МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №2**

**з дисципліни «Комп’ютерна графіка»**

**на тему: «Системи координат та афінні перетворення»**

Виконав: студент гр. ПЗ1911

Сіньков Г.О.

Прийняла: ас. каф. КІТ

Нежуміра О. І.

Дніпро, 2020

**Лабораторна робота №2**

**Тема.** Системи координат та афінні перетворення.

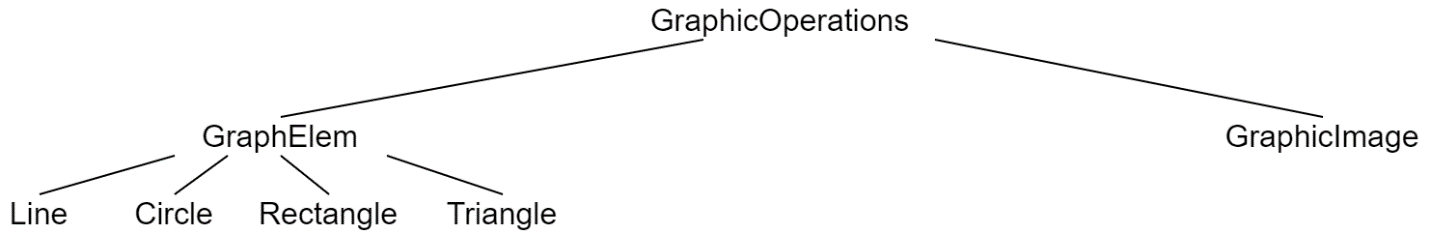
**Мета.** Отримати практичні навички у створенні графічних об’єктів і реалізації та використання афінних перетворень.

**Постановка задачі**

**Загальне завдання**

Розробити об’єктно-орієнтовану програму для використання операцій перетворення над графічними об’єктами на площині

**Структура графічного об’єкта**



**Текст програми**

GraphicOperation.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Drawing;

using System.Drawing.Drawing2D;

namespace Lr2

{

public interface GraphicOperations

{

void draw(Graphics gr); // рисование елемента

void move(int delX, int delY); // двигаться в стороны

void rotate(int angle, int px, int py); // вращение

void scale(int px, int py, double w, double h); // масштаб

}

class GraphElem : GraphicOperations

{

protected List<Point> points = new List<Point>(); // масив точек

protected String name; // имя елемента

public override String ToString()

{

return name;

}

public void draw(Graphics gr)

{

gr.DrawLines(Pens.Black, points.ToArray());

}

public void move(int delX, int delY)

{

for (int i = 0; i < points.Count; i++)

points[i] = new Point(points[i].X + delX, points[i].Y + delY);

}

public void rotate(int angle, int px, int py)

{

double rad = angle \* Math.PI / 180;

for (int i = 0; i < points.Count; i++)

{

int x = (int)((points[i].X - px) \* Math.Cos(rad) - (points[i].Y - py) \* Math.Sin(rad) + px);

int y = (int)((points[i].X - px) \* Math.Sin(rad) + (points[i].Y - py) \* Math.Cos(rad) + py);

//int x = (int)(points[i].X \* Math.Cos(rad) - points[i].Y \* Math.Sin(rad) - px \* Math.Cos(rad) + py \* Math.Sin(rad) + px);

//int y = (int)(points[i].X \* Math.Sin(rad) + points[i].Y \* Math.Cos(rad) - px \* Math.Sin(rad) - py \* Math.Cos(rad) + py);

points[i] = new Point(x, y);

}

}

public void scale(int px, int py, double w, double h)

{

for (int i = 0; i < points.Count; i++)

{

int x = (int)(points[i].X \* w - w \* px + px);

int y = (int)(points[i].Y \* h - h \* py + py);

points[i] = new Point(x, y);

}

}

}

class Line : GraphElem

{

public Line(String aname, Point p1, Point p2)

{

points.Add(p1);

points.Add(p2);

name = aname;

}

}

class Circle : GraphElem

{

public Circle(String aname, Point cntr, int rd)

