

begin

pen.color:=col;

moveto(xs,ys);

lineto(xe,ye);

end;

end;

procedure ellipse.drawellipse(col: TColor);

begin

with can.Canvas do

begin

pen.color:=col;

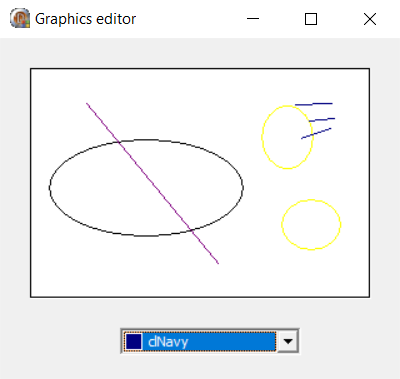
ellipse(xs,ys,xe,ye);

end;

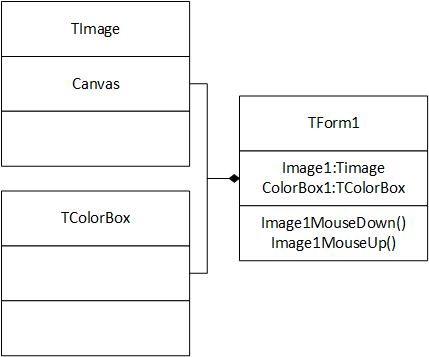
end;

end.

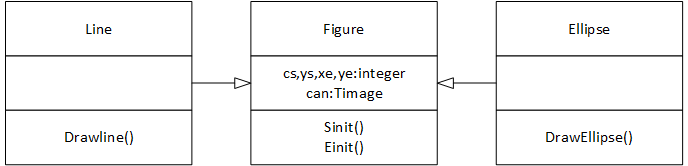
**Скриншоты:**

****

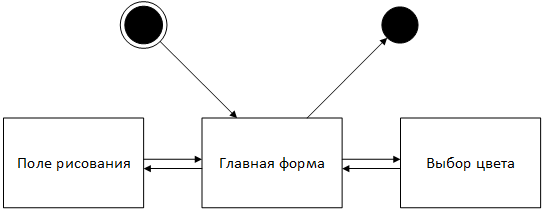
**Диаграмма используемых классов VCL:**

****

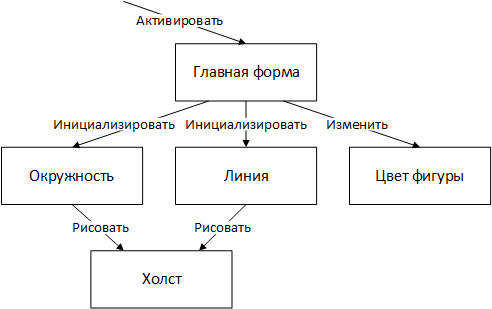
**Диаграмма разработанных классов:**

****

**Диаграмма состояний интерфейса:**

****

**Объектная декомпозиция:**

****

**Часть 1.2. Полиморфное наследование**

**Задание:**

Разработать программу, содержащую описание трех графических объектов: круг с вырезанной четвертью, спираль Архимеда, квадрат.

Реализуя механизм полиморфизма, привести объекты в одновременное вращение вокруг их геометрических центров с различными угловыми скоростями.

В отчете привести диаграмму используемых классов VCL и разработанных классов, граф состояний пользовательского интерфейса и объектную декомпозицию.

