



**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет**

**«СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

---

**Институт  
информационных технологий**

**Кафедра  
Инженерной Графики**

**Основная образовательная программа 09.03.02**

**«Информационные системы и технологии»**

**Отчет по дисциплине «Компьютерная геометрия и графика»**

**по лабораторной работе № 4**

**Студент  
группы ИДБ-21-06**

**Музафаров.К.Р**

**Преподаватель**

**к.т.н. , доц. Разумовский А.И**

**Москва, 2022 г.**

## ОГЛАВЛЕНИЕ

Введение.....	3
Задание 1 .....	4
Задание 2 .....	8
Задание 3 .....	12
Задание 4 .....	16
Выводы.....	20

## ВВЕДЕНИЕ

В этой лабораторной работе мы рассмотрим понятия:

1. Поворот точек
2. Линии
3. Объекта

Пусть необходимо повернуть точку  $P(x, y)$  вокруг начала координат  $O$  на угол  $\phi$ . Изображение новой точки обозначим через  $P'(x', y')$ . Существует четыре числа  $a, b, c, d$  такие, что новые координаты  $x'$  и  $y'$  могут быть вычислены по значениям старых координат  $x$  и  $y$

## ЗАДАНИЕ 1

Общее задание заключалось в реализации изображения движения цели.

Код программы:

```
#include <windows.h>
#include <tchar.h>
#include <math.h>

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
TCHAR WinName[] = _T("MainFrame");
int APIENTRY WinMain(HINSTANCE This, HINSTANCE Prev, LPSTR cmd,
int mode)
{
    HWND hWnd; //Дескриптор главного окна программы
    MSG msg; //Структура для хранения сообщений
    WNDCLASS wc; //Класс окна
    //Определени класса окна
    wc.hInstance = This;
    wc.lpszClassName = WinName; //Имя класса окна
    wc.lpfnWndProc = WndProc; //функция окна
    wc.style = CS_HREDRAW | CS_VREDRAW; //Стиль окна
    wc.hIcon = LoadIcon(NULL, IDI_APPLICATION); //Стандартная иконка
    wc.hCursor = LoadCursor(NULL, IDC_ARROW); //Стандартный курсор
    wc.lpszMenuName = NULL; //Нет меню
    wc.cbClsExtra = 0; //Нет дополнительных данных класса
    wc.cbWndExtra = 0; //Нет дополнительных окон класса
    //Заполнение окна белым цветом
    wc.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1); //Установка
    if (!RegisterClass(&wc))
        return 0; //Регистрация класса окна
    //Создание окна
    hWnd = CreateWindow(WinName, //Имя класса окна
```

```

_T("Каркас Windows-приложения"), //Заголовок окна
WS_OVERLAPPEDWINDOW,
CW_USEDEFAULT, //X
CW_USEDEFAULT, //Y
CW_USEDEFAULT, //Width
CW_USEDEFAULT, //Height
HWND_DESKTOP, //Дескриптор родительского окна
NULL, //Нет меню
This, //Дескриптор приложения
NULL); // Доп инфы нет
ShowWindow(hWnd, mode); //Показать окно
//Цикл обработки сообщений
while (GetMessage(&msg, NULL, 0, 0)) //Цикл получения сообщений
{
    TranslateMessage(&msg); //Функция трансляции кодов нажато
    DispatchMessage(&msg); //Посылает сообщение функции
//WndProc()
}
return 0;
}
/*Оконная функция вызываемая операционной системой
и получает сообщения из очереди для данного приложения*/
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM
wParam, LPARAM lParam)
{
    PAINTSTRUCT ps;
    HDC hdc;
    static int sx, sy;
    static HPEN hpen;
    int a, b, x_scr, y_scr; //Экранные координаты

```

```

float x[4] = { 6.0,6.0,5.9,6.1 },
y[4] = { -0.25, 0.25,0.0,0.0 };
float xMax = 6.5, yMax = 6.5;
float Kx, Ky;
//Физические координаты
switch (message) //Обработчик сообщений
{
case WM_SIZE:
    sx = LOWORD(lParam); //Ширина окна
    sy = HIWORD(lParam); //Высота окна
    break;
case WM_CREATE:
    hpen = CreatePen(PS_SOLID, 2, RGB(0, 0, 0));
    break;
case WM_PAINT:
    {
        hdc = BeginPaint(hWnd, &ps);
        int i, j;
        float pi, phi, cos_phi, sin_phi, xx, yy;
        pi = 4.0 * atan(1.0);
        phi = 6 * pi / 180;
        cos_phi = cos(phi);
        sin_phi = sin(phi);
        SelectObject(hdc, hpen);
        Kx = sx / xMax;
        Ky = sy / yMax;
        for (i = 1; i <= 14; i++) {
            for (j = 0; j <= 3; j++)
            {
                xx = x[j];

