| *voenmeh* | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| --- | --- |
| БГТУ.СМК-Ф-4.2-К5-01 |

| Факультет |  | О |  | Естественнонаучный |
| --- | --- | --- | --- | --- |
|  |  | шифр |  | наименование |
| Кафедра |  | О7 |  | Информационные системы и программная инженерия |
|  |  | шифр |  | наименование |
| Дисциплина |  | Компьютерная геометрия и графика | | |

Отчёт по практической работе № 3

| Использование массивов вершин. |
| --- |
| Преобразования координат |
|  |

| Выполнил студент группы | | | |  | И508Б |
| --- | --- | --- | --- | --- | --- |
| Столяров А. К. | | | | | |
| Фамилия И.О. | | | | | |
| **РУКОВОДИТЕЛЬ** | | | | | |
|  | |  |  | | |
| Фамилия И.О. Подпись | | | | | |
| Оценка |  | | | |  |
| «\_\_\_\_\_» |  | | | | 2022г. |

САНКТ-ПЕТЕРБУРГ

2022 г

**Вариант 3**

**Задания к работе**

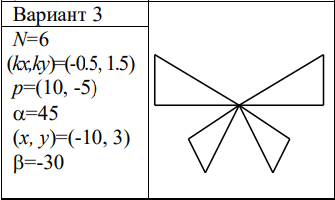


Рисунок 1 - Задание по варианту

1. Создать массив вершин, содержащий координаты правильного N-угольника. Используя сформированный массив, нарисовать:

А) полигон:

Содержимое функции DrawGLScene:

glClear(GL\_COLOR\_BUFFER\_BIT); // очистка экрана

glEnableClientState(GL\_VERTEX\_ARRAY);

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);

srand(time(NULL));

GLdouble vertexArray[6][2] = { };

double radius = 0.9;

glLineWidth(N);

glColor3f((float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX);

glVertexPointer(2, GL\_DOUBLE, 0, vertexArray);

for (int i = 0; i < (int) N; i++)

{

vertexArray[i][0] = radius \* cos(2.0\*M\_PI\*i/N);

vertexArray[i][1] = radius \* sin(2.0\*M\_PI\*i/N);

}

glDrawArrays(GL\_POLYGON, 0, (int) N);

glDisableClientState(GL\_VERTEX\_ARRAY);

