Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**ЗВІТ**

**Про виконання практичної роботи № 4**

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

**на тему**

**«Побудова багатопотокових програм за допомогою стандартних 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**

Побудова багатопотокових програм за допомогою стандартних бібліотек C#

**МЕТА РОБОТИ**

Освоїти елементи програмування мовою C#, а саме засоби створення потоків

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

Розробка ColorInteraction. Створити кілька потоків, кожен з яких виконує певне завдання:

1. Оголошує знаходження сервера.
2. Приймає вхідні повідомлення.
3. Прослуховує знаходження сервера.
4. Відправляє інформацію про комп’ютер і його колір.

**ТЕКСТ ПРОГРАМИ**

internal class Announcer //Оголошує знаходження сервера

{

public void Start()

{

new Thread(BroadcastAnnounce).Start();

}

private void BroadcastAnnounce()

{

}

private static IPAddress GetBroadcastAddress(IPAddress address, IPAddress subnetMask)

{

byte[] ipAdressBytes = address.GetAddressBytes();

byte[] subnetMaskBytes = subnetMask.GetAddressBytes();

byte[] broadcastAddress = new byte[ipAdressBytes.Length];

for (int i = 0; i < broadcastAddress.Length; i++)

{

broadcastAddress[i] = (byte)(ipAdressBytes[i] | ~subnetMaskBytes[i]);

}

return new IPAddress(broadcastAddress);

}

}

internal class Listener //Прослуховує знаходження сервера

{

public delegate void ServerIpAddressChangedHandler(IPAddress ipAddress)

public event ServerIpAddressChangedHandler OnServerIpAddressChanged;

public IPAddress ServerIpAddress { get; private set; } = IPAddress.Any;

public void Start()

{

new Thread(Listen).Start();

}

private void Listen()

{

}

}

internal class ColorCollector //Приймає вхідні повідомлення

{

public Dictionary<string, (Color Color, DateTime LastUpdate)> Colors { get; } = new Dictionary<string, (Color, DateTime)>();

public delegate void ColorsChangedHandler(string machine, Color color, DateTime lastUpdate);

public event ColorsChangedHandler OnColorsChanged;

public void Start()

{

new Thread(Collect).Start();

}

private void Collect()

{

}

}

internal class ColorGenerator //Відправляє інформацію про комп’ютер і його колір

{

public static IPAddress ServerIpAddress;

public delegate void ColorGeneratedHandler(Color color);

public event ColorGeneratedHandler OnColorGenerated;

public void Start()

{

new Thread(Generate).Start();

}

private void Generate()

{

}

}

**ПРОТОКОЛ ВИКОНАННЯ ПРОГРАМИ**

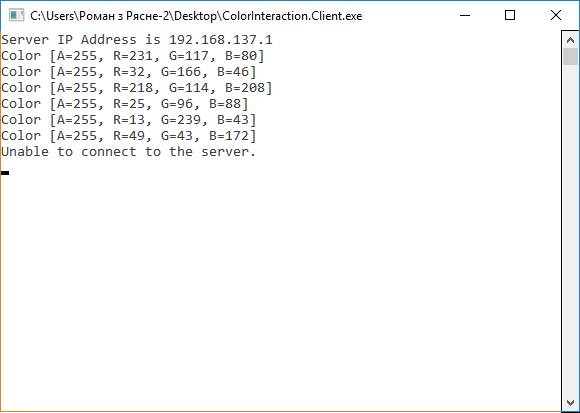


Рис. 1. Протокол роботи клієнта

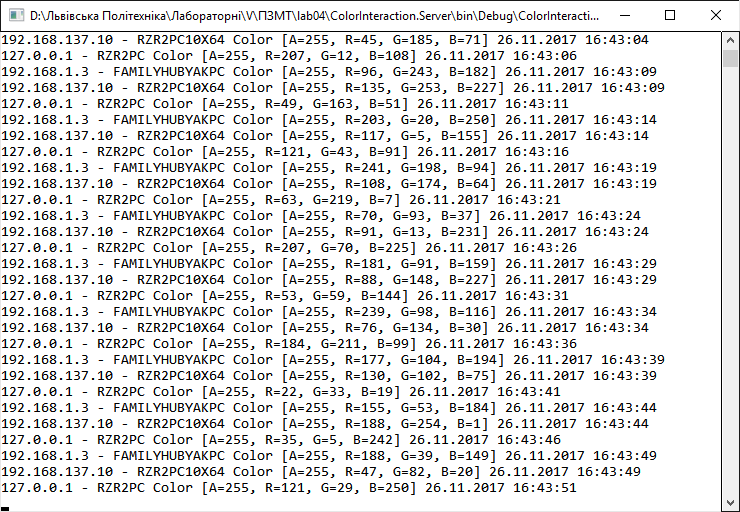


Рис. 2. Протокол роботи сервера

**ВИСНОВОК**

Під час виконання цієї лабораторної роботи я освоїв елементи програмування мовою C#, а саме засоби створення потоків. Створив багатопотокові програми згідно з індивідуальним завданням.