```

```

yy = y[j];
x[j] = xx * cos_phi - yy * sin_phi;
y[j] = xx * sin_phi + yy * cos_phi;
}
MoveToEx(hdc, x[0] * Kx, sy - y[0] * Ky, NULL);
for (j = 1; j <= 3; j++) LineTo(hdc, x[j] * Kx, sy - y[j] * Ky);
LineTo(hdc, x[1] * Kx, sy - y[1] * Ky);
}
EndPaint(hWnd, &ps);
break;
}
case WM_DESTROY:
DeleteObject(hpen);
PostQuitMessage(0);
break;
default:
return DefWindowProc(hWnd, message, wParam, lParam); //Возврат
}
return 0;
}

```

Результат работы программы представлен на рисунке 1:

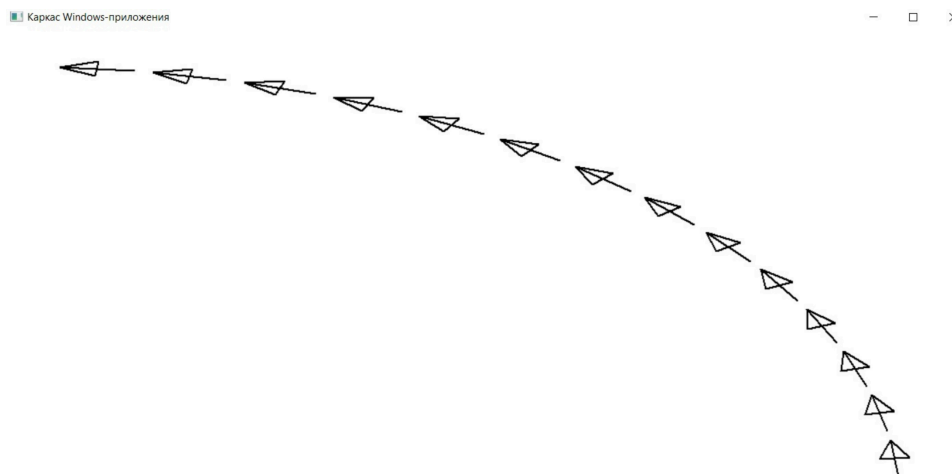


Рис.1. Движение стрелки, результат выполнения задания 1.

## ЗАДАНИЕ 2

Общее задание заключалось в изображение поворота стрелки вокруг точки.

Код программы:

```
#include <windows.h>
#include <tchar.h>
#include <math.h>

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
TCHAR WinName[] = _T("MainFrame");
int APIENTRY WinMain(HINSTANCE This, HINSTANCE Prev, LPSTR cmd,
int mode)
{
    HWND hWnd; //Дескриптор главного окна программы
    MSG msg; //Структура для хранения сообщений
    WNDCLASS wc; //Класс окна
    //Определени класса окна
    wc.hInstance = This;
    wc.lpszClassName = WinName; //Имя класса окна
    wc.lpfnWndProc = WndProc; //функция окна
    wc.style = CS_HREDRAW | CS_VREDRAW; //Стиль окна
    wc.hIcon = LoadIcon(NULL, IDI_APPLICATION); //Стандартная иконка
    wc.hCursor = LoadCursor(NULL, IDC_ARROW); //Стандартный курсор
    wc.lpszMenuName = NULL; //Нет меню
    wc.cbClsExtra = 0; //Нет дополнительных данных класса
    wc.cbWndExtra = 0; //Нет дополнительных окон класса
    //Заполнение окна белым цветом
    wc.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1); //Установка
    if (!RegisterClass(&wc))
        return 0; //Регистрация класса окна
```



```

//Создание окна
hWnd = CreateWindow(WinName, //Имя класса окна
    _T("Каркас Windows-приложения"), //Заголовок окна
    WS_OVERLAPPEDWINDOW,
    CW_USEDEFAULT, //X
    CW_USEDEFAULT, //Y
    CW_USEDEFAULT, //Width
    CW_USEDEFAULT, //Height
    HWND_DESKTOP, //Дескриптор родительского окна
    NULL, //Нет меню
    This, //Дескриптор приложения
    NULL); // Доп инфы нет
ShowWindow(hWnd, mode); //Показать окно
//Цикл обработки сообщений
while (GetMessage(&msg, NULL, 0, 0)) //Цикл получения сообщений
{
    TranslateMessage(&msg); //Функция трансляции кодов нажатой
    DispatchMessage(&msg); //Посылает сообщение функции
}
return 0;
}
/*Оконная функция вызываемая операционной системой
и получает сообщения из очереди для данного приложения*/
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM
wParam, LPARAM lParam)
{
    PAINTSTRUCT ps;
    HDC hdc;
    static int sx, sy;
    static HPEN hpen;

