МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Аффинные преобразования на плоскости

Отчет

Лабораторная работа №3 по дисциплине

«Компьютерная графика»

Выполнил студент группы ИВТ-23 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Кудяшев Я.Ю./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Коржавина А.С./

Киров 2020

**Цель работы**

Цель работы: закрепить лекционный материал по изучению одноименной

темы, реализовав матрицы переноса, масштабирования, отражения и вращения

применительно к координатам описанной в программе плоской фигуры

(многоугольника) с целью демонстрации движения и преобразования формы этой

фигуры на плоскости.

**Листинг программы**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls, ExtCtrls,

TAIntervalSources, TAGraph, TASeries, Types;

type

{ TForm1 }

TForm1 = class(TForm)

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Button6: TButton;

Button7: TButton;

Button8: TButton;

ToggleBox1: TToggleBox;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure Button7Click(Sender: TObject);

procedure Button8Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure FormMouseWheelDown(Sender: TObject; Shift: TShiftState;

MousePos: TPoint; var Handled: Boolean);

procedure FormMouseWheelUp(Sender: TObject; Shift: TShiftState;

MousePos: TPoint; var Handled: Boolean);

procedure ToggleBox1Change(Sender: TObject);

private

public

end;

var

Form1: TForm1;

implementation

const m=4;{фигур }

n=3;{размер матриц коэффициентов}

type mas=array[1..m] of real;

mtr=array[1..n,1..n] of real;

var xa,ya,xb,yb:mas;{фигуры после перемещения}

a,r:mtr;{матрицы преобразований}

xc,yc:integer;{центр экрана }

ms:real;{масштаб для перевода реальных координат в экранные}

s:string;{строка для вывода текста}

x,y:integer;

{$R \*.lfm}

{ TForm1 }

procedure Osi;{рисование осей координат}

var i:integer;

begin

Form1.Canvas.Pen.Width:=1;

Form1.Canvas.Pen.Color:=clBlack;

Form1.Canvas.MoveTo(0,yc);

Form1.Canvas.LineTo(Form1.Width,yc);

Form1.Canvas.MoveTo(xc,0);

Form1.Canvas.LineTo(xc,Form1.Height);

for i:=1 to 12 do

if i mod 2=0 then

begin

Form1.Canvas.MoveTo(xc-3,yc-round(i\*ms));

Form1.Canvas.LineTo(xc+3,yc-round(i\*ms));

Form1.Canvas.MoveTo(xc-3,yc+round(i\*ms));

Form1.Canvas.LineTo(xc+3,yc+round(i\*ms));

Form1.Canvas.MoveTo(xc+round(i\*ms),yc-3);

Form1.Canvas.LineTo(xc+round(i\*ms),yc+3);

Form1.Canvas.MoveTo(xc-round(i\*ms),yc-3);

Form1.Canvas.LineTo(xc-round(i\*ms),yc+3);

str(i,s);

Form1.Canvas.textout(xc-20,yc-round(i\*ms),s;

Form1.Canvas.textout(xc-25,yc+round(i\*ms),'-'+s);

Form1.Canvas.textout(xc+round(i\*ms),yc+10,s);

Form1.Canvas.textout(xc-round(i\*ms),yc+10,'-'+s);

end;

Form1.Canvas.textout(xc+5,yc+10,'0');

end;

Procedure clear;

var i:integer;

begin

Form1.Canvas.Pen.Color:=clWhite;

for i:=0 to 1000 do

begin

Form1.Canvas.MoveTo(i,0);

Form1.Canvas.LineTo(i,800);

end;

end;

procedure Draw(x,y:mas);

begin

Osi;

Form1.Canvas.Pen.Width:=3;

Form1.Canvas.Pen.Color:=clGreen;

Form1.Canvas.moveto(xc+round(ms\*x[1]),yc-round(ms\*y[1]));

