**ЛАБОРАТОРНА РОБОТА № 5**

з дисципліни

«Системне програмне забезпечення»

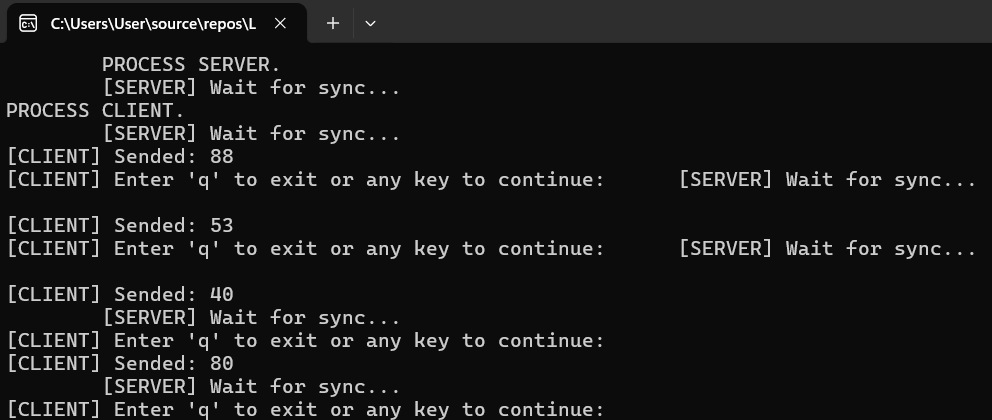
Шейко Р. О.

Групи КІ-21-2

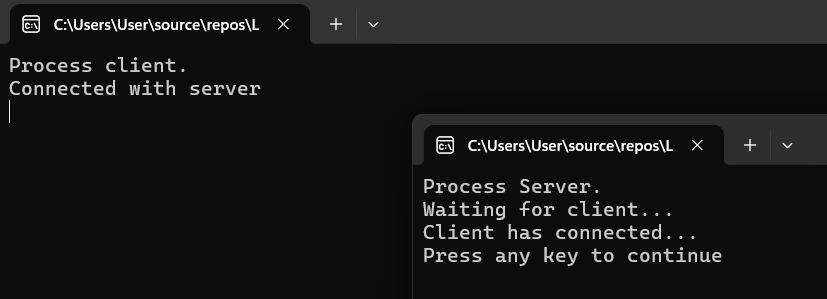
ЗАВДАННЯ

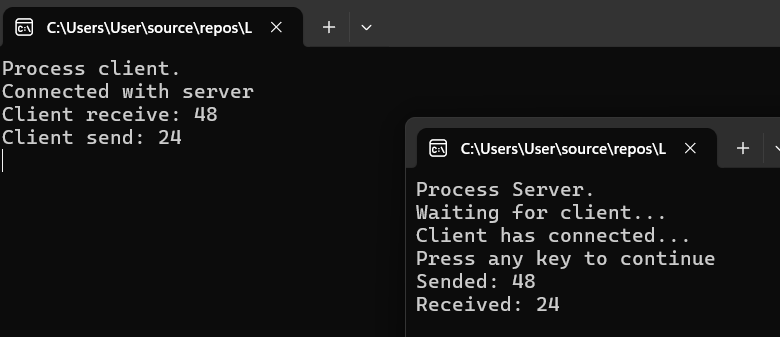
1. Виконати усі приклади програм.

Виконаємо лістинги 5.1 і 5.1.1.