```

```

int a, b, x_scr, y_scr; //Экранные координаты
float x[4] = { 0.0, 0.0, -0.08, 0.08 },
y[4] = { -0.25, 0.25, 0.0, 0.0 }; //Объектные координаты стрелки
float xMax = 8.5, yMax = 8.5; //Объектные координаты окна
float Kx, Ky; //Коэффициенты масштабирования объектных координат в
float x0 = 4.5, y0 = 4.5, r = 3.0; //Координаты центра и радиус
switch (message) //Обработчик сообщений
{
case WM_SIZE:
    sx = LOWORD(lParam); //Ширина окна
    sy = HIWORD(lParam); //Высота окна
    break;
case WM_CREATE:
    hpen = CreatePen(PS_SOLID, 2, RGB(0, 0, 0));
    break;
case WM_PAINT:
    {
        hdc = BeginPaint(hWnd, &ps);
        int i, j;
        float pi, phi, cos_phi, sin_phi, dx, dy;
        pi = 4.0 * atan(1.0);
        phi = 6 * pi / 90;
        cos_phi = cos(phi);
        sin_phi = sin(phi);
        SelectObject(hdc, hpen);
        Kx = sx / xMax;
        Ky = sy / yMax;
        //Перенос в начальную позицию (x0+ r,y0)
        for (j = 0; j < 4; j++) { x[j] += x0 + r; y[j] += y0; }
        for (i = 0; i < 30; i++)

```

```

{
for (j = 0; j < 4; j++)//Цикл пересчёта координат текущей
{
dx = x[j] - x0; dy = y[j] - y0; //Сдвиг по осям в точку(x0,y0)
x[j] = x0 + dx * cos_phi - dy * sin_phi; y[j] = y0 + dx *
sin_phi + dy * cos_phi;
}
MoveToEx(hdc, x[0] * Kx, sy - y[0] * Ky, NULL); //начало
for (j = 1; j <= 3; j++) LineTo(hdc, x[j] * Kx, sy - y[j] * Ky);
LineTo(hdc, x[1] * Kx, sy - y[1] * Ky); //завершение стрелки
}
EndPaint(hWnd, &ps);
break;
}
case WM_DESTROY:
DeleteObject(hpen);
PostQuitMessage(0);
break;
default:
return DefWindowProc(hWnd, message, wParam, lParam); //Возврат
}
return 0;
}

```

Результат выполнения программы представлен на рисунке 2:

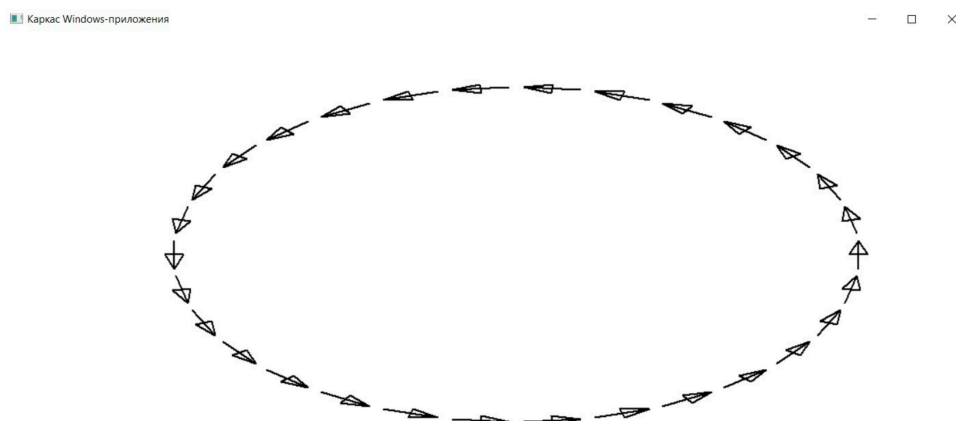


Рис.2. Движение стрелки вокруг точки, результат выполнения задания 2.

