

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”
Факультет прикладної математики
Кафедра системного програмування і спеціалізованих комп’ютерних систем

Лабораторна робота № 1

з дисципліни:

«Інженерна та комп’ютерна графіка-2. Комп’ютерна графіка»

Виконав: Стецюренко І. С,

Студент групи КВ-03

Перевірив(ла): _____

Постановка задачі

Програмно реалізувати алгоритми растрівання відрізків та кола, що використовують псевдопиксель розміром $N \times N$.

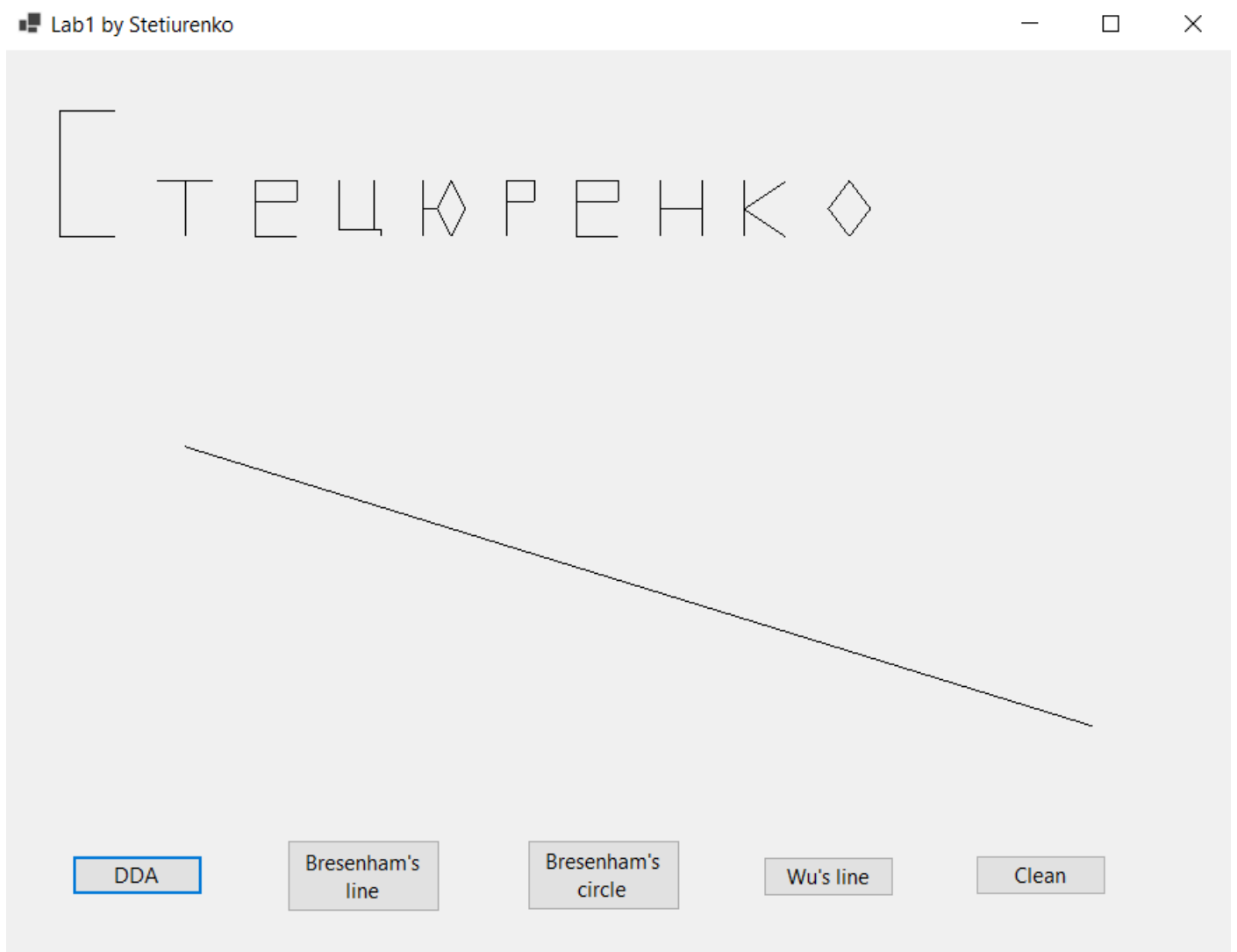
Алгоритми для растрівання:

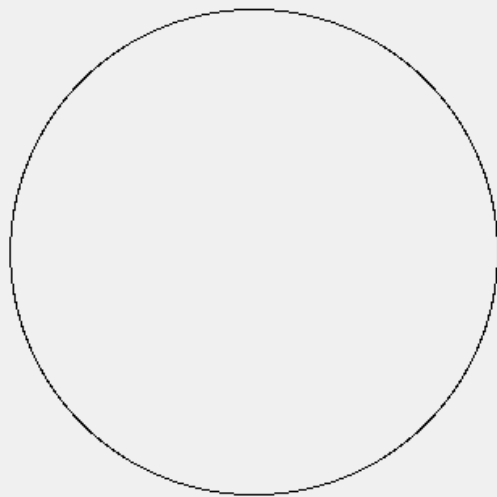
- *алгоритм ЦДА* (цифровий диференціальний аналізатор, *DDA*) для растрівання відрізка;
- *алгоритми Брезенхема* для растрівання відрізка та кола;
- *алгоритм Ву* для растрівання відрізка.

Завдання для лабораторної роботи:

1. Вивести на екран монітора результати роботи кожного із алгоритмів, що розглядаються.
2. Оцінити (порівняти) швидкодію кожного із алгоритмів растрівання відрізків.
3. Відтворити на екрані монітора власне прізвище.

Результат роботи





DDA

Bresenham's
lineBresenham's
circle

Wu's line

Clean

DDA

-Time required to draw a line:18
-The time required to draw a surname:31
-Total time:49

Bresenham's line

-Time required to draw a line:18
-The time required to draw a surname:34
-Total time:52

Bresenham's circle:

-Time:27

Wu's line

-Time required to draw a line:395
-The time required to draw a surname:29
-Total time:425

Порівняння роботи алгоритмів

Алгоритм\Час(мс)		1 запуск		2 пуск		3 пуск	
		Пряма	Прізвище	пряма	прізвище	пряма	прізвище
ЦДА		18	31	18	30	18	31
Брезенхема	лінія	18	34	18	30	20	30
	коло	27	-	24	-	26	-
Бу		395	29	346	31	329	34

Код програми([github](#))

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace lab1
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.SetHighDpiMode(HighDpiMode.SystemAware);
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
```

```

using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;

namespace lab1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            int[,] LastName = {
                { 10, 10, 50, 10, 10, 10, 10, 100, 10, 100, 50, 100, -1, -1, -1, -1, -
1, -1, -1, -1, -1, -1, -1, -1}, //C
                { 100, 60, 100, 100, 80, 60, 120, 60, -1, -1, -1, -1, -1, -1, -1, -1, -
1, -1, -1, -1, -1, -1, -1, -1}, //T
                { 150, 60, 180, 60, 150, 75, 180, 75, 150, 60, 150, 100, 150, 100,
180, 100, 180, 60, 180, 75, -1, -1, -1, -1}, //e
                { 210, 60, 210, 95, 235, 60, 235, 95, 210, 95, 240, 95, 240, 95,
240, 100, -1, -1, -1, -1, -1, -1, -1, -1}, //Ц
                { 270, 60, 270, 100, 270, 80, 280, 80, 290, 60, 280, 80, 290, 60,
300, 80, 300, 80, 290, 100, 280, 80, 290, 100}, //ю
                { 330, 60, 350, 60, 330, 60, 330, 100, 330, 75, 350, 75, 350, 60, 350,
75, -1, -1, -1, -1, -1, -1, -1}, //p
                { 380, 60, 410, 60, 380, 75, 410, 75, 380, 60, 380, 100, 380, 100,
410, 100, 410, 60, 410, 75, -1, -1, -1, -1}, //e
                { 440, 60, 440, 100, 470, 60, 470, 100, 440, 80, 470, 80, -1, -1, -1, -
1, -1, -1, -1, -1, -1, -1, -1}, //H
                { 500, 60, 500, 100, 500, 80, 530, 60, 500, 80, 530, 100, -1, -1, -1, -
1, -1, -1, -1, -1, -1, -1, -1}, //K
                { 575, 60, 560, 80, 575, 60, 590, 80, 590, 80, 575, 100, 560, 80,
575, 100, -1, -1, -1, -1, -1, -1, -1, -1} //o
            };