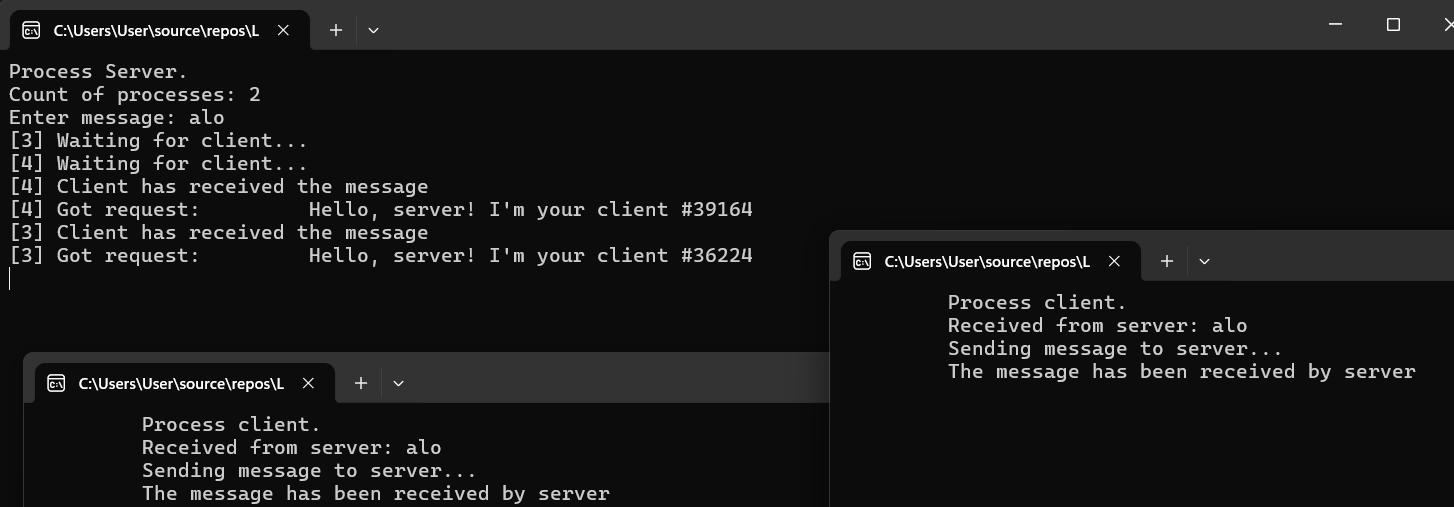


Виконаємо лістинги 5.2 і 5.2.1.





Виконаємо лістинги 5.3 і 5.3.1.



1. Виконати усі приклади програм в проектах C# Windows Forms.

Лістинг 2.1.1

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO.Pipes;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab5\_t2\_1c

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

Console.WriteLine("PROCESS CLIENT.");

Random rand = new Random();

string[] args = Environment.GetCommandLineArgs();

if (args.Length > 1)

{

using (PipeStream pipeClient = new AnonymousPipeClientStream(PipeDirection.Out, args[1]))

{

using (StreamWriter sw = new StreamWriter(pipeClient))

{

sw.AutoFlush = true;

// Send a 'sync message' and wait for client to receive it.

sw.WriteLine("SYNC");

pipeClient.WaitForPipeDrain();

do

{

int num = rand.Next(0, 100);

sw.WriteLine(num);

richTextBox1.AppendText("[CLIENT] Sent: " + num + "\n");

richTextBox1.AppendText("[CLIENT] Enter 'q' to exit or any key to continue: ");

}

while (Console.ReadLine() != "q");

}

}

}

}

}

}

Лістинг 2.1.2

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.IO.Pipes;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab5\_t2\_1s

{

public partial class Form1 : Form

{

private Process client;

private const string EXE\_PATH = @"C:\Users\User\source\repos\Lab5\_t2\_1c\Lab5\_t2\_1c\bin\Debug\Lab5\_t2\_1c.exe";

public Form1()

{

InitializeComponent();

}

private void StartClient(string clientHandle)

{

ProcessStartInfo info = new ProcessStartInfo(EXE\_PATH);

info.Arguments = clientHandle;

info.UseShellExecute = false;

client = Process.Start(info);

}

private void ServerForm\_Load(object sender, EventArgs e)

{

using (AnonymousPipeServerStream pipeServer = new AnonymousPipeServerStream(PipeDirection.In, HandleInheritability.Inheritable))