{

name = aname;

for(int alpha = 0; alpha <= 360; alpha += 6)

{

int x = (int)(cntr.X + rd \* Math.Sin(alpha \* Math.PI / 100));

int y = (int)(cntr.Y + rd \* Math.Cos(alpha \* Math.PI / 100));

points.Add(new Point(x, y));

}

}

}

class Rectangle : GraphElem

{

public Rectangle(String aname, Point cntr, int width, int height)

{

points.Add(cntr);

points.Add(new Point(cntr.X + width, cntr.Y));

points.Add(new Point(cntr.X + width, cntr.Y + height));

points.Add(new Point(cntr.X, cntr.Y + height));

points.Add(cntr);

name = aname;

}

}

class Triangle : GraphElem

{

public Triangle(String aname, Point lf, Point up, Point rh)

{

points.Add(lf);

points.Add(up);

points.Add(rh);

points.Add(lf);

name = aname;

}

}

class GraphicImage : GraphicOperations

{

private List<GraphElem> items = new List<GraphElem>();

private List<bool> SelectedCheck = new List<bool>();

public List<String> getNames()

{

List<String> lst = new List<String>();

foreach (var x in items)

lst.Add(x.ToString());

return lst;

}

public void draw(Graphics gr)

{

foreach (var x in items)

x.draw(gr);

}

public void move(int delX, int delY)

{

for (int x = 0; x < items.Count; x++)

{

if(SelectedCheck[x])

items[x].move(delX, delY);

}

}

public void rotate(int angle, int px, int py)

{

for (int x = 0; x < items.Count; x++)

{

if (SelectedCheck[x])

items[x].rotate(angle, px, py);

}

}

public void scale(int px, int py, double w, double h)

{

for (int x = 0; x < items.Count; x++)

{

if (SelectedCheck[x])

items[x].scale(px, py, w, h);

}

}

public void AddLine(String name, Point p1, Point p2)

{

items.Add(new Line(name, p1, p2));

SelectedCheck.Add(false);

}

public void AddCircle(String name, Point p1, int r)

{

items.Add(new Circle(name, p1, r));

SelectedCheck.Add(false);

}

public void AddRectangle(String name, Point cntr, int width, int height)

{

items.Add(new Rectangle(name, cntr, width, height));

SelectedCheck.Add(false);

}

public void AddTriangle(String name, Point lf, Point up, Point rh)

{

items.Add(new Triangle(name, lf, up, rh));

SelectedCheck.Add(false);

}

public void SetCheck(int index, bool val)

{

SelectedCheck[index] = val;

}

}

}

Form1.cs

using System;

using System.Drawing;

using System.Windows.Forms;

namespace Lr2

{

public partial class Form1 : Form

{

private GraphicImage img = new GraphicImage();

private Point center = new Point(0, 0);

public Form1()

{

InitializeComponent();

img.AddTriangle("Roof", new Point(150, 100), new Point(233, 50), new Point(315, 100));

img.AddRectangle("1th column", new Point(150, 100), 15, 100);

img.AddRectangle("2th column", new Point(180, 100), 15, 100);

img.AddRectangle("3th column", new Point(210, 100), 15, 100);

img.AddRectangle("4th column", new Point(240, 100), 15, 100);

img.AddRectangle("5th column", new Point(270, 100), 15, 100);

img.AddRectangle("6th column", new Point(300, 100), 15, 100);

img.AddRectangle("Second stairs", new Point(140, 200), 190, 15);

img.AddRectangle("First stairs", new Point(120, 215), 230, 15);

foreach (var n in img.getNames()) // добаляем имя елемента в checkedListBox1

checkedListBox1.Items.Add(n);

}

private void pictureBox1\_Paint(object sender, PaintEventArgs e)

{

img.draw(e.Graphics); // рисуем нашу фигуру

e.Graphics.DrawEllipse(Pens.Red, center.X, center.Y, 4, 4); // рисуем выбраную точку

}

private void buttonRotate\_Click(object sender, EventArgs e)

{

img.rotate(10, center.X, center.Y);

pictureBox1.Refresh();