Form1.Canvas.lineto(xc+round(ms\*x[2]),yc-round(ms\*y[2])); //3

Form1.Canvas.lineto(xc+round(ms\*x[3]),yc-round(ms\*y[3])); //2

Form1.Canvas.lineto(xc+round(ms\*x[4]),yc-round(ms\*y[4])); //4

Form1.Canvas.lineto(xc+round(ms\*x[1]),yc-round(ms\*y[1]));

end;

procedure Ed;

var i,j:integer;

begin

for i:=1 to n do

begin

for j:=1 to n do

r[i,j]:=0;

r[i,i]:=1;

end;

end;

procedure Mult;

var b:mtr;

z:real;

i,j,k:integer;

begin

for i:=1 to n do

for j:=1 to n do

begin

z:=0;

for k:=1 to n do

z:=z+a[i,k]\*r[k,j];

b[i,j]:=z

end;

for i:=1 to n do

for j:=1 to n do

r[i,j]:=b[i,j]

end;

procedure Stretch;

var i,j:integer;

begin

for i:=1 to n do

begin

for j:=1 to n do

a[i,j]:=0;

a[i,i]:=1;

end;

a[2,2]:=1.05;

a[1,1]:=1.05;

Mult;

end;

procedure Compress;

var i,j:integer;

for i:=1 to n do

begin

for j:=1 to n do a[i,j]:=0;

a[i,i]:=1;

end;

a[2,2]:=0.95; a[1,1]:=0.95;

Mult;

end;

procedure MirrorX;

var i,j:integer;

begin

for i:=1 to n do

begin

for j:=1 to n do

a[i,j]:=0;

a[i,i]:=1;

end;

a[2,2]:=-1;

Mult;

end;

procedure MirrorY;

var i,j:integer;

begin

for i:=1 to n do

begin

for j:=1 to n do a[i,j]:=0;

a[i,i]:=1;

end;

a[1,1]:=-1;

Mult;

end;

procedure MoveY(dy:integer);

var i,j:integer;

begin

for i:=1 to n do

begin

for j:=1 to n do

a[i,j]:=0;

a[i,i]:=1;

end;

a[2,3]:=dy;

Mult;

end;

procedure MoveX(dx:integer);

var i,j:integer;

begin

for i:=1 to n do

begin

for j:=1 to n do

a[i,j]:=0;

a[i,i]:=1;

end;

a[1,3]:=dx;

Mult;

end;

procedure Rotate(u:real);

var c, s: real;

i,j:integer;

begin

for i:=1 to 3 do

for j:=1 to 3 do

a[i,j]:=0;

a[3,3]:=1;

c:=cos(u); a[1,1]:= c; a[2,2]:=c;

s:=sin(u); a[1,2]:=-s; a[2,1]:=s;

Mult;

end;

procedure New\_XY;

var i:integer;

begin

for i:=1 to m do

begin

xb[i]:=xa[i]\*r[1, 1]+ ya[i]\*r[1, 2]+ r[1, 3];

yb[i]:=xa[i]\*r[2, 1]+ ya[i]\*r[2, 2]+ r[2, 3]

end;

end;

procedure TForm1.FormCreate(Sender: TObject);

var i:integer;

begin

xc:=Form1.Width div 2;

yc:=Form1.Height div 2;

ms:=(yc-30)/10;

xa[1]:=-4;ya[1]:=2;//1

xa[2]:=4;ya[2]:=2;

xa[3]:=4;ya[3]:=-2;

xa[4]:=-4;ya[4]:=-2;//-2

end;

procedure TForm1.FormMouseWheelDown(Sender: TObject; Shift: TShiftState;

MousePos: TPoint; var Handled: Boolean);

begin

clear;

Ed;

Compress; { сжатие}

New\_XY;

Draw(xb,yb);

xa:=xb;

ya:=yb;

end;

procedure TForm1.FormMouseWheelUp(Sender: TObject; Shift: TShiftState;

MousePos: TPoint; var Handled: Boolean);

begin

clear;

Ed;

Stretch; { растяжение}

New\_XY;

