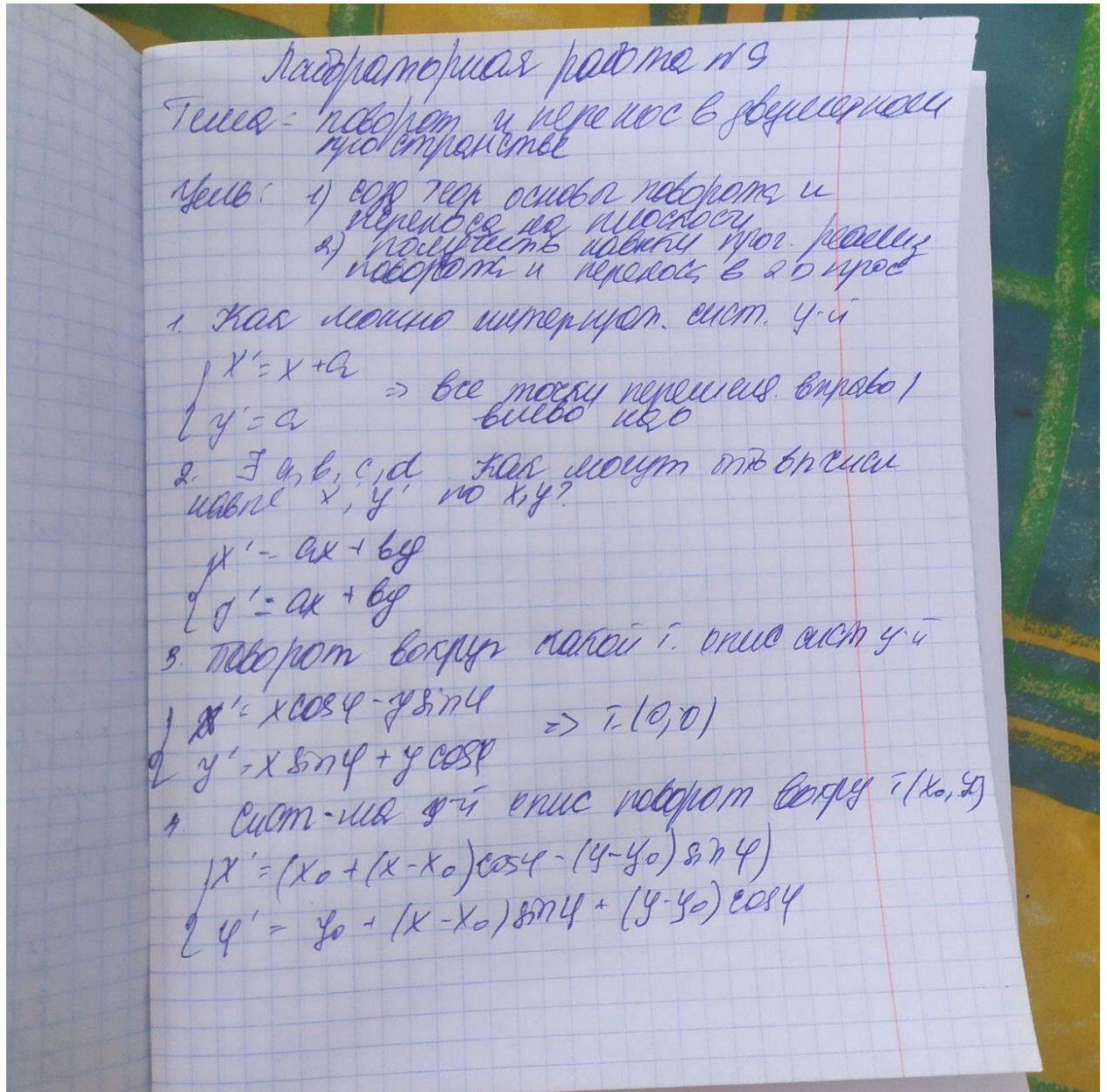


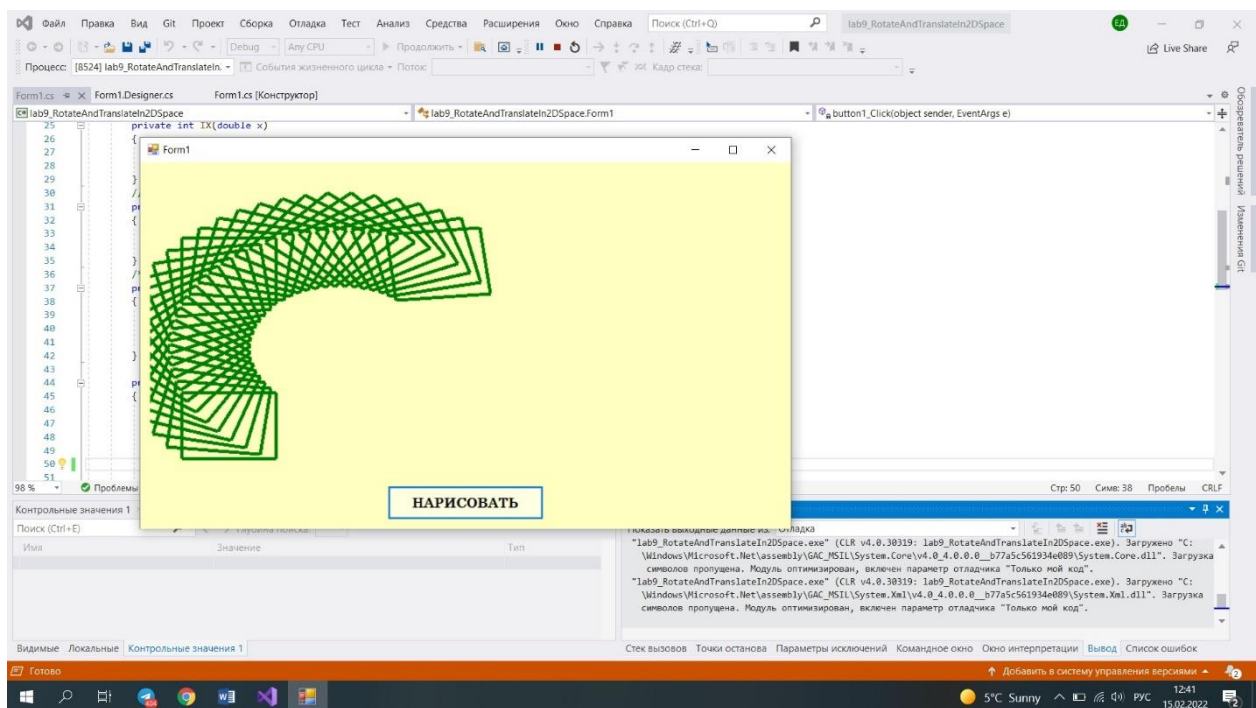
