

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ІНФОРМАТИКИ

Кафедра дискретного аналізу та
інтелектуальних систем

ПАРАЛЕЛЬНІ ТА РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ
ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №4
“ФОРМИ”

Роботу виконав:

Сенюк Віталій Васильович

Студент групи ПМІ-33с

Перевірів:

Доц. Пасічник Т.В.

кафедри програмування

Львівського національного

Університету імені Івана Франка

Тема: Паралельність у Windows Forms.

Мета роботи: Розпаралелити виконання окремих вікн у Windows Forms.

Завдання

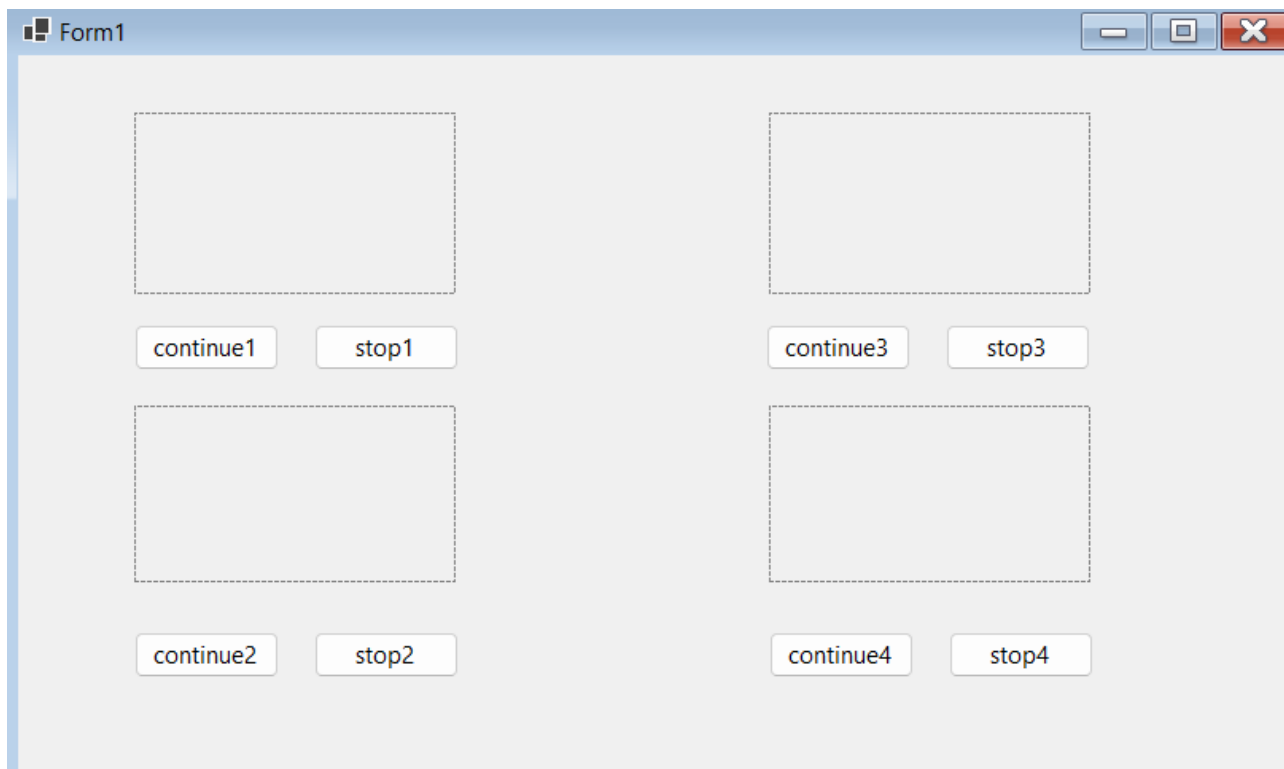
Створити 4 потоки, в кожному з яких зробити графічне представлення його роботи. Тобто в одному потоці літає куля по заданій траєкторії, в іншому прямокутник змінює свої розміри, ще в іншому малюється синусоїда тощо. Ці графічні представлення повинні показувати різні (а не однотипні - sin, cos, tg) процеси та чітко відображати безперервну роботу потоку. В кожній з цих 4-х форм створити дві обов'язкові кнопки: одна призупиняє потік (не завершує!), інша – відновлює роботу потоку з точки зупинки. Ще можливі (за бажанням) кнопки завершення потоку та відкривання потоку (у вікні головного потоку).

Виконання роботи

1. Створюю головну форму з BackgroundWorker для кожного вікна.

```
public partial class Form1 : Form
{
    private BackgroundWorker ballWorker = new BackgroundWorker();
    private BackgroundWorker sinusoidWorker = new BackgroundWorker();
    private BackgroundWorker rectangleWorker = new BackgroundWorker();
    private BackgroundWorker clockWorker = new BackgroundWorker();
    1 reference
    public Form1()
    {
        InitializeComponent();
        // ball
        ballWorker.DoWork += BallWorker_DoWork;
        ballWorker.ProgressChanged += ballWorker_ProgressChanged;
        ballWorker.WorkerReportsProgress = true;
        ballWorker.RunWorkerAsync();
        // sin
        sinusoidWorker.DoWork += SinusoidWorker_DoWork;
        sinusoidWorker.WorkerReportsProgress = true;
        sinusoidWorker.RunWorkerAsync();
        // rectangle
        rectangleWorker.DoWork += RectangleRepresentation_DoWork;
        rectangleWorker.WorkerReportsProgress = true;
        rectangleWorker.RunWorkerAsync();
        // clock
        clockWorker.DoWork += CustomRepresentation_DoWork;
        clockWorker.WorkerReportsProgress = true;
        clockWorker.RunWorkerAsync();
    }
}
```

2. Створюю об'єкти Panel та Button для форми.