Draw(xb,yb);

xa:=xb;ya:=yb;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

begin

clear;

Ed;

Rotate(pi/10); {поворот на pi/4 относительно начала координат}

New\_XY;

Draw(xb,yb);

xa:=xb;ya:=yb;

end;

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

clear;

Ed;

MirrorX; {отражение по Х}

New\_XY;

Draw(xb,yb);

xa:=xb;ya:=yb;

end;

procedure TForm1.Button3Click(Sender: TObject);

begin

clear;

Ed;

MirrorY; {отражение по Y}

New\_XY;

Draw(xb,yb);

xa:=xb;ya:=yb;

end;

procedure TForm1.Button4Click(Sender: TObject);

begin

Form1.Close;

end;

procedure TForm1.Button5Click(Sender: TObject);

begin

clear;

Ed;

MoveY(1); {перемещение по Y}

New\_XY;

Draw(xb,yb);

xa:=xb;ya:=yb;

end;

procedure TForm1.Button6Click(Sender: TObject);

begin

clear;

Ed;

MoveY(-1); { перемещение по Y}

New\_XY;

Draw(xb,yb);

xa:=xb;ya:=yb;

end;

procedure TForm1.Button7Click(Sender: TObject);

begin

clear;

Ed;

MoveX(1); { перемещение по Х}

New\_XY;

Draw(xb,yb);

xa:=xb;ya:=yb;

end;

procedure TForm1.Button8Click(Sender: TObject);

begin

clear;

Ed;

MoveX(-1); { перемещение по Y}

New\_XY;

Draw(xb,yb);

xa:=xb;ya:=yb;

end;

procedure TForm1.ToggleBox1Change(Sender: TObject);

begin

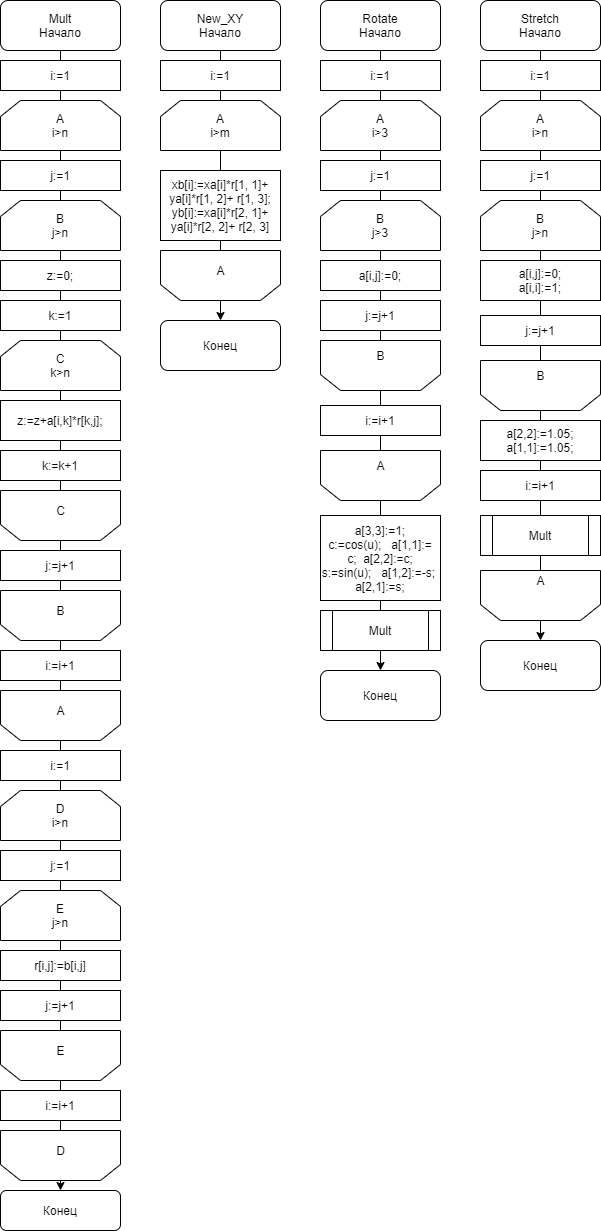
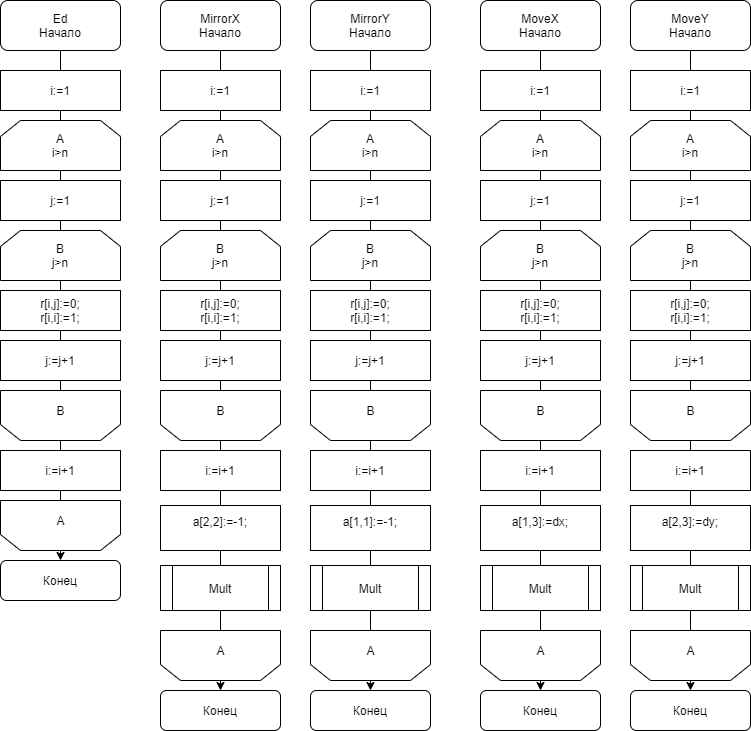
Osi;