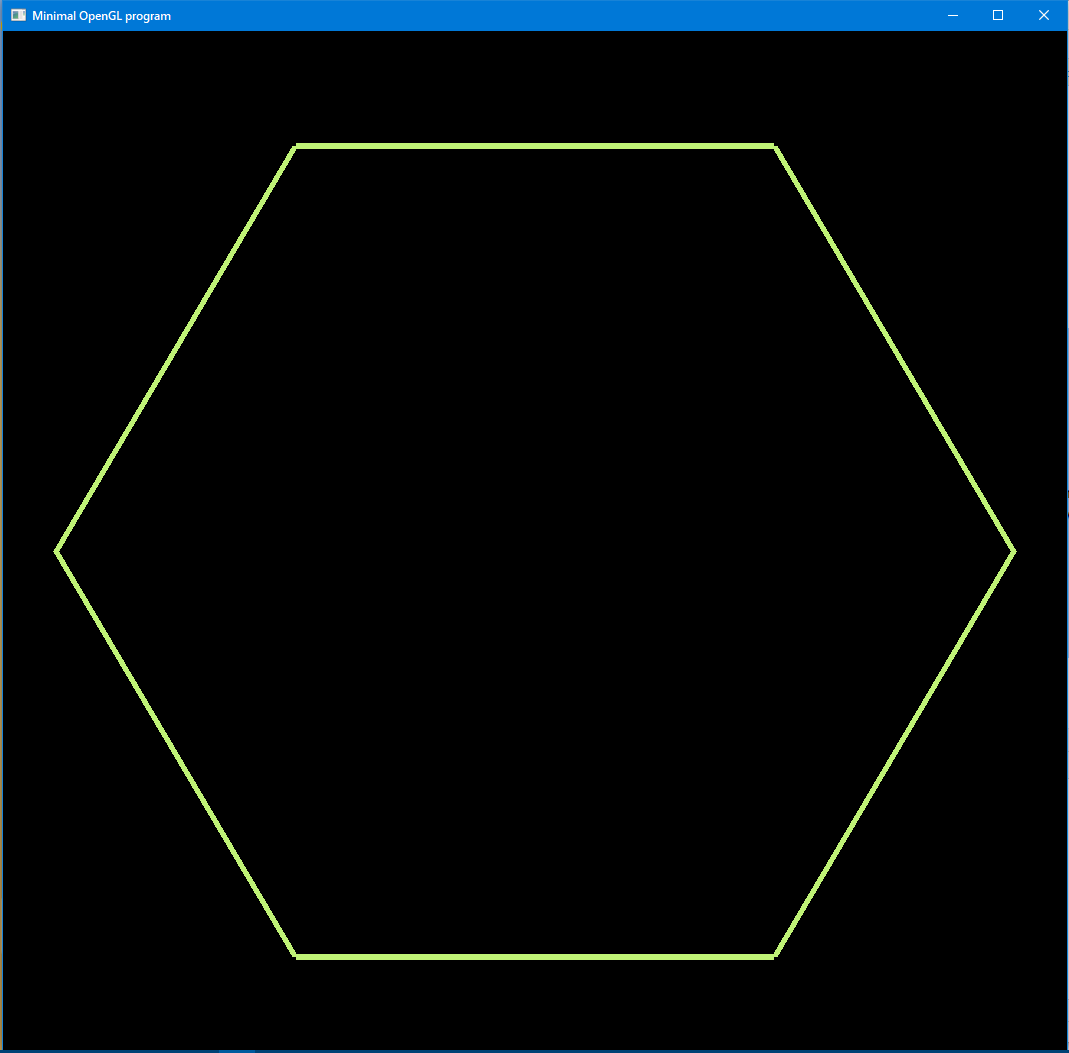


Рисунок 2 - Полигон

Б) набор четырехугольников:

Содержимое функции DrawGLScene:

glDrawArrays(GL\_QUADS, 0, (int) N);