### ЗАДАНИЕ 3

Общее задание заключалось в изображении вращающегося красного квадрата.

Код программы:

```
#include <windows.h>
#include <tchar.h>
#include <math.h>

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
TCHAR WinName[] = _T("MainFrame");
//-----

int APIENTRY WinMain(HINSTANCE This, HINSTANCE Prev, LPSTR cmd,
int mode)
{
    HWND hWnd; //Дескриптор главного окна программы
    MSG msg; // Структура для хранения сообщений
    WNDCLASS wc; // Класс окна
    // Определение класса окна
    wc.hInstance = This;
    wc.lpszClassName = WinName; // Имя класса окна
    wc.lpfnWndProc = WndProc; // функция окна
    wc.style = CS_HREDRAW | CS_VREDRAW; // Стиль окна
    wc.hIcon = LoadIcon(NULL, IDI_APPLICATION); // Стандартная иконка
    wc.hCursor = LoadCursor(NULL, IDC_ARROW); // Стандартный курсор
    wc.lpszMenuName = NULL; // Нет меню
    wc.cbClsExtra = 0; // Нет дополнительных данных класса
```

```

wc.cbWndExtra = 0; //Нет дополнительных данных окна
// Заполнение окна белым цветом
wc.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1); //установка цвета
if (!RegisterClass(&wc)) return 0; // Резистрация класса окна
// Создание окна
hWnd = CreateWindow(WinName, //Имя класса окна
    _T("Каркас Windows-приложения"), // Заголовок окна
    WS_OVERLAPPEDWINDOW, // Стилъ окна
    CW_USEDEFAULT, //X
    CW_USEDEFAULT, //Y
    CW_USEDEFAULT, // Width
    CW_USEDEFAULT, // Height
    HWND_DESKTOP, // Дескриптор родительского окна
    NULL, // Нет меню
    This, // Дескриптор приложения
    NULL); // Дополнительной информации нет
ShowWindow(hWnd, mode); // Показать окно
// Цикл обработки сообщений
while (GetMessage(&msg, NULL, 0, 0)) //цикл получения сообщений
{
    TranslateMessage(&msg); // Функция трансляции кодов нажатой клавиши
    DispatchMessage(&msg); // Посылает сообщение функции WndProco
}
return 0; // при положительном завершении програимы в главизо функцию
//возвращается «0»
}
//-----
//Оконная функция вызываемая операционной системой
// и получает сообщения из очереди для данного приложения
//-----

```

```

LRESULT CALLBACK WndProc(HWND hWnd, UINT message,
WPARAM wParam, LPARAM lParam)
{
    PAINTSTRUCT ps;
    HDC hdc; int y;
    POINT pt[4] = { { -100, 100}, { -100, -100 }, { 100, -100}, { 100, 100} };
    POINT pt1[4];
    const int WIDTH = 400;
    const int HEIGHT = 300;
    int n = 8, i, j;
    double alpha = 3.14 / n;
    int sx, sy;
    static HBRUSH hBrush;
    int a, b, x_scr, y_scr; //Экранные координаты
    double x;
    static HPEN hpen;
    //Обработчик сообщений
    switch (message)
    {
    case WM_SIZE:
        sx = LOWORD(lParam); //Ширина окна
        sy = HIWORD(lParam); //Высота окна
        break;
    case WM_PAINT:
        {
            hdc = BeginPaint(hWnd, &ps);
            for (i = 0; i <= n; i++)
            {
                hBrush = CreateSolidBrush(RGB(255, 255 - 255. / n * i, 255 - 255. / n *
                i));
            }
        }
    }
}

```

```

// Создание изменения цвета кисти
SelectObject(hdc, hBrush);
for (j = 0; j <= 3; j++)
{
    pt1[j].x = 500 + (pt[j].x * cos(alpha * i) - pt[j].y * sin(alpha * i)) + i * 10;
    pt1[j].y = 300 + (pt[j].x * sin(alpha * i) + pt[j].y * cos(alpha * i));
}
Polygon(hdc, pt1, 4);
}
EndPaint(hWnd, &ps);
break;
}
case WM_DESTROY: PostQuitMessage(0);
break; // Завершение программы
// Обработка сообщения по умолчанию
default: return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
}

```

Результат выполнения программы представлен на рисунке 3:

Каркас Windows-приложения

— □ ×

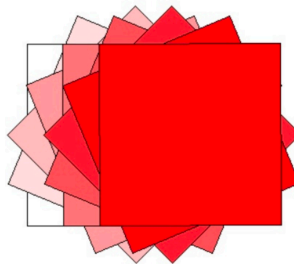


Рис.3. Поверхность, результат выполнения задания 3.