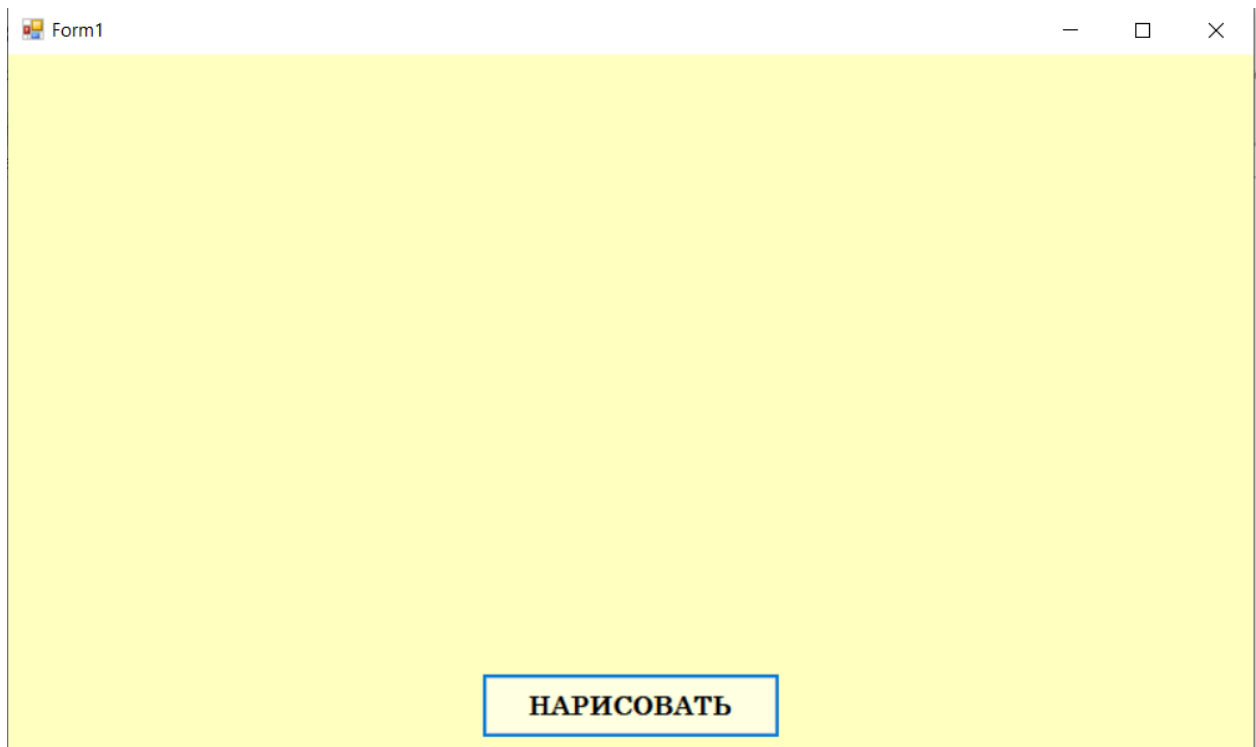
5. Напишите программу для вычерчивания 30 прямоугольников.