{

StartClient(pipeServer.GetClientHandleAsString());

pipeServer.DisposeLocalCopyOfClientHandle();

using (StreamReader sr = new StreamReader(pipeServer))

{

string temp;

// Wait for 'sync message' from the client.

do

{

richTextBox1.AppendText("[SERVER] Wait for sync...\n");

temp = sr.ReadLine();

if (temp == null)

break;

}

while (!temp.StartsWith("SYNC"));

richTextBox1.AppendText("[SERVER] Client has connected\n");

do

{

temp = sr.ReadLine();

richTextBox1.AppendText(string.IsNullOrEmpty(temp) ? "[SERVER] No response from client\n" : "[SERVER] Received: " + temp + "\n");

}

while (!client.HasExited);

}

client.Close();

richTextBox1.AppendText("[SERVER] Client quit. Server terminating.\n");

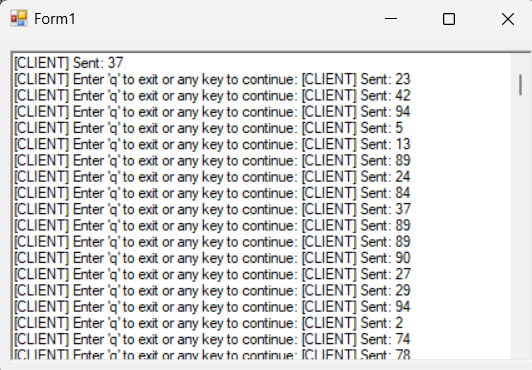
}

}

}

}

Запустимо програму:



Лістинг 2.2.1

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO.Pipes;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab5\_t2\_2s

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

this.Text = "Server";

}

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

{

using (NamedPipeServerStream serverPipe = new NamedPipeServerStream("myPipe", PipeDirection.InOut, 1))

{

MessageBox.Show("Waiting for client...");

serverPipe.WaitForConnection();

MessageBox.Show("Client has connected...");

byte sendData = 48;

serverPipe.WriteByte(sendData);

MessageBox.Show("Server sent: " + sendData.ToString());

int dataReceive = serverPipe.ReadByte();

MessageBox.Show("Server received: " + dataReceive);

}

}

}

}

Лістинг 2.2.2

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO.Pipes;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab5\_t2\_2c

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

this.Text = "Client";

}

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

{

using (NamedPipeClientStream pipe = new NamedPipeClientStream(".", "myPipe", PipeDirection.InOut))

{

pipe.Connect();

int dataReceive = pipe.ReadByte();

MessageBox.Show("Client received: " + dataReceive.ToString());

byte dataSend = 24;

pipe.WriteByte(dataSend);

MessageBox.Show("Client sent: " + dataSend.ToString());

}

}

}

}

|  |  |
| --- | --- |
|  |  |
|  |  |

Лістинг 3.1(c)

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.IO.Pipes;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab5\_t2\_3c

{

public partial class Form1 : Form

{

private readonly NamedPipeClientStream clientPipe;

private readonly int clientID;

private readonly RichTextBox richTextBox;

public Form1(string serverID, RichTextBox richTextBox)

{

clientID = Process.GetCurrentProcess().Id;

clientPipe = new NamedPipeClientStream(".", "pipe" + serverID,

PipeDirection.InOut, PipeOptions.None);

this.richTextBox = richTextBox;

}

public void Start()

{

try

{

// Receive

clientPipe.Connect();

byte[] inBytes = new byte[100];

clientPipe.Read(inBytes, 0, 100);

string inStr = Encoding.ASCII.GetString(inBytes);

AppendText("Received from server: " + inStr);

// Send

AppendText("Sending message to server...");

byte[] outBytes = Encoding.ASCII.GetBytes("Hello, server! I'm your client #" + clientID);

clientPipe.Write(outBytes, 0, outBytes.Length);

// Wait for server to receive the message

clientPipe.WaitForPipeDrain();

AppendText("The message has been received by server");

}

catch (Exception ex)

{

AppendText("Error: " + ex.Message);

}

}

private void AppendText(string text)

