```cpp
// Polygon_PS1.cpp

#include "Polygon.h"
#include <cmath>

float Polygon::getSignedArea() const noexcept {
    float fResult = 0.0f;

    if (fNumberOfVertices > 2) {
        for (size_t aIndex = 0; aIndex < fNumberOfVertices; ++aIndex) {
            size_t aNextIndex = (aIndex + 1) % fNumberOfVertices;
            float fHeight = fVertices[aNextIndex].y() - fVertices[aIndex].y();
            float fTrapezoidArea = (fVertices[aIndex].x() + fVertices[aNextIndex].x()) * fHeight;
            fResult += fTrapezoidArea;
        }
        return fResult / 2.0f;
    }
}

Polygon Polygon::transform(const Matrix3x3& transformationMatrix) const noexcept {
    Polygon transformedPolygon;
    for (size_t aIndex = 0; aIndex < fNumberOfVertices; ++aIndex) {
        Vector3D transformedVertex = transformationMatrix * Vector3D(fVertices[aIndex].x(), fVertices[aIndex].y(), 1.0f);
        transformedPolygon.fVertices[aIndex] = Vector2D(transformedVertex.x(), transformedVertex.y());
    }
    transformedPolygon.fNumberOfVertices = fNumberOfVertices;
    return transformedPolygon;
}

```