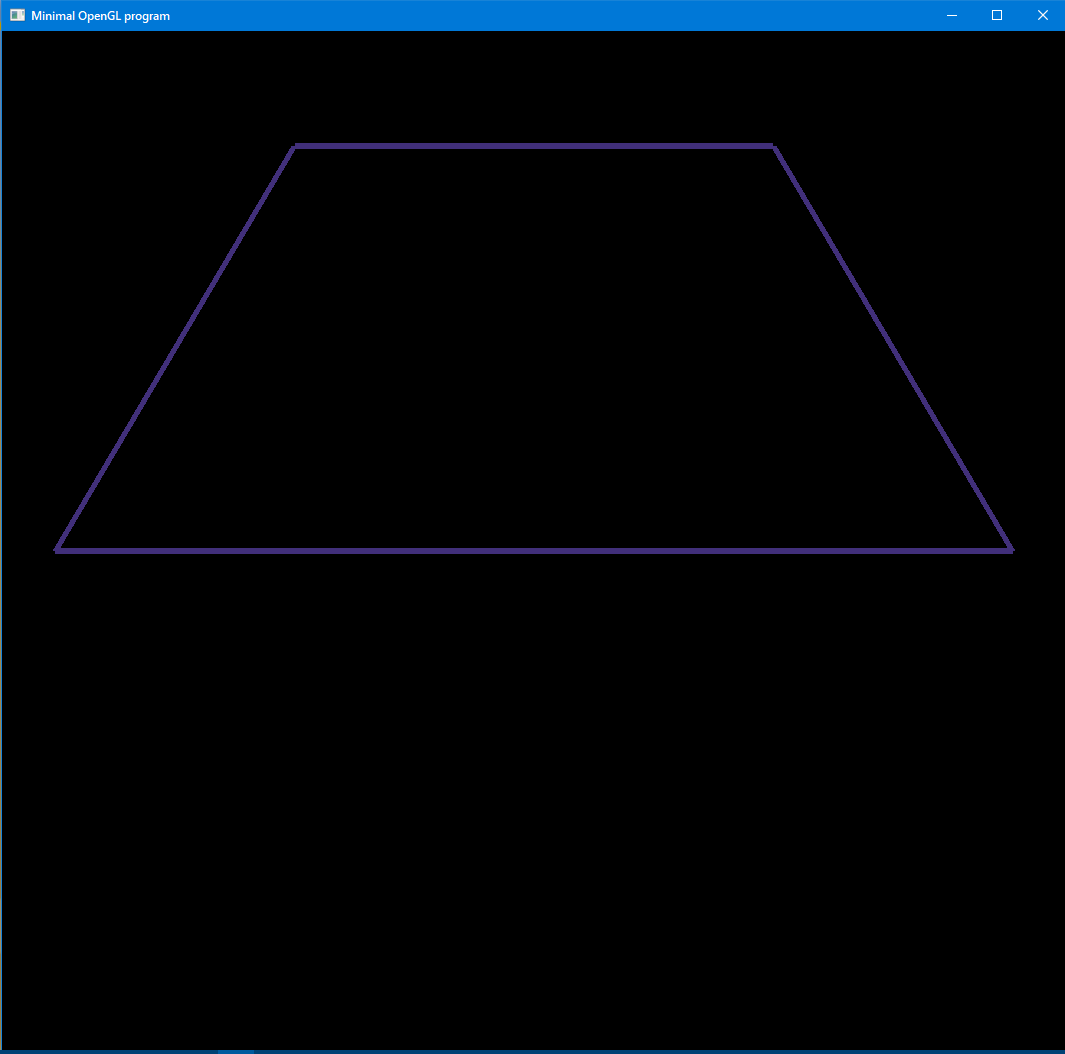


Рисунок 3 - Набор четырехугольников

В) набор треугольников:

Содержимое функции DrawGLScene:

glDrawArrays(GL\_TRIANGLES, 0, (int) N);