**Текст программы:**

unit Main;

interface

uses

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

Dialogs, StdCtrls, ExtCtrls, FigStor;

type

TForm1 = class(TForm)

Image1: TImage;

Timer1: TTimer;

Image3: TImage;

Timer2: TTimer;

Image2: TImage;

Timer3: TTimer;

procedure Timer1Timer(Sender: TObject);

procedure Timer2Timer(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure Timer3Timer(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

var cir:circle;

squ:square;

arc:archi;

t1:real=0;

t2:real=0;

t3:real=0;

procedure TForm1.FormCreate(Sender: TObject);

begin

squ.init(image1);

cir.init(image2);

arc.init(image3);

end;

procedure TForm1.Timer1Timer(Sender: TObject);

begin

squ.draw(t1);

t1:=t1+0.1;

if (t1>=2\*pi) then

t1:=0;

end;

procedure TForm1.Timer2Timer(Sender: TObject);

begin

arc.draw(t2);

t2:=t2+0.1;

if (t2>=2\*pi) then

t2:=0;

end;

procedure TForm1.Timer3Timer(Sender: TObject);

begin

cir.draw(t3);

t3:=t3+0.1;

if (t3>=2\*pi) then

t3:=0;

end;

end.

unit FigStor;

interface

uses extctrls, graphics;

type fig=object

cnv:timage;

procedure init(icn:timage);

procedure draw(n:real);

end;

square=object(fig)

procedure draw(n:real);

end;

circle=object(fig)

procedure draw(n:real);

end;

archi=object(fig)

procedure draw(n:real);

end;

implementation

procedure fig.init(icn: TImage);

begin

cnv:=icn;

end;

procedure fig.draw(n: Real);

begin

//abstract

end;

procedure square.draw;

var a:array [0..3,1..2] of integer;

i:integer;

begin

for i := 0 to 3 do

begin

a[i,1]:=50+round(30\*cos(n+i\*pi/2));

a[i,2]:=50+round(30\*sin(n+i\*pi/2));

end;

with cnv.Canvas do

begin

pen.color:=clwhite;

brush.Color:=clwhite;

rectangle(0,0,100,100);

pen.color:=clred;

moveto(a[3,1],a[3,2]);

for i := 0 to 3 do

lineto(a[i,1],a[i,2]);

end;

end;

procedure circle.draw(n: Real);

var c:array[1..2,1..2] of integer;

i:integer;

begin

for i := 1 to 2 do

begin

c[i,1]:=50-round(30\*cos(n-(i-1)\*3\*pi/2));

c[i,2]:=50+round(30\*sin(n-(i-1)\*3\*pi/2));

end;

with cnv.canvas do

begin

pen.color:=clwhite;

brush.color:=clwhite;

rectangle(0,0,99,99);

pen.color:=clgreen;

pie(20,20,80,80,c[2,1],c[2,2],c[1,1],c[1,2]);

end;

end;

procedure archi.draw(n: Real);

var b:array [1..6,1..6] of integer;

i:integer;

begin

b[1,1]:=50;

b[1,2]:=50;

b[5,1]:=50;

b[5,2]:=50;

for i := 2 to 4 do

begin

if (i=3) then

begin

b[i,1]:=50+round(sqrt(50)\*cos(n-(i-2)\*pi/4));

b[i,2]:=50+round(sqrt(50)\*sin(n-(i-2)\*pi/4));

end

else

begin

b[i,1]:=50+round(5\*cos(n-(i-2)\*pi/4));

b[i,2]:=50+round(5\*sin(n-(i-2)\*pi/4));

end;

end;

b[6,1]:=b[2,1];

b[6,2]:=b[2,2];

for i := 1 to 6 do

begin

b[i,3]:=b[i,1]+round(i\*5\*cos(n+(2\*pi-i\*pi/2)));

b[i,4]:=b[i,2]+round(i\*5\*sin(n+(2\*pi-i\*pi/2)));

b[i,5]:=b[i,1]+round(i\*5\*cos(n+(3\*pi/2-(i mod 4)\*pi/2)));

b[i,6]:=b[i,2]+round(i\*5\*sin(n+(3\*pi/2-(i mod 4)\*pi/2)));

end;

with cnv.Canvas do

begin

pen.color:=clwhite;

brush.color:=clwhite;

rectangle(0,0,100,100);

pen.Color:=clblue;

for i := 1 to 6 do

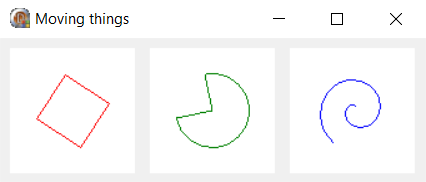
arc(b[i,1]-5\*i,b[i,2]-5\*i,b[i,1]+5\*i,b[i,2]+5\*i,b[i,3],b[i,4],b[i,5],b[i,6]);

end;