}

private void buttonRotateAgainst\_Click(object sender, EventArgs e)

{

img.rotate(-10, center.X, center.Y);

pictureBox1.Refresh();

}

private void buttonScale\_Click(object sender, EventArgs e)

{

double kX = 1, kY = 1;

if(checkBoxX.Checked)

kX = 1.5;

if (checkBoxY.Checked)

kY = 1.5;

img.scale(center.X, center.Y, kX, kY);

pictureBox1.Refresh();

}

private void checkedListBox1\_ItemCheck(object sender, ItemCheckEventArgs e)

{

img.SetCheck(e.Index, e.NewValue == CheckState.Checked);

}

private void pictureBox1\_MouseClick(object sender, MouseEventArgs e)

{

center.X = e.X;

center.Y = e.Y;

textBoxPx.Text = (e.X).ToString();

textBoxPy.Text = (e.Y).ToString();

pictureBox1.Refresh();

}

private void button1\_Click(object sender, EventArgs e) // buttonScale-

{

double kX = 1, kY = 1;

if (checkBoxX.Checked)

kX = 0.5;

if (checkBoxY.Checked)

kY = 0.5;

img.scale(center.X, center.Y, kX, kY);

pictureBox1.Refresh();

}

private void buttonMoveUP\_Click(object sender, EventArgs e)

{

img.move(0, -10); // двигаем вверх на 10

pictureBox1.Refresh();

}

private void buttonMoveLeft\_Click(object sender, EventArgs e)

{

img.move(-10, 0); // двигаем влево на 10

pictureBox1.Refresh();

}

private void buttonMoveRight\_Click(object sender, EventArgs e)

{

img.move(10, 0); // двигаем вправо на 10

pictureBox1.Refresh();

}

private void buttonMoveDown\_Click(object sender, EventArgs e)

{

img.move(0, 10); // двигаем вниз на 10

pictureBox1.Refresh();