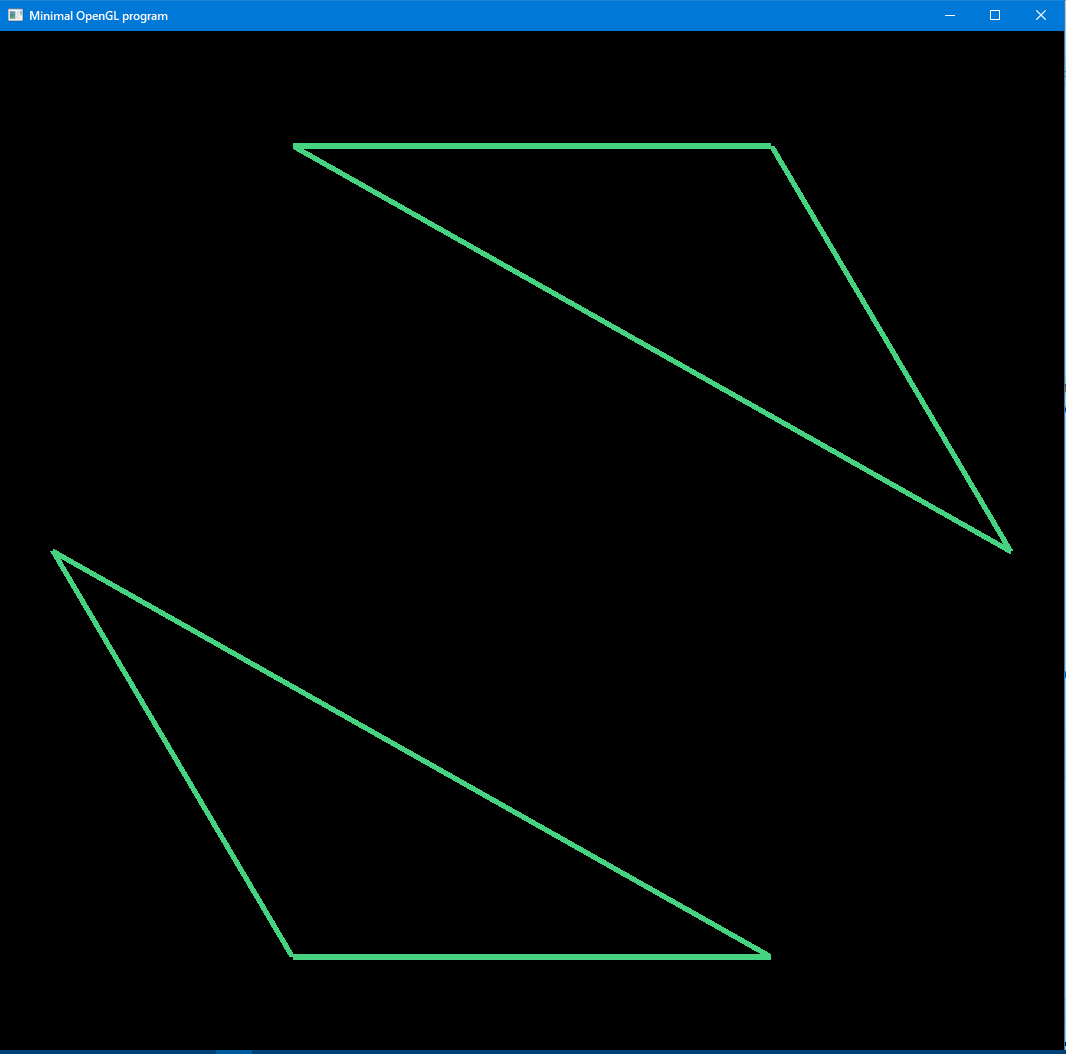


Рисунок 4 - Набор треугольников

Г) набор линий:

Содержимое функции DrawGLScene:

glDrawArrays(GL\_LINES, 0, (int) N);