{

richTextBox.Invoke((MethodInvoker)delegate {

richTextBox.AppendText(text + "\n");

});

}

}

}

Лістинг 3.2(s)

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Diagnostics;

using System.Drawing;

using System.IO.Pipes;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab5\_t2\_3c

{

public partial class Form1 : Form

{

private readonly NamedPipeClientStream clientPipe;

private readonly int clientID;

private readonly RichTextBox richTextBox;

public Form1(string serverID, RichTextBox richTextBox)

{

clientID = Process.GetCurrentProcess().Id;

clientPipe = new NamedPipeClientStream(".", "pipe" + serverID,

PipeDirection.InOut, PipeOptions.None);

this.richTextBox = richTextBox;

}

public void Start()

{

try

{

// Receive

clientPipe.Connect();

byte[] inBytes = new byte[100];

clientPipe.Read(inBytes, 0, 100);

string inStr = Encoding.ASCII.GetString(inBytes);

AppendText("Received from server: " + inStr);

// Send

AppendText("Sending message to server...");

byte[] outBytes = Encoding.ASCII.GetBytes("Hello, server! I'm your client #" + clientID);

clientPipe.Write(outBytes, 0, outBytes.Length);

// Wait for server to receive the message

clientPipe.WaitForPipeDrain();

AppendText("The message has been received by server");

}

catch (Exception ex)

{

AppendText("Error: " + ex.Message);

}

}

private void AppendText(string text)

{

richTextBox.Invoke((MethodInvoker)delegate {

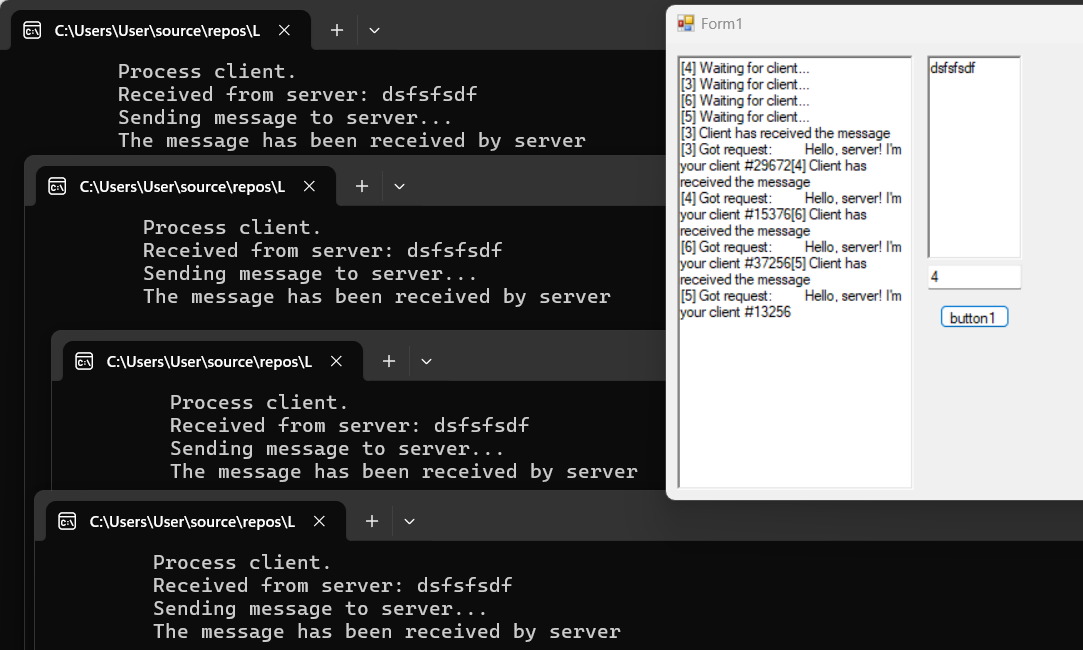
richTextBox.AppendText(text + "\n");

});

}

}

}



3. Написати програму. Використовуючи анонімний канал, сервер надсилає клієнту число і рядок. Клієнт повинен вивести рядок, який він отримує, задану кількість разів.