            private static void PutPixel(Graphics g, Color col, float x, float y,
int alpha) //Метод, що встановлює пікселі на формі із заданим кольором і
прозорістю
            {
                g.FillRectangle(new SolidBrush(Color.FromArgb(alpha, col)), x, y,
1, 1);
            }

            public static void DDALine(Graphics g, Color clr, float x0, float y0,
float x1, float y1)
            {
                float step, x, y;
                float dx = (x1 - x0);
                float dy = (y1 - y0);
                if (Math.Abs(dx) >= Math.Abs(dy))
                    step = (Math.Abs(dx));
                else
                    step = (Math.Abs(dy));
                dx = dx / step;
            }
        }
    }
}

```

```

        dy = dy / step;
        x = x0;
        y = y0;
        for(int i=1; i <=step; i++)
        {
            PutPixel(g, clr, x, y, 255);
            x = x + dx;
            y = y + dy;
        }
    }

    private void button1_Click(object sender, EventArgs e) //DDA
    {
        Stopwatch sLine = new Stopwatch();
        Stopwatch sLastName = new Stopwatch();
        Stopwatch sWatch = new Stopwatch();

        sLine.Start();
        sWatch.Start();

        Graphics g = pictureBox1.CreateGraphics();
        DDALine(g, Color.Black, 100, 250, pictureBox1.Width - 50,
pictureBox1.Height - 50);

        sLine.Stop();

        sLastName.Start();

        int rows = LastName.GetUpperBound(0) + 1;    // кількість рядків
        int columns = LastName.Length / rows;        // кількість рядків

        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < columns; j = j + 4)
            {
                if (LastName[i, j] > 0)
                    DDALine(g, Color.Black, LastName[i, j], LastName[i,
j+1], LastName[i, j+2], LastName[i, j+3]);
            }
        }
        sLastName.Stop();
        sWatch.Stop();

        Console.WriteLine("\n\nDDA");
        Console.WriteLine("-Time required to draw a line:");
        Console.WriteLine(sLine.ElapsedMilliseconds.ToString());
        Console.WriteLine("\n-The time required to draw a surname:");
        Console.WriteLine(sLastName.ElapsedMilliseconds.ToString());
        Console.WriteLine("\n-Total time:");
        Console.WriteLine(sWatch.ElapsedMilliseconds.ToString());
    }

    public static void BresenhamLine(Graphics g, Color clr, float x0, float
y0, float x1, float y1)
    {
        //Зміни координат
        float dx = (x1 > x0) ? (x1 - x0) : (x0 - x1);
        float dy = (y1 > y0) ? (y1 - y0) : (y0 - y1);
        //Напрямок збільшення
        float sx = (x1 >= x0) ? (1) : (-1);
        float sy = (y1 >= y0) ? (1) : (-1);
    }

```

```

if (dy < dx)
{
    float d = (dy * 2) - dx; //(dy << 1) - dx;
    float d1 = dy * 2; //dy << 1;
    float d2 = (dy - dx) * 2; //(dy - dx) << 1;
    PutPixel(g, clr, x0, y0, 255);
    float x = x0 + sx;
    float y = y0;
    for (int i = 1; i <= dx; i++)
    {
        if (d > 0)
        {
            d += d2;
            y += sy;
        }
        else
            d += d1;
        PutPixel(g, clr, x, y, 255);
        x++;
    }
}
else
{
    float d = (dx * 2) - dy; //(dx << 1) - dy;
    float d1 = dx * 2; // dx << 1;
    float d2 = (dx - dy) * 2; //(dx - dy) << 1;
    PutPixel(g, clr, x0, y0, 255);
    float x = x0;
    float y = y0 + sy;
    for (int i = 1; i <= dy; i++)
    {
        if (d > 0)
        {
            d += d2;
            x += sx;
        }
        else
            d += d1;
        PutPixel(g, clr, x, y, 255);
        y++;
    }
}
}

```