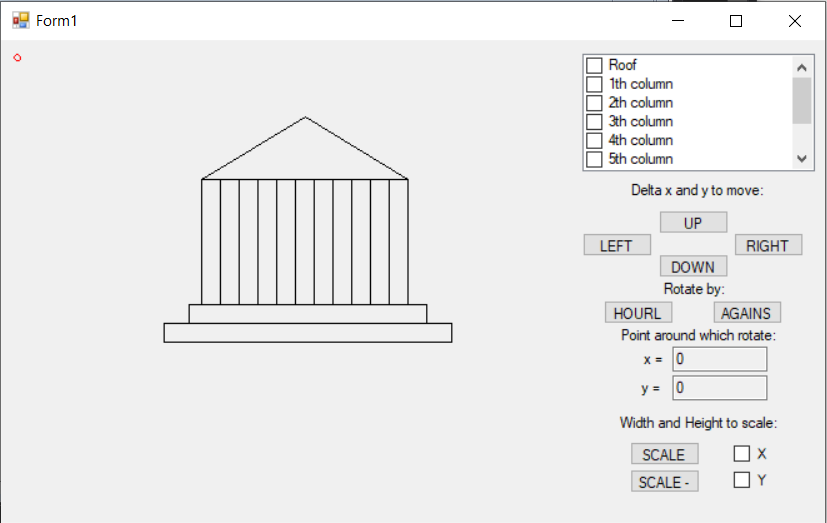
}

}

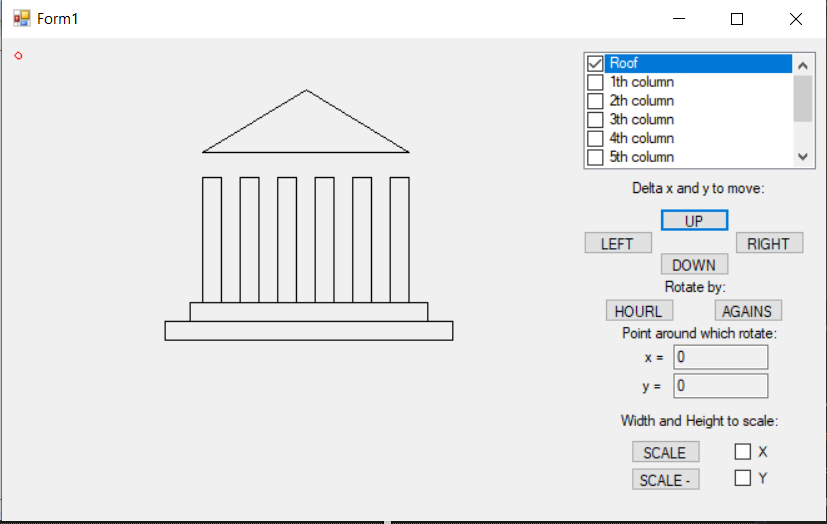
}

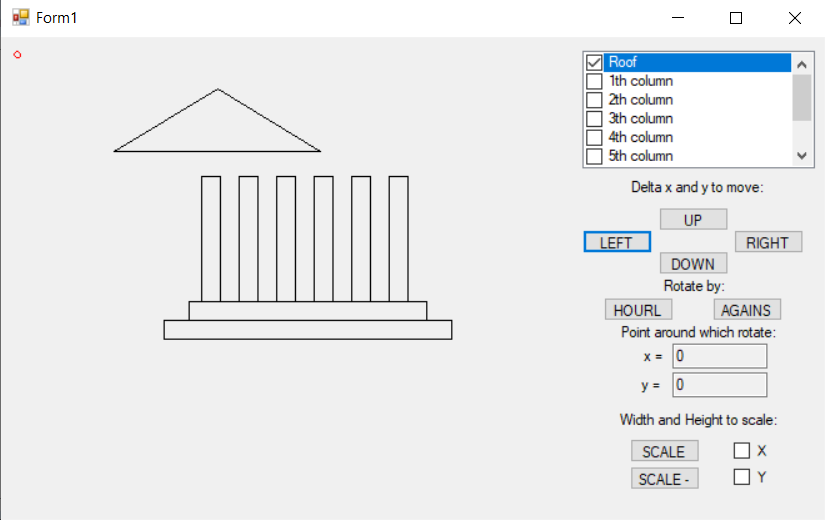
**Демонстрація роботи**

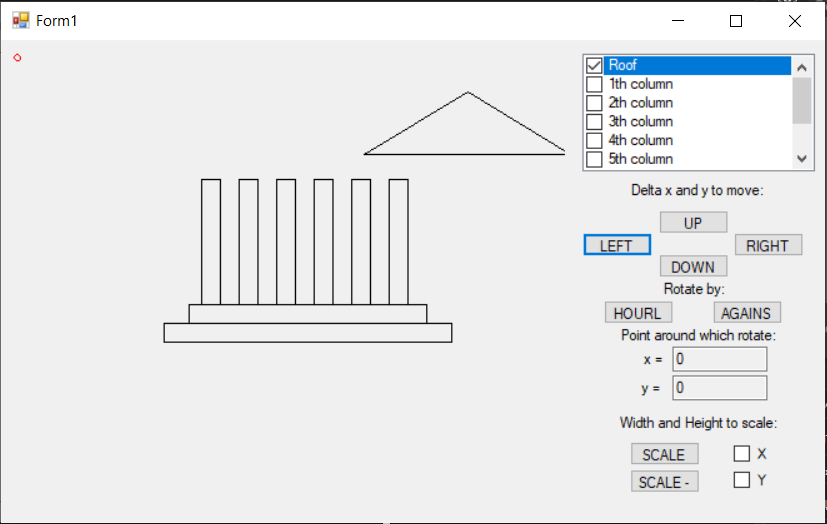
1.При запуску програми.