Вершины первого прямоугольника заданы в точках (6, 1), (6, 2.5), (8.5, 2.5),

(8.5, 1). Каждый следующий прямоугольник получается путем поворота

предыдущего против часовой стрелки на угол 60° вокруг точки (5, 1).





```
namespace lab9_RotateAndTranslateIn2DSpace
{
    public partial class Form1 : Form
    {
        Graphics dc;
        Pen p;
        public Form1()
        {
            InitializeComponent();
            dc = pictureBox1.CreateGraphics();
            p = new Pen(Brushes.Green, 4);
        }

        // Метод преобразования вещественной координаты X в целую */
        private int IX(double x)
    }
}
```

```

    {
        double xx = x * (pictureBox1.Size.Width / 10.0) + 0.5;
        return (int)xx;
    }
    // Метод преобразования вещественной координаты Y в целую */
    private int IY(double y)
    {
        double yy = pictureBox1.Size.Height - y * (pictureBox1.Size.Height / 7.0) +
0.5;
        return (int)yy;
    }
    /* Своя функция вычерчивания линии (экран 10x7 условных единиц) */
    private void Draw(double x1, double y1, double x2, double y2)
    {
        Point point1 = new Point(IY(x1), IY(y1));
        Point point2 = new Point(IY(x2), IY(y2));
        dc.DrawLine(p, point1, point2);
    }

    private void button1_Click(object sender, EventArgs e)
    {
        double[] x; x = new double[4] { 1.0, 1.0, 2.5, 2.5 };
        double[] y; y = new double[4] { 1.0, 2.5, 2.5, 1.0 };
        int i, j;
        double Phi, cos_Phi, sin_Phi, dx, dy;
        double x0 = 5.0, y0 = 1.0, xold = 0.0, yold = 0.0;
        Phi = 6 * Math.PI / 180;
        cos_Phi = Math.Cos(Phi);
        sin_Phi = Math.Sin(Phi);

        //смещение относительно центра вращения
        for (j = 0; j < 4; j++) { x[j] += x0; y[j] += y0; }
        //цикл прорисовки прямоугольников
        for (i = 0; i < 30; i++)
        {
            //прорисовка текущего прямоугольника
            for (j = 0; j <= 3; j++)
            {
                //пересчет координат для текущего прямоугольника

                dx = x[j] - x0;
                dy = y[j] - y0;
                x[j] = x0 + dx * cos_Phi - dy * sin_Phi;
                y[j] = y0 + dx * sin_Phi + dy * cos_Phi;
            }
            // прорисовка прямоугольника
            xold = x[3]; yold = y[3];
            for (j = 0; j <= 3; j++)
            {
                Draw(xold, yold, x[j], y[j]);
                xold = x[j]; yold = y[j];
            }
        }
    }
}

```