Draw(xa,ya);

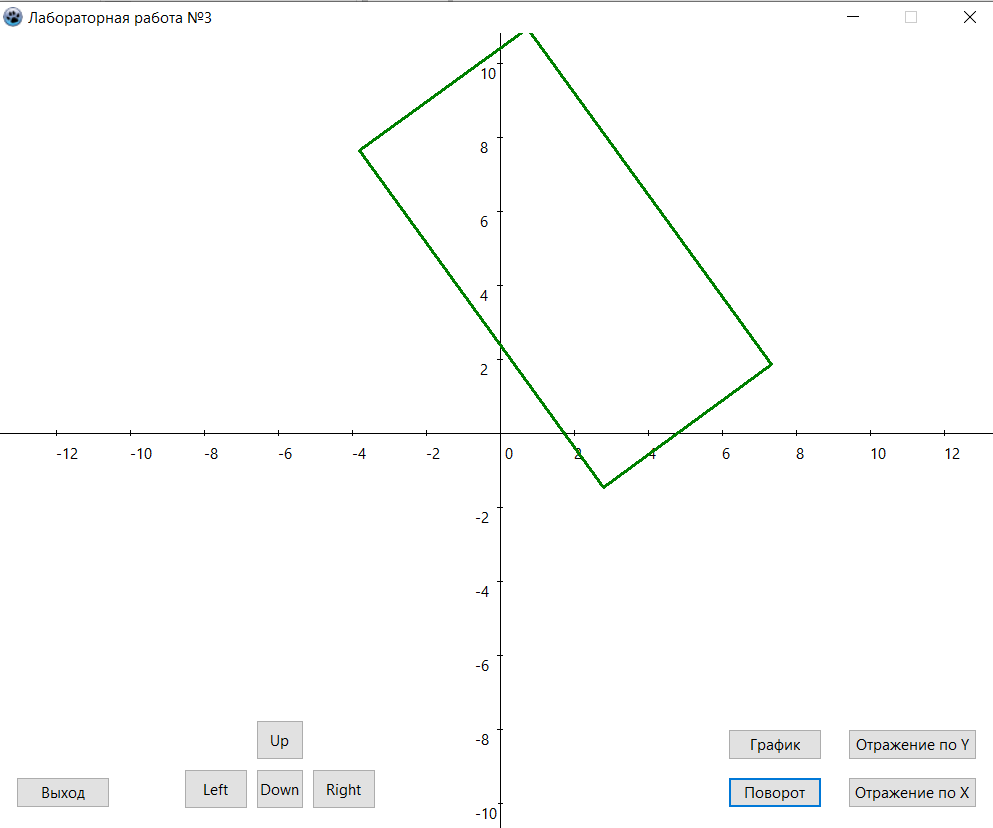
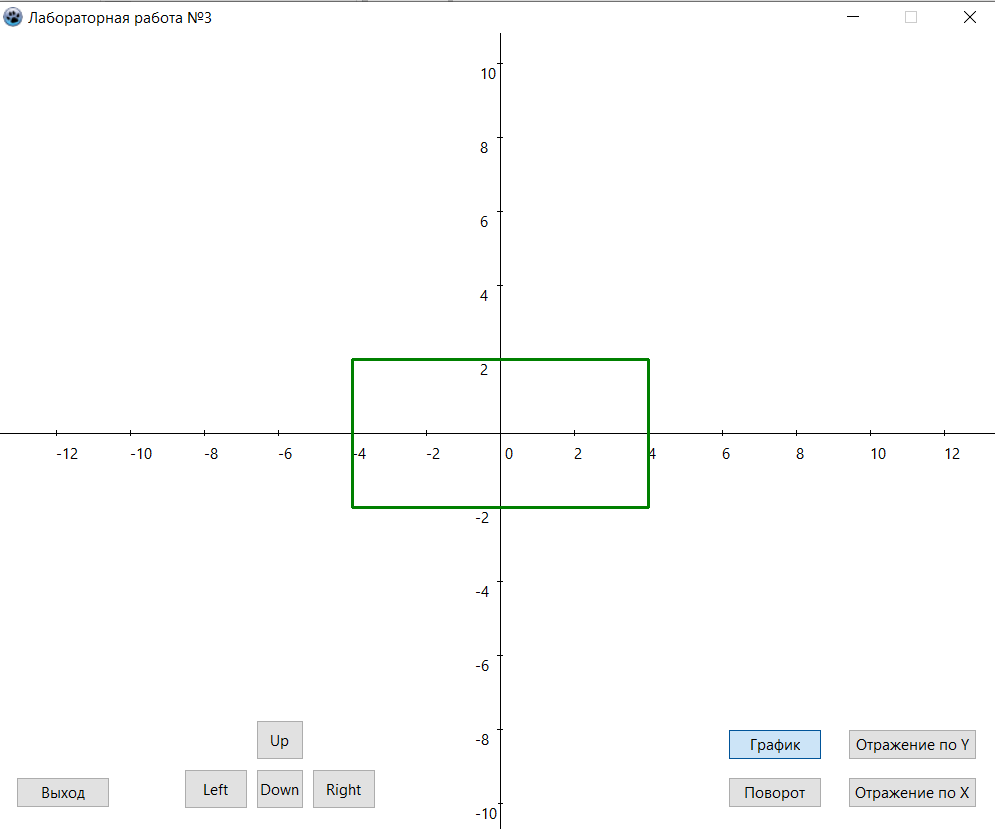
end;

end.

**Схемы алгоритмов**



**Экранные формы**



**Вывод**

В ходе выполнения данной лабораторной работы были получены навыки в реализации аффинных преобразований. При помощи среды программирования Lazarus были реализованы все виды аффинных преобразований: увеличение объекта и его уменьшение, поворот на определенное количество градусов по часовой стрелке, передвижение вправо, влево, вниз, вверх относительно графика и отражение относительно оси X и Y. Часть была реализована при помощи мыши.