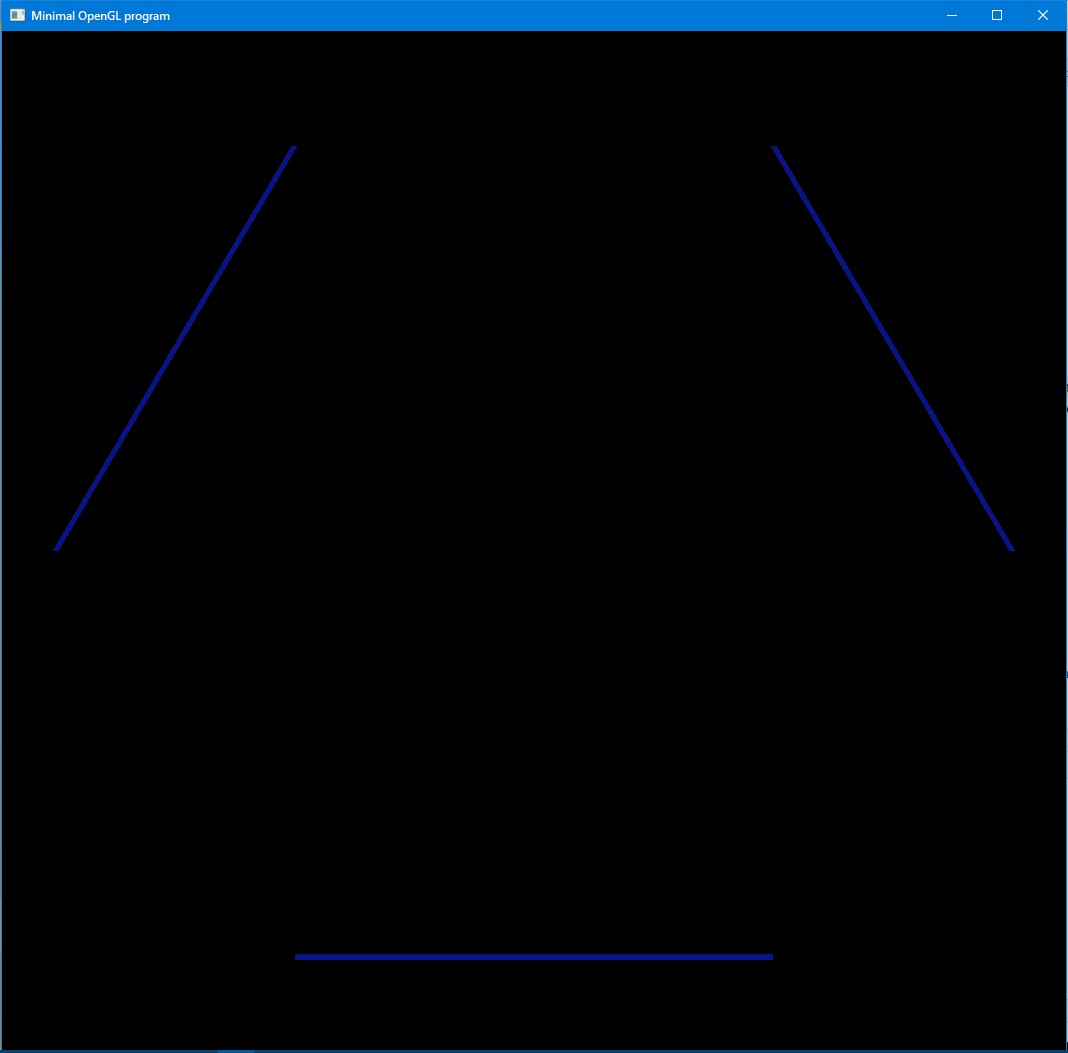


Рисунок 5 - Набор линий

1. Нарисовать треугольник, прямоугольник и линию

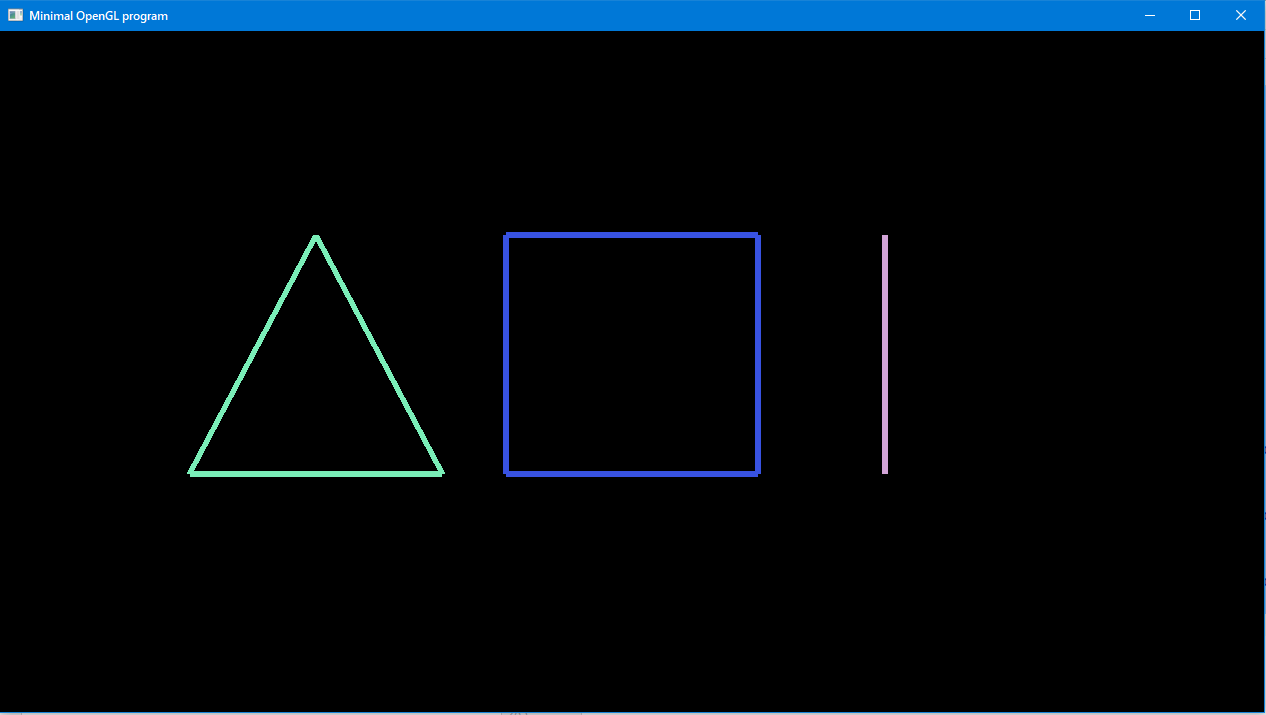


Рисунок 6 - Треугольник, прямоугольник и линия

Содержимое функции DrawGLScene:

glClear(GL\_COLOR\_BUFFER\_BIT); // очистка экрана

glEnableClientState(GL\_VERTEX\_ARRAY);

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);

srand(time(NULL));

GLdouble vertexArray[9][2] = {

{-0.7, -0.3},

{-0.5, 0.4},

{-0.3, -0.3},

{-0.2, -0.3},

{0.2, -0.3},

{0.2, 0.4},

{-0.2, 0.4},

{0.4, -0.3},

{0.4, 0.4},

};

glLineWidth(6);

glVertexPointer(2, GL\_DOUBLE, 0, vertexArray);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

// Треугольник

glScaled(-0.5, 1.5, 0);

glTranslated(1, -0.5, 0);

glColor3f((float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX);

glDrawArrays(GL\_TRIANGLES, 0, 3);

// Прямоугольник

glLoadIdentity();

glRotated(-30, -10, 3, 0);

glColor3f((float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX);

glDrawArrays(GL\_POLYGON, 3, 4);

// Линия

glRotated(30, -10, 3, 0); // вернули в исходное положение

glRotated(45, 0, 0, 1);

glColor3f((float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX);

glDrawArrays(GL\_LINES, 7, 2);

glDisableClientState(GL\_VERTEX\_ARRAY);

Получить новое изображение, на котором:

* треугольник следует масштабировать с коэффициентами (kx, ky) и перенести на вектор p;
* линию повернуть на угол α относительно начала координат;
* прямоугольник повернуть относительно точки с координатами (x, y) на угол β.