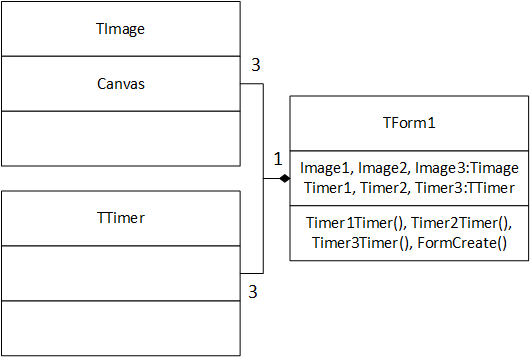
end;

end.

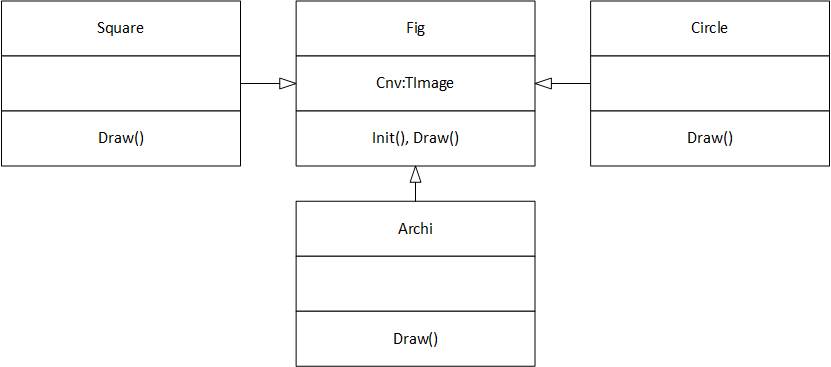
**Скриншоты:**

****

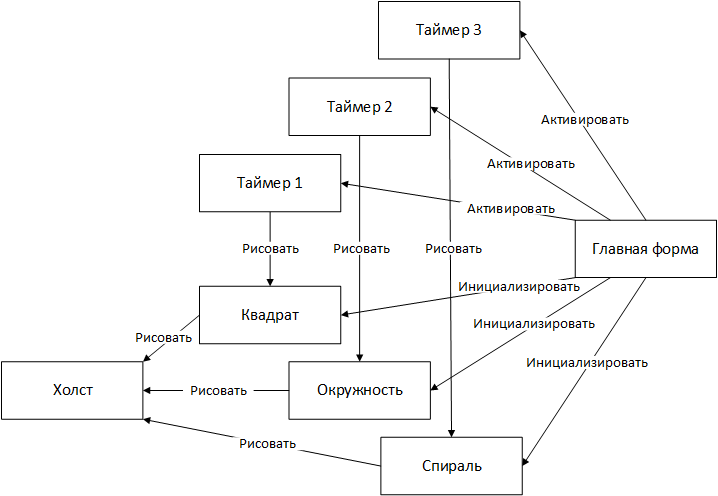
**Диаграмма используемых классов VCL:**

****

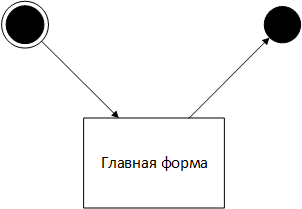
**Диаграмма разработанных классов:**

****

**Объектная декомпозиция:**

****

**Диаграмма состояний интерфейса:**

****

**Вывод:**

Получены навыки программирования на Object Pascal с использованием классов VCL, а также собственно разработанных классов. Отдельно была отработана реализация полиморфного наследования.