```

private void button2_Click(object sender, EventArgs e) //Bresenham's
line
{
    Stopwatch sLine = new Stopwatch();
    Stopwatch sLastName = new Stopwatch();
    Stopwatch sWatch = new Stopwatch();

    sLine.Start();
    sWatch.Start();

    Graphics g = pictureBox1.CreateGraphics();
    BresenhamLine(g, Color.Black, 100, 250, pictureBox1.Width - 50,
pictureBox1.Height - 50);

    sLine.Stop();
}

```

```

sLastName.Start();

int rows = LastName.GetUpperBound(0) + 1;    // кількість рядків
int columns = LastName.Length / rows;        // кількість рядків

for (int i = 0; i < rows; i++)
{
    for (int j = 0; j < columns; j = j + 4)
    {
        if (LastName[i, j] > 0)
            BresenhamLine(g, Color.Black, LastName[i, j],
LastName[i, j + 1], LastName[i, j + 2], LastName[i, j + 3]);
    }
}
sLastName.Stop();
sWatch.Stop();

Console.WriteLine("\n\nBresenham's line");
Console.WriteLine("-Time required to draw a line:");
Console.WriteLine(sLine.ElapsedMilliseconds.ToString());
Console.WriteLine("\n-The time required to draw a surname:");
Console.WriteLine(sLastName.ElapsedMilliseconds.ToString());
Console.WriteLine("\n-Total time:");
Console.WriteLine(sWatch.ElapsedMilliseconds.ToString());
}

public static void BresenhamCircle(Graphics g, Color clr, float rx,
float ry, float radius)
{
    float x = 0, y = radius, gap = 0, delta = (2 - 2 * radius);
    while (y >= 0)
    {
        PutPixel(g, clr, rx + x, ry + y, 255);
        PutPixel(g, clr, rx + x, ry - y, 255);
        PutPixel(g, clr, rx - x, ry - y, 255);
        PutPixel(g, clr, rx - x, ry + y, 255);
        gap = 2 * (delta + y) - 1;
        if (delta < 0 && gap <= 0)
        {
            x++;
            delta += 2 * x + 1;
            continue;
        }
        if (delta > 0 && gap > 0)
        {
            y--;
            delta -= 2 * y + 1;
            continue;
        }
        x++;
        delta += 2 * (x - y);
        y--;
    }
}

private void button5_Click(object sender, EventArgs e) //Bresenham's
circle
{
    Stopwatch sWatch = new Stopwatch();

    sWatch.Start();

```



```

Graphics g = pictureBox1.CreateGraphics();
BresenhamCircle(g, Color.Black, 350, 165, 150);

sWatch.Stop();

Console.WriteLine("\n\nBresenham's circle:");
Console.WriteLine("\n-Time:");
Console.WriteLine(sWatch.ElapsedMilliseconds.ToString());
}

//Целая часть числа
private static int IPart(float x)
{
    return (int)x;
}

//дробная часть числа
private static float FPart(float x)
{
    while (x >= 0)
        x--;
    x++;
    return x;
}

public static void DrawWuLine(Graphics g, Color clr, int x0, int y0,
int x1, int y1)
{
    //Вычисление изменения координат
    int dx = (x1 > x0) ? (x1 - x0) : (x0 - x1);
    int dy = (y1 > y0) ? (y1 - y0) : (y0 - y1);
    //Если линия параллельна одной из осей, рисуем обычную линию -
заполняем все пиксели в ряд
    if (dx == 0 || dy == 0)
    {
        g.DrawLine(new Pen(clr), x0, y0, x1, y1);
        return;
    }

