МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Вятский государственный университет»**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

АФИННЫЕ ПРЕОБРАЗОВАНИЯ НА ПЛОСКОСТИ

Отчет по лабораторной работе №6

по дисциплине «Компьютерная графика»

Выполнил студент группы ИВТ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Седов М.Д./

Проверил преподаватель кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Клюкин В.Л./

Киров 2018

1 Цель работы

Закрепить лекционный материал по изучению материала 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.

2 Постановка задачи

Реализовать 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.

3 Алгоритм работы программы

Схема алгоритма умножения матрицы на вектор представлена на рисунке 1.



Рисунок 1 - Алгоритм работы программы

4 Экранная форма работы программы

Экранная форма работы программы представлена на рисунке 2.

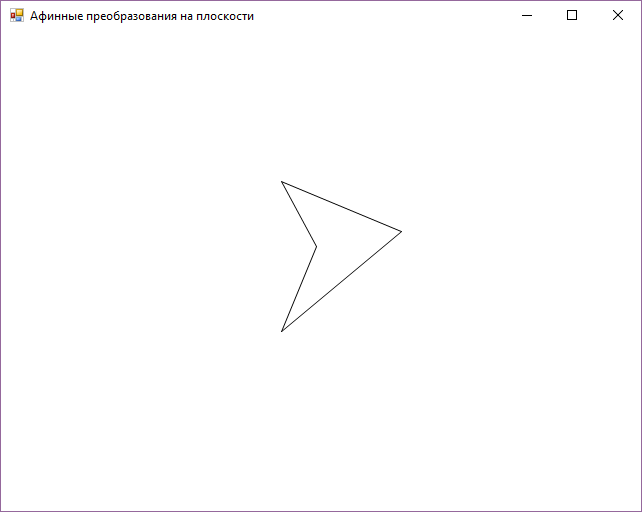


Рисунок 2 – экранная форма работы программы

5 Код программы

Код программы представлен в приложении А.

6 Вывод

В ходе лабораторной работы были реализованы 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.

Приложение А

(обязательное)

Код программы

**uses GraphABC;**

**type TXY = record x, y : integer; end;**

**const V : array [0..4] of TXY = (**

**(x : 50; y : 50),**

**(x : 100; y : 30),**

**(x : 130; y : 60),**

**(x : 70; y : 80),**

**(x : 30; y : 70)**

**);**

**dx = 200;**

**kx = 1.1;**

**ky = 1.1;**

**wx = 0.9;**

**wy = 0.9;**

**var a : array [0..4] of TXY;**

**i : integer;**

**procedure draw\_item();**

**var i : integer;**

**begin**

**for i := 0 to 4 do Line(a[i].x + dx, a[i].y + dx, a[(i + 1) mod 5].x + dx, a[(i + 1) mod 5].y + dx);**

**end;**

**procedure KeyDown(Key: integer);**

**var i : integer;**

**begin**

**case Key of**

**VK\_S : for i := 0 to 4 do begin**

**a[i].x := a[i].x + 10;**

**a[i].y := a[i].y + 10;**

**end;**

**VK\_x: for i := 0 to 4 do begin**

**a[i].x := a[i].x;**

**a[i].y := -a[i].y;**

**end;**

**VK\_y: for i := 0 to 4 do begin**

**a[i].x := -a[i].x;**

**a[i].y := a[i].y;**

**end;**

**VK\_q: for i := 0 to 4 do begin**

**a[i].x := round(a[i].x \* kx);**

**a[i].y := round(a[i].y \* ky);**

**end;**

**VK\_w: for i := 0 to 4 do begin**

**a[i].x := round(a[i].x \* wx);**

**a[i].y := round(a[i].y \* wy);**

**end;**

**VK\_e: for i := 0 to 4 do begin**

**a[i].x := round(a[i].x \* cos(-pi/18) - a[i].y \* sin(-pi/18));**

**a[i].y := round(a[i].x \* sin(-pi/18) + a[i].y \* cos(-pi/18));**

**end;**

**VK\_r: for i := 0 to 4 do begin**

**a[i].x := round(a[i].x \* cos(pi/18) - a[i].y \* sin(pi/18));**

**a[i].y := round(a[i].x \* sin(pi/18) + a[i].y \* cos(pi/18));**

**end;**

**VK\_Left: for i := 0 to 4 do begin**

**a[i].x := a[i].x - 10;**

**end;**

**VK\_Right: for i := 0 to 4 do begin**

**a[i].x := a[i].x + 10;**

**end;**

**VK\_Up: for i := 0 to 4 do begin**

**a[i].y := a[i].y - 10;**

**end;**

**VK\_Down: for i := 0 to 4 do begin**

**a[i].y := a[i].y + 10;**

**end;**

**end;**

**FillRect(0, 0, Window.Width, Window.Height);**

**draw\_item();**

**end;**

**begin**

**OnKeyDown := KeyDown;**

**SetPenColor(clRed);**

**for i := 0 to 4 do begin**

**a[i].x := V[i].x;**

**a[i].y := V[i].y;**

**end;**

**draw\_item();**

**end.**