Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**ЗВІТ**

**Про виконання практичної роботи № 5**

**з дисципліни «Програмне забезпечення мережевих технологій»**

**на тему**

**«Створення клієнт-серверних програм за допомогою бібліотек C#»**

**Лектор:**

професор кафедри ПЗ

Мельник Р. А.

**Виконав:**

студент групи ПІ-31

Губ’як Р. І.

**Прийняв:**

старший викладач кафедри ПЗ

Гасько Р. Т.

«\_\_\_» \_\_\_\_\_\_\_\_ 2017 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2017

**ТЕМА РОБОТИ**

Створення клієнт-серверних програм за допомогою бібліотек C#

**МЕТА РОБОТИ**

Освоїти елементи програмування мовою C#, а саме засоби створення клієнт-серверних програм

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

Розробка ColorInteraction. Створити дві частини системи: серверну та клієнтську.

Сервер повинен оголошувати про себе у локальній мережі з допомогою протоколу UDP і приймати повідомлення від клієнтів за допомогою TCP-з’єднання.

Клієнт повинен знаходити сервер і відправляти йому повідомлення відомої структури.

**ТЕКСТ ПРОГРАМИ**

//Оголошення знаходження сервера

private void BroadcastAnnounce()

{

byte[] data = Encoding.Unicode.GetBytes(Constants.ImHere);

while (true)

{

foreach (NetworkInterface networkInterface in NetworkInterface.GetAllNetworkInterfaces())

{

if (networkInterface.OperationalStatus is OperationalStatus.Up)

{

foreach (UnicastIPAddressInformation unicast in networkInterface.GetIPProperties().UnicastAddresses)

{

if (unicast.Address.AddressFamily is AddressFamily.InterNetwork)

{

IPAddress broadcast = GetBroadcastAddress(unicast.Address, unicast.IPv4Mask);

using (UdpClient udpClient = new UdpClient())

{

udpClient.Send(data, data.Length, new IPEndPoint(broadcast, 2727));

}

}

}

}

}

Thread.Sleep(5000);

}

}

//Отримання повідомлень від сервера

private void Collect()

{

TcpListener tcpListener = new TcpListener(IPAddress.Any, 2727);

tcpListener.Start();

while (true)

{

using (TcpClient tcpClient = tcpListener.AcceptTcpClient())

using (StreamReader reader = new StreamReader(tcpClient.GetStream()))

{

string data = reader.ReadToEnd();

ColorMessage colorMessage = JsonConvert.DeserializeObject<ColorMessage>(data);

string machine = $"{(tcpClient.Client.RemoteEndPoint as IPEndPoint)?.Address} - {colorMessage.MachineName}";

(Color Color, DateTime LastUpdate) value = (colorMessage.Color, DateTime.Now);

if (Colors.ContainsKey(machine))

{

if (Colors[machine].Color != value.Color)

{

Colors[machine] = value;

OnColorsChanged?.Invoke(machine, value.Color, value.LastUpdate);

}

else

{

Colors[machine] = value;

}

}

else

{

Colors.Add(machine, value);

OnColorsChanged?.Invoke(machine, value.Color, value.LastUpdate);

}

}

}

}

//Прослуховування знаходження сервера

private void Listen()

{

using (UdpClient udpClient = new UdpClient())

{

IPEndPoint serverIpAddress = new IPEndPoint(IPAddress.Any, 2727);

udpClient.Client.Bind(serverIpAddress);

while (true)

{

byte[] data = udpClient.Receive(ref serverIpAddress);

string message = Encoding.Unicode.GetString(data);

if (message == Constants.ImHere && !ServerIpAddress.Equals(serverIpAddress.Address))

{

ServerIpAddress = serverIpAddress.Address;

OnServerIpAddressChanged?.Invoke(ServerIpAddress);

}

}

}

}

//Відправлення повідомлень серверу

private void Generate()

{

Random random = new Random();

while (true)

{

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

ColorMessage colorMessage = new ColorMessage

{

Color = color,

MachineName = Environment.MachineName

};

string data = JsonConvert.SerializeObject(colorMessage);

try

{

using (TcpClient tcpClient = new TcpClient())

{

tcpClient.Connect(new IPEndPoint(ServerIpAddress, 2727));

using (StreamWriter writer = new StreamWriter(tcpClient.GetStream()))

{

writer.Write(data);

}

}

OnColorGenerated?.Invoke(color);

}

catch

{

Console.WriteLine("Unable to connect to the server.");

}

Thread.Sleep(5000);

}

}

**ПРОТОКОЛ ВИКОНАННЯ ПРОГРАМИ**

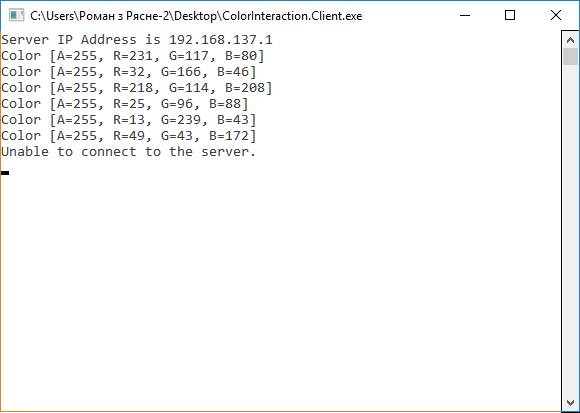


Рис. 1. Протокол роботи клієнта

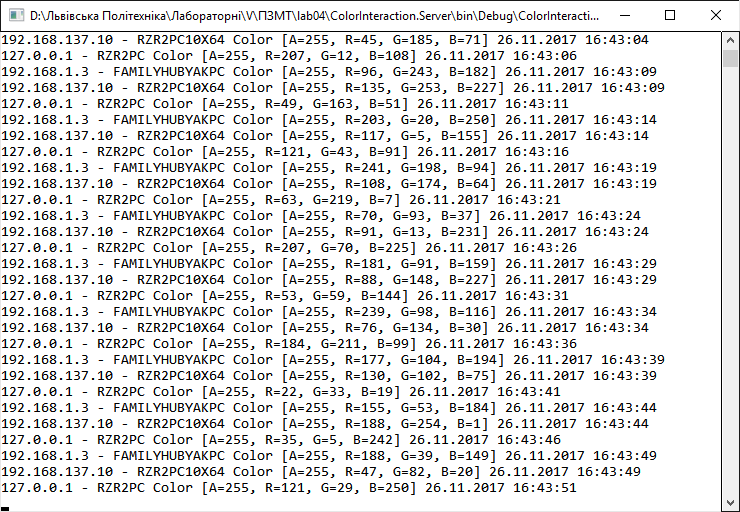


Рис. 2. Протокол роботи сервера

**ВИСНОВОК**

Під час виконання цієї лабораторної роботи я освоїв елементи програмування мовою C#, а саме засоби створення клієнт-серверних програм. Реалізував серверну та клієнтську частини згідно з індивідуальним завданням.