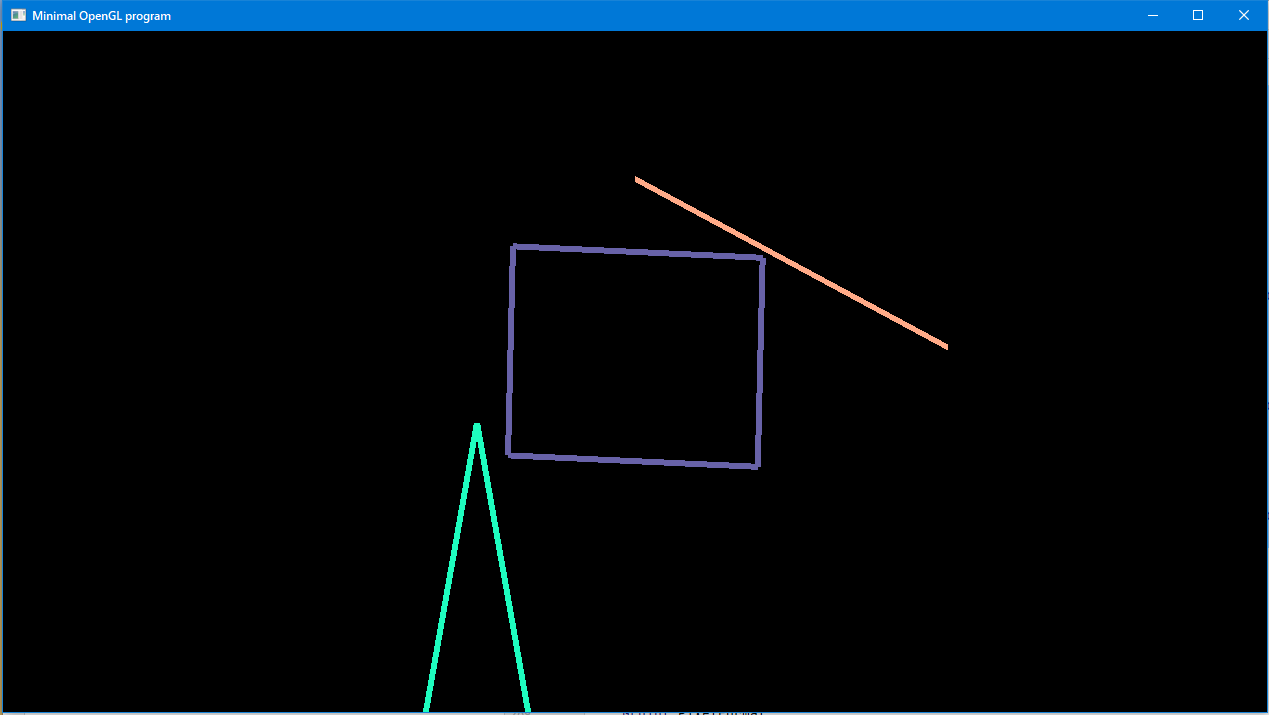


Рисунок 7 - Треугольник, прямоугольник и линия после преобразований

Содержимое функции DrawGLScene:

glClear(GL\_COLOR\_BUFFER\_BIT); // очистка экрана

glEnableClientState(GL\_VERTEX\_ARRAY);

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);

srand(time(NULL));

GLdouble vertexArray[9][2] = {

{-0.7, -0.3},

{-0.5, 0.4},

{-0.3, -0.3},

{-0.2, -0.3},

{0.2, -0.3},

{0.2, 0.4},

{-0.2, 0.4},

{0.4, -0.3},

{0.4, 0.4},

};

glLineWidth(6);

glVertexPointer(2, GL\_DOUBLE, 0, vertexArray);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

// Треугольник

glScaled(-0.5, 1.5, 1);

glTranslated(1, -0.5, 1);

glColor3f((float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX);

glDrawArrays(GL\_TRIANGLES, 0, 3);

// Прямоугольник

glLoadIdentity();

glRotated(-30, -10, 3, 1);

glColor3f((float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX);

glDrawArrays(GL\_POLYGON, 3, 4);

// Линия

glRotated(30, -10, 3, 1); // вернули в исходное положение

glRotated(45, 0, 0, 1);

glColor3f((float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX);

glDrawArrays(GL\_LINES, 7, 2);

glDisableClientState(GL\_VERTEX\_ARRAY);

1. Используя примитив для рисования линий и операции геометрических преобразований, изобразить фигуру, соответствующую варианту задания.