Лістинг 3.1(сервер)

using System;

using System.Diagnostics;

using System.IO.Pipes;

using System.Threading;

class Program

{

static void Main(string[] args)

{

Console.InputEncoding = System.Text.Encoding.UTF8;

using (AnonymousPipeServerStream pipeServer = new AnonymousPipeServerStream(PipeDirection.Out, HandleInheritability.Inheritable))

{

string clientHandle = pipeServer.GetClientHandleAsString();

Console.WriteLine("Server pipe handle: " + clientHandle);

// Отримання повного шляху до клієнта

string clientPath = Path.Combine(Directory.GetCurrentDirectory(), "Lab5\_4-1.exe");

Console.WriteLine("Client path: " + clientPath);

// Перевірка існування файлу

if (!File.Exists(clientPath))

{

Console.WriteLine("Client executable not found at: " + clientPath);

return;

}

// Запуск клієнта з передачею дескриптора каналу через стандартний вхід

Process clientProcess = new Process();

clientProcess.StartInfo.FileName = clientPath;

clientProcess.StartInfo.UseShellExecute = false;

clientProcess.StartInfo.RedirectStandardInput = true;

try

{

clientProcess.Start();

clientProcess.StandardInput.WriteLine(clientHandle);

clientProcess.StandardInput.Close();

}

catch (Exception ex)

{

Console.WriteLine("Failed to start client process: " + ex.Message);

return;

}

// Надсилання даних клієнту

using (StreamWriter writer = new StreamWriter(pipeServer))

{

writer.AutoFlush = true;

string message = "Hello from Server!";

int count = 5;

writer.WriteLine(message);

writer.WriteLine(count);

}

clientProcess.WaitForExit();

pipeServer.DisposeLocalCopyOfClientHandle();

}

}

}

Лістинг 3.2(кліент)

using System;

using System.IO.Pipes;

class Program

{

static void Main()

{

// Отримання інформації про ідентифікатор каналу від сервера через стандартний вхід

string pipeHandle = Console.ReadLine();

// Підключення до анонімного каналу

using (AnonymousPipeClientStream pipeClient = new AnonymousPipeClientStream(PipeDirection.In, pipeHandle))

{

// Створення читача для отримання даних від сервера

using (StreamReader sr = new StreamReader(pipeClient))

{

// Отримання рядка та числа від сервера

string lineFromServer = sr.ReadLine();

int count = int.Parse(sr.ReadLine());

// Виведення рядка вказану кількість разів

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

{

Console.WriteLine(lineFromServer);

}

}

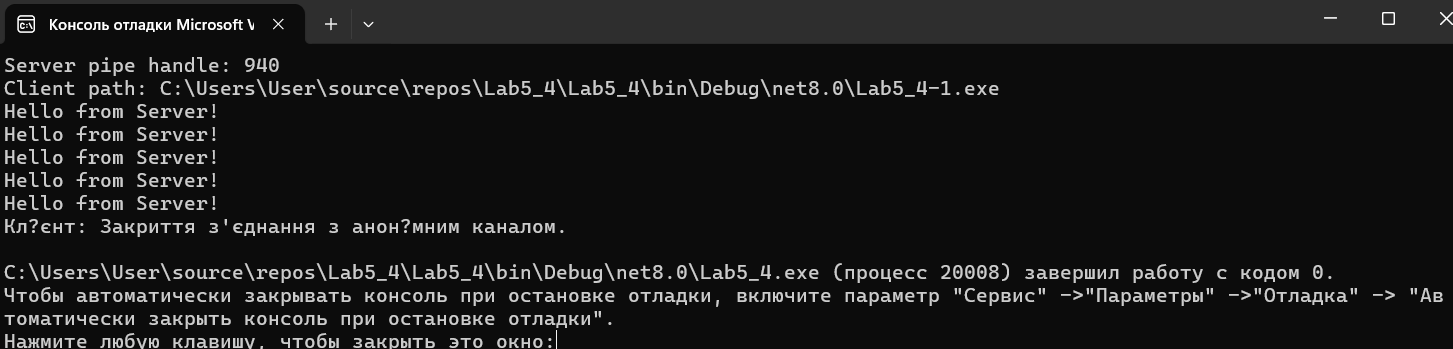
}

Console.WriteLine("Клієнт: Закриття з'єднання з анонімним каналом.");