3. Для створення малюнку використовую PaintEventHandler.

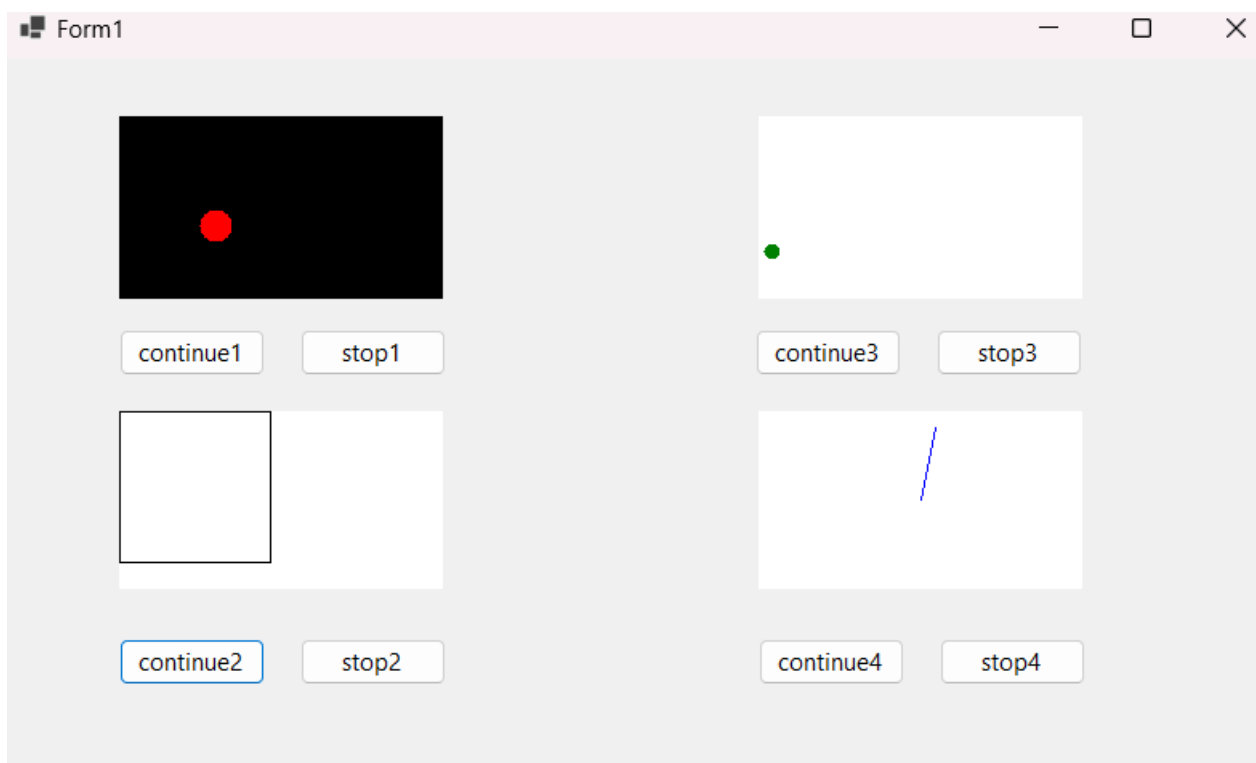
```
//
// rectanglePanel
//
this.rectanglePanel.Location = new System.Drawing.Point(73, 220);
this.rectanglePanel.Name = "rectanglePanel";
this.rectanglePanel.Size = new System.Drawing.Size(202, 111);
this.rectanglePanel.TabIndex = 3;
this.rectanglePanel.Paint += new System.Windows.Forms.PaintEventHandler(this.rectanglePanel_Paint);
//
// circlePanel
//
this.circlePanel.Location = new System.Drawing.Point(472, 220);
this.circlePanel.Name = "circlePanel";
this.circlePanel.Size = new System.Drawing.Size(202, 111);
this.circlePanel.TabIndex = 4;
this.circlePanel.Paint += new System.Windows.Forms.PaintEventHandler(this.circlePanel_Paint);
//
```

4. Для обнови малюнку використовую уже створений BackgroundWorker, та додаю у нього власні DoWorkEventHandlers.

Приклад одного з DoWorkEventHandlers

```
private void BallWorker_DoWork(object sender, DoWorkEventArgs e)
{
    while (true)
    {
        if (isBallRunning)
        {
            if (ballY + ballRadius < ballPanel.Height)
            {
                ballY += 1;
            }
            else
            {
                ballY = 0;
            }
            ballWorker.ReportProgress(0);
            Thread.Sleep(1);
        }
        else
        {
            Thread.Sleep(100);
        }
    }
}
```

5. Приклад роботи програми додав у gif файлі.



Висновки

У результаті виконання роботи:

1. Навчився працювати з Windows Forms.
2. Використав багатопоточність для обчислення зображення, що спростило роботу головного потоку.

Список використаних джерел

1. Вільний простір інтернету.