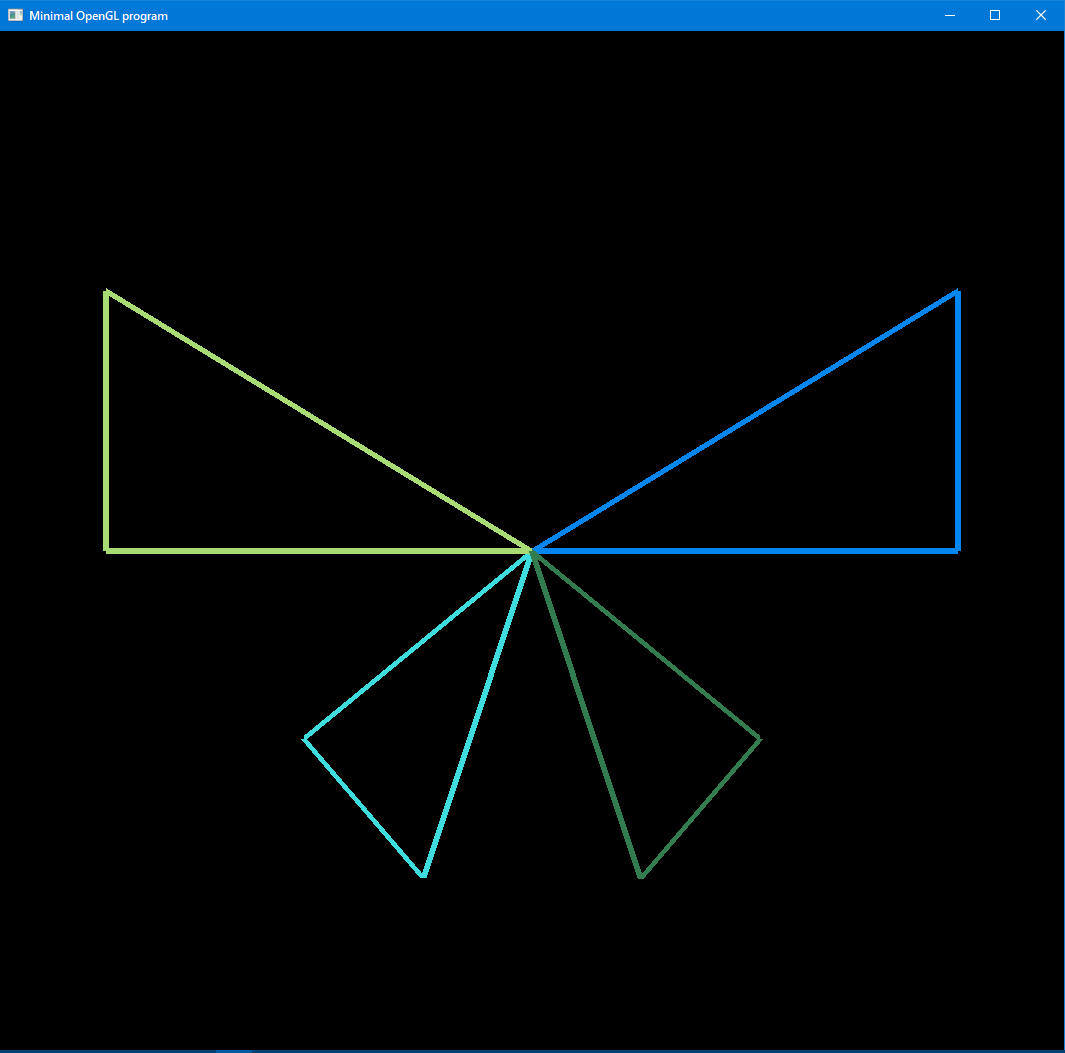


Рисунок 8 – Фигура по варианту

Содержимое функции DrawGLScene:

glClear(GL\_COLOR\_BUFFER\_BIT); // очистка экрана

glEnableClientState(GL\_VERTEX\_ARRAY);

glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE);

srand(time(NULL));

GLdouble vertexArray[6][2] = {

{0,0},

{0.8,0},

{0.8,0.5},

{0,0},

{-0.8,0},

{-0.8,0.5},

};

glLineWidth(6);

glVertexPointer(2, GL\_DOUBLE, 0, vertexArray);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity(); // делаем матрицу единичной

// Верхний правый треугольник

glColor3f((float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX);

glDrawArrays(GL\_LINE\_LOOP, 0, 3);

// Верхний левый треугольник

glColor3f((float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX);

glRotated(-140, 0, 0, 1);

glScaled(0.7, 0.7, 0);

glDrawArrays(GL\_LINE\_LOOP, 0, 3);

// Правый нижний треугольник

glColor3f((float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX);

glRotated(140, 0, 0, 1);

glScaled(1.43, 1.43, 0);

glDrawArrays(GL\_LINE\_LOOP, 3, 3);

//Левый нижний треугольник

glColor3f((float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX, (float)rand() / RAND\_MAX);

glRotated(140, 0, 0, 1);

glScaled(0.7, 0.7, 0);

glDrawArrays(GL\_LINE\_LOOP, 3, 3);

glDisableClientState(GL\_VERTEX\_ARRAY);