}

}

Запустимо код:



4. Виконати свій варіант по списку підгрупи в проекті C# Windows Forms, використовуючи іменовані канали:

Варіант 5

Написати програму. У вікні серверного процесу по таймеру фону задається випадковий колір. У вікні клієнтського процесу виводиться RGB-код цього кольору.

// Серверний процес (Server)

using System;

using System.Drawing;

using System.IO.Pipes;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab5\_t4\_v5s

{

public partial class Form1 : Form

{

private NamedPipeServerStream pipeServer;

private Thread serverThread;

private Random random;

public Form1()

{

InitializeComponent();

random = new Random();

StartServer();

}

private void StartServer()

{

serverThread = new Thread(ServerThreadFunc);

serverThread.IsBackground = true;

serverThread.Start();

}

private void ServerThreadFunc()

{

try

{

pipeServer = new NamedPipeServerStream("MyPipe", PipeDirection.Out);

pipeServer.WaitForConnection();

while (true)

{

Color randomColor = Color.FromArgb(random.Next(256), random.Next(256), random.Next(256));

UpdateFormColor(randomColor);

string colorCode = $"{randomColor.R},{randomColor.G},{randomColor.B},";

byte[] buffer = Encoding.ASCII.GetBytes(colorCode);

pipeServer.Write(buffer, 0, buffer.Length);

Thread.Sleep(1000);

}

}

catch (Exception ex)

{

MessageBox.Show("Помилка сервера: " + ex.Message);

}

finally

{

pipeServer?.Close();

}

}

private void UpdateFormColor(Color color)

{

if (this.InvokeRequired)

{

this.Invoke((MethodInvoker)delegate {

this.BackColor = color;

});

}

else

{

this.BackColor = color;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

}

}

// Клієнтський процес (Client)

using System;

using System.Drawing;

using System.IO;

using System.IO.Pipes;

using System.Text;

using System.Threading;

using System.Windows.Forms;

namespace Lab5\_t4\_v5\_2c

{

public partial class Form1 : Form

{

private NamedPipeClientStream pipeClient;

private StreamReader reader;

private Thread clientThread;

public Form1()

{

InitializeComponent();

StartClient();

}

private void StartClient()

{

clientThread = new Thread(ClientThreadFunc);

clientThread.IsBackground = true;

clientThread.Start();

}

private void ClientThreadFunc()

{

try

{

using (pipeClient = new NamedPipeClientStream(".", "MyPipe", PipeDirection.In))

{

pipeClient.Connect();

using (reader = new StreamReader(pipeClient))

{

while (true)

{

string colorCode = reader.ReadLine();

if (colorCode != null)

{

string[] rgb = colorCode.Split(',');

int r = int.Parse(rgb[0]);

int g = int.Parse(rgb[1]);

int b = int.Parse(rgb[2]);

UpdateInterface(r, g, b);

}

}

}

}

}

catch (IOException ex)

{

MessageBox.Show("Сервер відключений.");

}

catch (Exception ex)

{

MessageBox.Show("Помилка клієнта: " + ex.Message);

}

}

private void UpdateInterface(int r, int g, int b)

{

if (this.InvokeRequired)

{

this.Invoke((MethodInvoker)delegate {

this.BackColor = Color.FromArgb(r, g, b);

this.label1.Text = $"RGB: {r}, {g}, {b}";

});

}

else

{

this.BackColor = Color.FromArgb(r, g, b);

this.label1.Text = $"RGB: {r}, {g}, {b}";

}

}

}

}