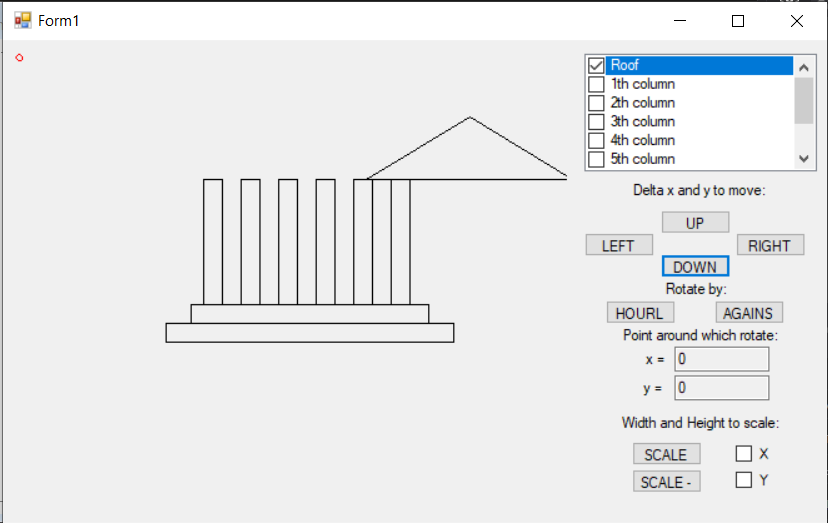


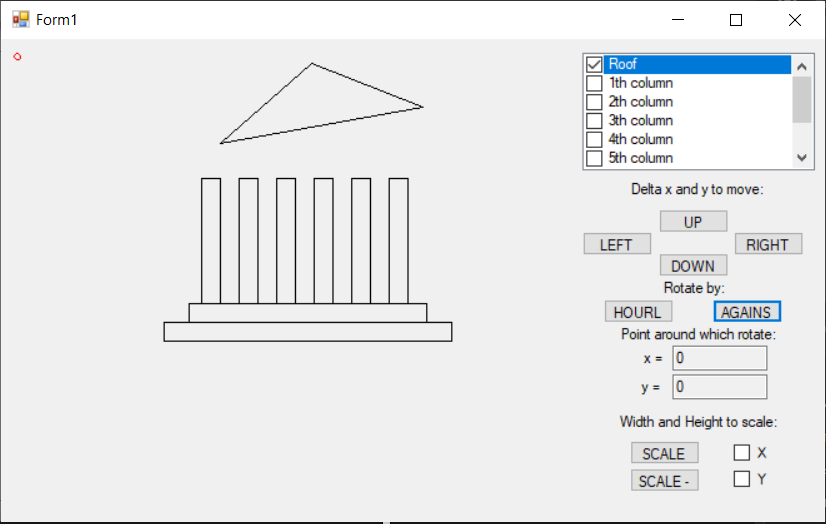
2.Перевірка усіх перетворень, відносно точки (0, 0).