    //Для X-линии (коэффициент наклона < 1)
    if (dy < dx)
    {
        //Первая точка должна иметь меньшую координату X
        if (x1 < x0)
        {
            x1 += x0; x0 = x1 - x0; x1 -= x0;
            y1 += y0; y0 = y1 - y0; y1 -= y0;
        }
        //Относительное изменение координаты Y
        float grad = (float)dy / dx;
        //Промежуточная переменная для Y
        float intery = y0 + grad;
        //Первая точка
        PutPixel(g, clr, x0, y0, 255);

        for (int x = x0 + 1; x < x1; x++)
        {
            //Верхняя точка
            PutPixel(g, clr, x, IPart(intery), (int)(255 -
FPart(intery) * 255));
            //Нижняя точка

```

```

        PutPixel(g, clr, x, IPart(intery) + 1, (int)(FPart(intery)
* 255));

        //Изменение координаты Y
        intery += grad;
    }
    //Последняя точка
    PutPixel(g, clr, x1, y1, 255);
}
//Для Y-линии (коэффициент наклона > 1)
else
{
    //Первая точка должна иметь меньшую координату Y
    if (y1 < y0)
    {
        x1 += x0; x0 = x1 - x0; x1 -= x0;
        y1 += y0; y0 = y1 - y0; y1 -= y0;
    }
    //Относительное изменение координаты X
    float grad = (float)dx / dy;
    //Промежуточная переменная для X
    float interx = x0 + grad;
    //Первая точка
    PutPixel(g, clr, x0, y0, 255);

    for (int y = y0 + 1; y < y1; y++)
    {
        //Верхняя точка
        PutPixel(g, clr, IPart(interx), y, 255 -
(int)(FPart(interx) * 255));
        //Нижняя точка
        PutPixel(g, clr, IPart(interx) + 1, y, (int)(FPart(interx)
* 255));

        //Изменение координаты X
        interx += grad;
    }
    //Последняя точка
    PutPixel(g, clr, x1, y1, 255);
}
}

private void button3_Click(object sender, EventArgs e) //Wu's line
{
    Stopwatch sLine = new Stopwatch();
    Stopwatch sLastName = new Stopwatch();
    Stopwatch sWatch = new Stopwatch();

    sLine.Start();
    sWatch.Start();

    Graphics g = pictureBox1.CreateGraphics();
    DrawWuLine(g, Color.Black, 100, 250, pictureBox1.Width - 150,
pictureBox1.Height - 100);

    sLine.Stop();

    sLastName.Start();

    int rows = LastName.GetUpperBound(0) + 1; // кількість рядків
    int columns = LastName.Length / rows; // кількість рядків

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

```

```

        {
            for (int j = 0; j < columns; j = j + 4)
            {
                if (LastName[i, j] > 0)
                    BresenhamLine(g, Color.Black, LastName[i, j],
LastName[i, j + 1], LastName[i, j + 2], LastName[i, j + 3]);
            }
        }
        sLastName.Stop();
        sWatch.Stop();

        Console.WriteLine("\n\nWu's line");
        Console.WriteLine("-Time required to draw a line:");
        Console.WriteLine(sLine.ElapsedMilliseconds.ToString());
        Console.WriteLine("\n-The time required to draw a surname:");
        Console.WriteLine(sLastName.ElapsedMilliseconds.ToString());
        Console.WriteLine("\n-Total time:");
        Console.WriteLine(sWatch.ElapsedMilliseconds.ToString());
    }

    private void button4_Click(object sender, EventArgs e) // clean
    {
        pictureBox1.Image = null;
    }

    private void Form1_Load(object sender, EventArgs e)
    {
    }
}
}

```

Form1.Designer.cs