## ЗАДАНИЕ 4

Индивидуальное задание, с помощью кода повернуть любой объект в соответствии с номером варианта с лабораторной номер 3. Вариант 15: задание трапеция с квадратом внутри.

Код программы:

```
#include <windows.h>
#include <tchar.h>
#include <math.h>
LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
TCHAR WinName[] = _T("MainFrame");
//-----
int APIENTRY WinMain(HINSTANCE This, HINSTANCE Prev, LPSTR cmd,
    int mode)
{
    HWND hWnd; //Дескриптор главного окна программы
    MSG msg; // Структура для хранения сообщений
    WNDCLASS wc; // Класс окна
    // Определение класса окна
    wc.hInstance = This;
    wc.lpszClassName = WinName; // Имя класса окна
    wc.lpfnWndProc = WndProc; // функция окна
    wc.style = CS_HREDRAW | CS_VREDRAW; // Стилль окна
    wc.hIcon = LoadIcon(NULL, IDI_APPLICATION); // Стандартная иконка
    wc.hCursor = LoadCursor(NULL, IDC_ARROW); // Стандартный курсор
    wc.lpszMenuName = NULL; // Нет меню
    wc.cbClsExtra = 0; // Нет доломительных данных класса
    wc.cbWndExtra = 0; //Нет дополнительных данных окна
    // Заполнение окна белым цветом
    wc.hbrBackground = (HBRUSH)(COLOR_WINDOW + 1); //установка цвета
```



```

if (!RegisterClass(&wc)) return 0; // Резистрация класса окна
// Создание окна
hWnd = CreateWindow(WinName, //Имя класса окна
    _T("Каркас Windows-приложения"), // Заголовок окна
    WS_OVERLAPPEDWINDOW, // Стилъ окна
    CW_USEDEFAULT, //X
    CW_USEDEFAULT, //Y
    CW_USEDEFAULT, // Width
    CW_USEDEFAULT, // Height
    HWND_DESKTOP, // Дескриптор родительского окна
    NULL, // Нет меню
    This, // Дескриптор приложения
    NULL); // Дополнительной информации нет
ShowWindow(hWnd, mode); // Показать окно
// Цикл обработки сообщений
while (GetMessage(&msg, NULL, 0, 0)) //цикл получения сообщений
{
    TranslateMessage(&msg); // Функция трансляции кодов нажатой клавиши
    DispatchMessage(&msg); // Посылает сообщение функции WndProc
}
return 0; // при положительном завершении программы в главную функцию
//возвращается «0»
}
//-----
//Оконная функция вызываемая операционной системой
// и получает сообщения из очереди для данного приложения
//-----
LRESULT CALLBACK WndProc(HWND hWnd, UINT message,
    WPARAM wParam, LPARAM lParam)
{

```

```

PAINTSTRUCT ps;
HDC hdc; int y;
POINT rectangleForm[4] = { { -95, -51}, { -59, 51 }, { 59, 51}, { 95, -51} };
POINT trapezoidForm[4] = { { -30, 30}, { -30, -30 }, { 30, -30}, { 30, 30} };
POINT rectangleFigure[4];
POINT trapezoidFigure[4];
const int WIDTH = 400;
const int HEIGHT = 300;
int n = 8, i, j;
double alpha = 3.14 / n;
int sx, sy;
static HBRUSH hBrush;
int a, b, x_scr, y_scr; //Экранные координаты
double x;
static HPEN hpen;
//Обработчик сообщений
switch (message)
{
case WM_SIZE:
    sx = LOWORD(lParam); //Ширина окна
    sy = HIWORD(lParam); //Высота окна
    break;
case WM_PAINT:
    {
        hdc = BeginPaint(hWnd, &ps);
        for (i = 0; i <= n; i++)
        {
            hBrush = CreateSolidBrush(RGB(166 + 89 / n*i, 0 , 166 + 89 / n * i));
            // Создание изменения цвета кисти
            SelectObject(hdc, hBrush);