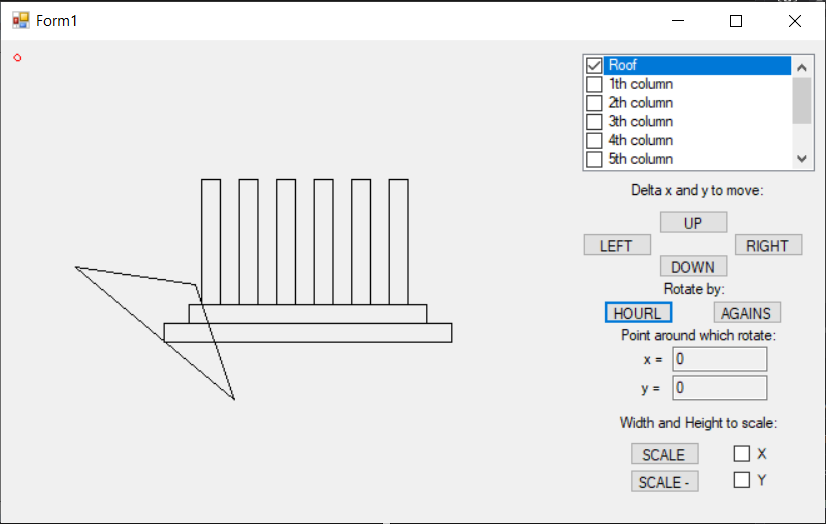


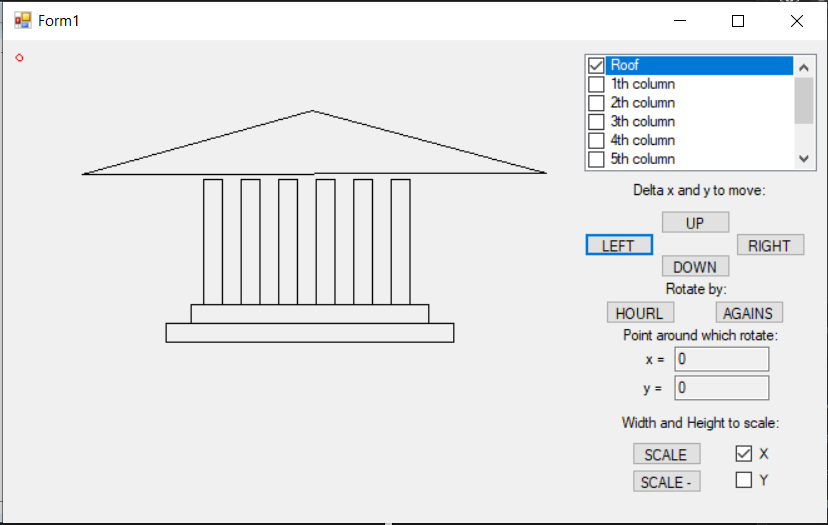


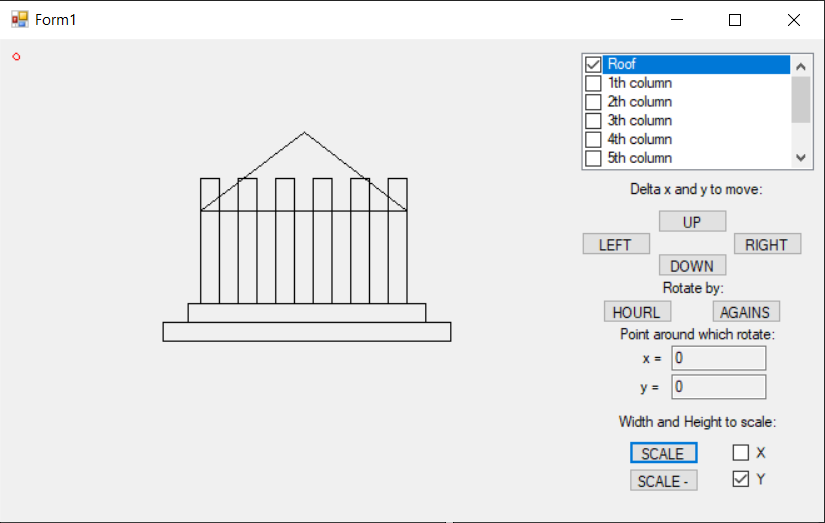


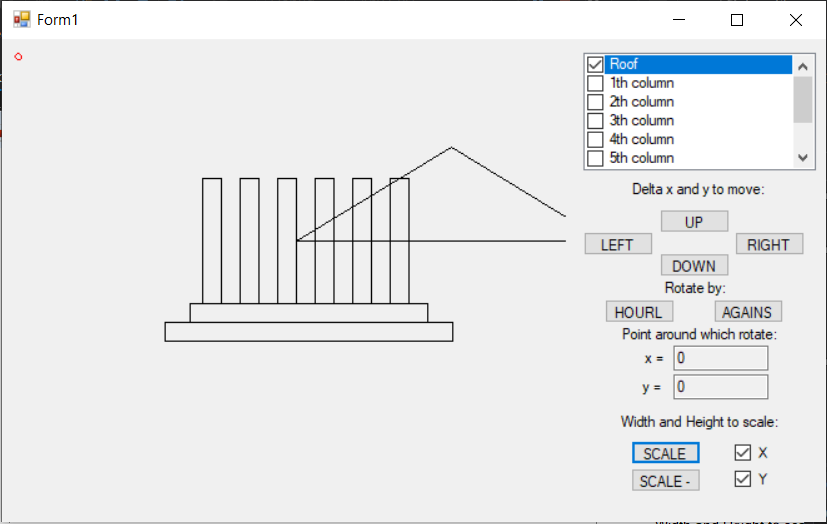


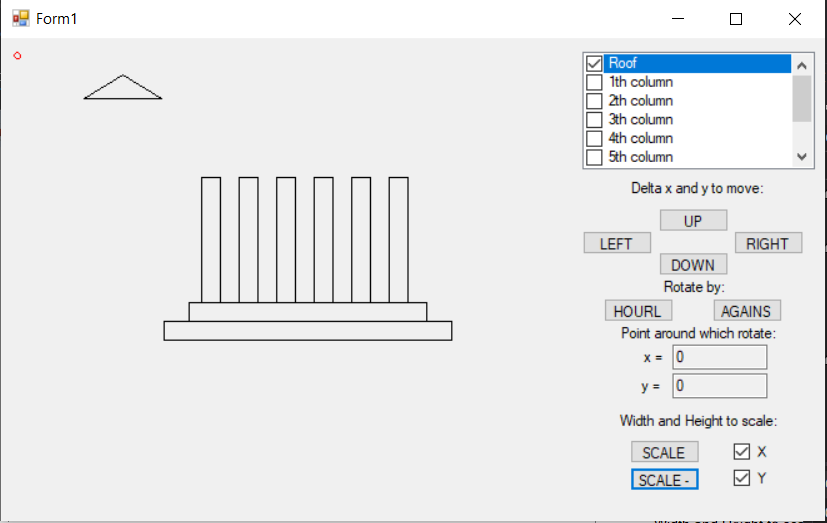




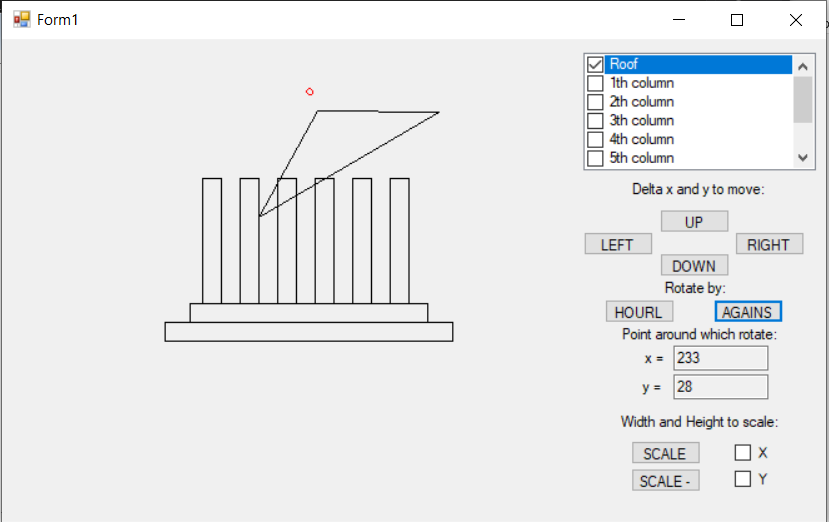


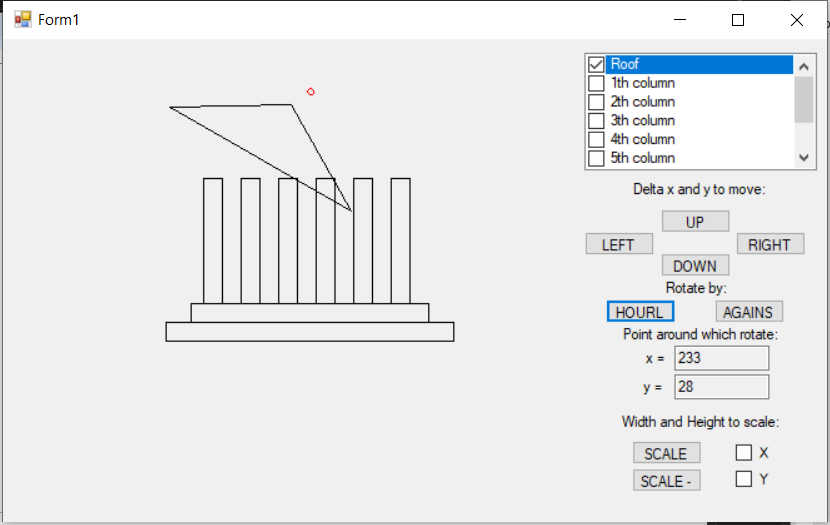


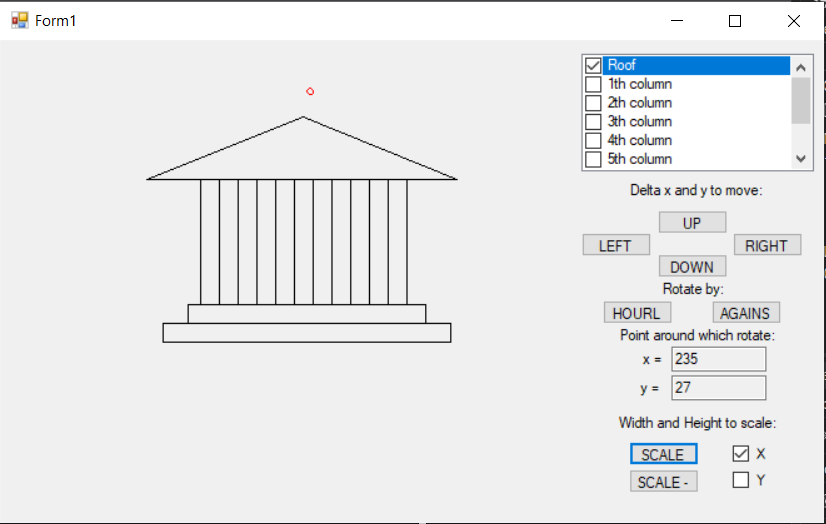


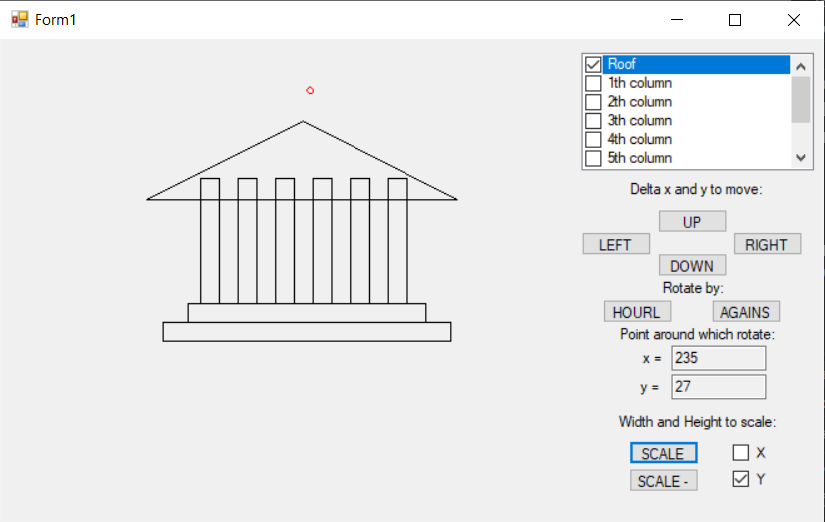


3.Зміна точки відносно якої буде відбуватися перетворення.









**Висновок**: Отже, в ході лабораторної роботи, ми розробляли об’єктно-орієнтовну програму для виконання операцій перетворення над графічними об'єктами на площині. Спершу ми створили загальний клас з набором усіх перетворень на площині. Унаслідували 2 класи: клас виводу графічних елементів та клас графічних елементів відповідно. В клас графічних елементів наслідували графічні об’єкти необхідні для малювання бажаної картинки (пряма, коло, прямокутник, трикутник). Графічні об’єкти будували по точках. Переміщення, нахил, масштабування – усе було реалізовано і працює при зміні прапору з елементами в checkListBox у різних напрямах, у різну сторону прокрутки, у різних осях та відносно будь-якої точки.