```

namespace lab1
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}

```

```
#region Windows Form Designer generated code
```

```
/// <summary>
```

```
/// Required method for Designer support - do not modify
```

```
/// the contents of this method with the code editor.
```

```
/// </summary>
```

```
private void InitializeComponent()
```

```
{
```

```
    this.button1 = new System.Windows.Forms.Button();
```

```
    this.pictureBox1 = new System.Windows.Forms.PictureBox();
```

```
    this.button2 = new System.Windows.Forms.Button();
```

```
    this.button3 = new System.Windows.Forms.Button();
```

```
    this.button4 = new System.Windows.Forms.Button();
```

```
    this.button5 = new System.Windows.Forms.Button();
```

```
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
```

```
    this.SuspendLayout();
```

```
    //
```

```
    // button1
```

```
    //
```

```
    this.button1.Location = new System.Drawing.Point(49, 576);
```

```
    this.button1.Name = "button1";
```

```
    this.button1.Size = new System.Drawing.Size(94, 29);
```

```
    this.button1.TabIndex = 0;
```

```
    this.button1.Text = "DDA";
```

```
    this.button1.UseVisualStyleBackColor = true;
```

```
    this.button1.Click += new System.EventHandler(this.button1_Click);
```

```
    //
```

```
    // pictureBox1
```

```
    //
```

```
    this.pictureBox1.Location = new System.Drawing.Point(30, 33);
```

```
    this.pictureBox1.Name = "pictureBox1";
```

```
    this.pictureBox1.Size = new System.Drawing.Size(800, 500);
```

```
    this.pictureBox1.TabIndex = 1;
```

```
    this.pictureBox1.TabStop = false;
```

```
    //
```

```
    // button2
```

```
    //
```

```
    this.button2.Location = new System.Drawing.Point(203, 565);
```

```
    this.button2.Name = "button2";
```

```
    this.button2.Size = new System.Drawing.Size(110, 52);
```

```
    this.button2.TabIndex = 2;
```

```
    this.button2.Text = "Bresenham's line";
```

```
    this.button2.UseVisualStyleBackColor = true;
```

```
    this.button2.Click += new System.EventHandler(this.button2_Click);
```

```
    //
```

```
    // button3
```

```
    //
```

```
    this.button3.Location = new System.Drawing.Point(544, 577);
```

```
    this.button3.Name = "button3";
```

```
    this.button3.Size = new System.Drawing.Size(94, 29);
```

```
    this.button3.TabIndex = 3;
```

```
    this.button3.Text = "Wu's line";
```

```
    this.button3.UseVisualStyleBackColor = true;
```

```
    this.button3.Click += new System.EventHandler(this.button3_Click);
```

```
    //
```

```
    // button4
```

```
    //
```

```
    this.button4.Location = new System.Drawing.Point(696, 576);
```

```
    this.button4.Name = "button4";
```

```
    this.button4.Size = new System.Drawing.Size(94, 29);
```

```

        this.button4.TabIndex = 4;
        this.button4.Text = "Clean";
        this.button4.UseVisualStyleBackColor = true;
        this.button4.Click += new System.EventHandler(this.button4_Click);
        //
        // button5
        //
        this.button5.Location = new System.Drawing.Point(375, 565);
        this.button5.Name = "button5";
        this.button5.Size = new System.Drawing.Size(110, 51);
        this.button5.TabIndex = 5;
        this.button5.Text = "Bresenham\'s circle";
        this.button5.UseVisualStyleBackColor = true;
        this.button5.Click += new System.EventHandler(this.button5_Click);
        //
        // Form1
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 20F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(882, 653);
        this.Controls.Add(this.button5);
        this.Controls.Add(this.button4);
        this.Controls.Add(this.button3);
        this.Controls.Add(this.button2);
        this.Controls.Add(this.pictureBox1);
        this.Controls.Add(this.button1);
        this.Name = "Form1";
        this.Text = "Lab1 by Stetiurenko";
        this.Load += new System.EventHandler(this.Form1_Load);

        ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
        this.ResumeLayout(false);

    }

#endregion

private System.Windows.Forms.Button button1;
private System.Windows.Forms.PictureBox pictureBox1;
private System.Windows.Forms.Button button2;
private System.Windows.Forms.Button button3;
private System.Windows.Forms.Button button4;
private System.Windows.Forms.Button button5;
}
}

```