```

```

    for (j = 0; j <= 3; j++)
    {
        rectangleFigure[j].x = 500 + (rectangleForm[j].x * cos(alpha * i) -
rectangleForm[j].y * sin(alpha * i)) + i * 10;
        rectangleFigure[j].y = 300 + (rectangleForm[j].x * sin(alpha * i) +
rectangleForm[j].y * cos(alpha * i));
        trapezoidFigure[j].x = 500 + (trapezoidForm[j].x * cos(alpha * i) -
trapezoidForm[j].y * sin(alpha * i)) + i * 10;
        trapezoidFigure[j].y = 300 + (trapezoidForm[j].x * sin(alpha * i) +
trapezoidForm[j].y * cos(alpha * i));
    }
    BeginPath(hdc);
    Polyline(hdc, rectangleFigure, 4);
    Polyline(hdc, trapezoidFigure, 4);
    CloseFigure(hdc);
    EndPath(hdc);
    SelectObject(hdc, hBrush);
    FillPath(hdc);
}
EndPaint(hWnd, &ps);

break;
}
case WM_DESTROY: PostQuitMessage(0);
    break; // Завершение программы
    // Обработка сообщения по умолчанию
default: return DefWindowProc(hWnd, message, wParam, lParam);
}
return 0;
}

```

Результат выполнения программы представлен на рисунке 4:

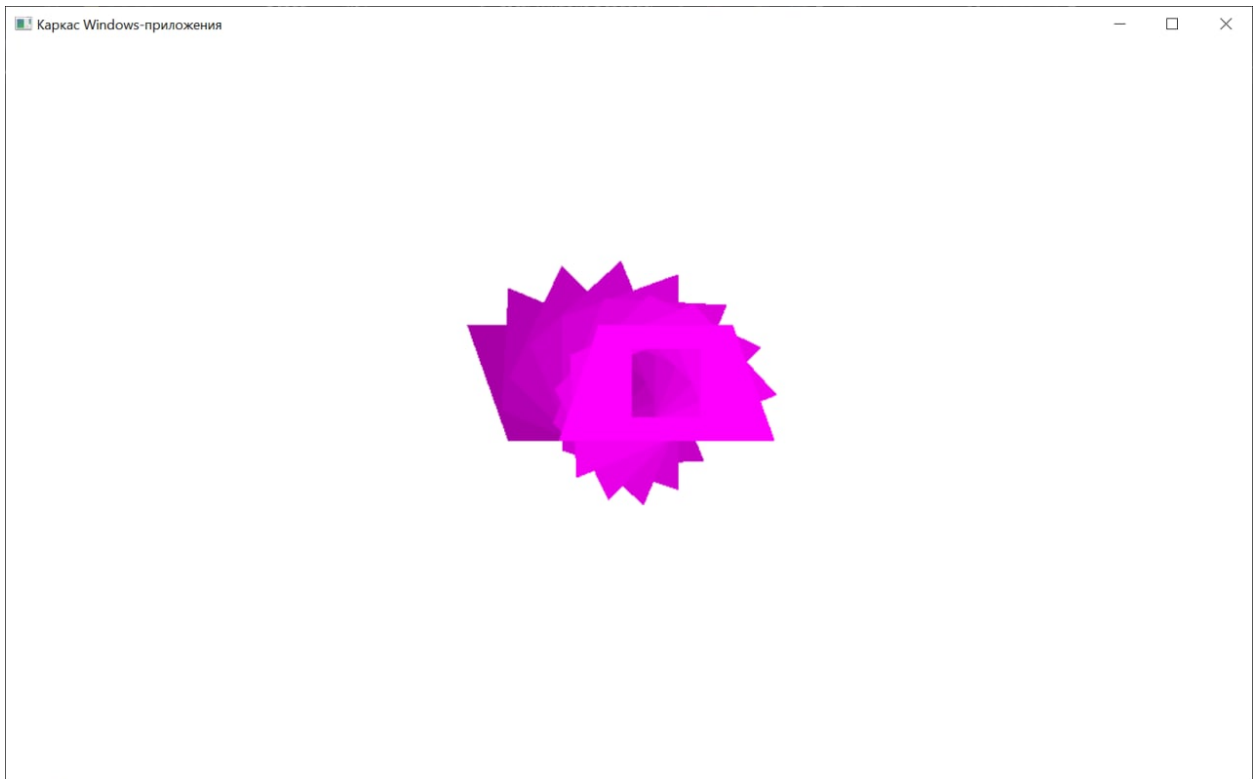


Рис.4. Вращение трапеции с квадратом внутри, результат выполнения задания 4.

## ВЫВОДЫ

В этой лабораторной работе были созданы 4 полноценные конструкции с реализацией движения графических объектов по средствам математических функций. В каждой программе используются различные функции, позволяющие графически изобразить задачу. Освоены новые навыки работы с функциями, отрисовкой объектов и циклами.