***KODLARIN KULLANIM AMAÇLARI***

***SERVER***

using System;

using System.IO;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Net;

using System.Net.Sockets;

using System.Threading; → //***Bu kodu aynı anda iki metodu çalıstırabilmek için kullanıyoruz.***

using System.Net; ***ve*** using System.Net.Sockets; → //***kodu ağ üzerinde çalışan iki programın (sunucu programı ve istemci programı) arasındaki çift yönlü iletişimi sağlar***

Thread t;

IPAddress ipadresimiz;

TcpListener dinle;

Socket soket;

NetworkStream ag;

StreamReader oku;

StreamWriter yaz;

public delegate void ricdegis(string text)

***//Bu kod dizisi program için gereken sınıfların nesnelerini tanımlıyor.***

soket = dinle.AcceptSocket();

ag = new NetworkStream(soket);

oku = new StreamReader(ag);

while (true)

{

try

{

string yazi = oku.ReadLine();

ekranabas(yazi);

}

catch

{

return;

}

}

NetworkStream ***sınıfı standart akış mekanizması ve Socket sınıf örneği ile ağ veri okuyup algılar.*** NetworkStream ***sınıfı ağ veri ve sistem aynı şekilde yazılmış olmasını sağlar.IO.Akış sınıfı.Bu sınıf, ağ veri içi zaman uyumlu ve zaman uyumsuz eşzamanlı erişim destekler.Rastgele erişim desteklenmiyor ve bu nedenle CanSeek özelliği her zaman döndürülür.***

***CanSeek özelliği geçerli akış aramayı destekleyip desteklemediğini gösteren bir değeri alır.***

StreamReader ***Stream reader metin dosyalrını okur.Sistemi bulur. İyi performans sağlar.Bu yapı, sistem kaynaklarını imha yardımcı olur.***

public void ekranabas(string s)

{

if (this.InvokeRequired)

{

ricdegis degis = new ricdegis(ekranabas);

this.Invoke(degis, s);

}

else

{

s = "Client: " + s;

richTextBox1.AppendText(s + "\n");

}

}

***// 2. method olarak yazılan bu dizi gelen veriyi richTextBox içine yazdırmak için kullanılır***

private void dinlemeye\_basla()

{

try

{

dinle = new TcpListener(IPAddress.Any, 1024);

dinle.Start();

t = new Thread(new ThreadStart(okumayabasla));

t.Start();

richTextBox1.AppendText(DateTime.Now.ToString() + " Dinleme baslatildi..."+"\n");

}

catch (Exception)

{

MessageBox.Show("Dinleme baslatilamadi");

}

}

***// 3. Metot olarak yazılan bu kod dizisi Serverimizin Port dinlemesine baslaması için kullanılır***

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

{

dinlemeye\_basla();

}

***“Dinlemeye Başla” butonuna bastığımızda 3. Methodu çalıştırır ve iki bilgisayar arasında dinlemeyi başlatır.***

private void button2\_Click(object sender, EventArgs e)

{

ag.Close();

}

***“Dinlemeyi Durdur” butonuna bastığımızda 2. Methodu çalıştırarak dinlemeyi bitirir.***

private void button3\_Click(object sender, EventArgs e)

{ if (textBox2.Text == "")

//Burda bos alan göndermeyi önlüyoruz...

return;

else

{

yaz = new StreamWriter(ag);

yaz.WriteLine(textBox2.Text);

yaz.Flush();

richTextBox1.AppendText("Server: "+textBox2.Text + "\n");

textBox2.Text = "";

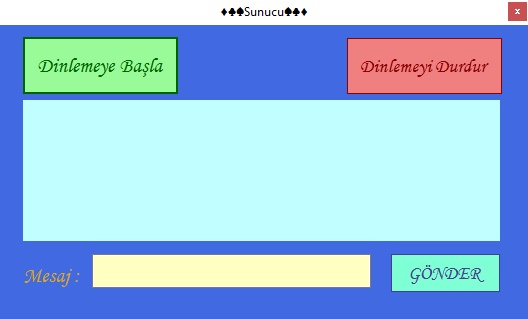
}

}

***3. butonumuz olan “Gönder” butonu yazılan gönderiyi ağda bağlı olduğumuz diğer bilgisayara aktarır.***

}

}

******

***CLİENT***

using System;

using System.IO;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Net;

using System.Net.Sockets;

using System.Threading; → //***Bu kodu aynı anda iki metodu çalıstırabilmek için kullanıyoruz.***

using System.Net; ***ve*** using System.Net.Sockets; → //***kodu ağ üzerinde çalışan iki programın (sunucu programı ve istemci programı) arasındaki çift yönlü iletişimi sağlar***

Thread t;

TcpClient baglantikur;

NetworkStream ag;

StreamReader oku;

StreamWriter yaz;

public delegate void ricdegis(string text)

***// Gerekli sınıfların nesneleri tanımlanıyor***

StreamWriter ***Kolay ve verimli metin çıktısı sağlar.***

public Client()

{

InitializeComponent();

}

***//Gelen veriyi okunuyor***

public void okumayabasla()

{

ag = baglantikur.GetStream();

oku = new StreamReader(ag);

while (true)

{

try

{

string yazi = oku.ReadLine();

ekranabas(yazi);

}

catch

{

return;

}

}

}

***// Okunan verinin richTextBox icine yazılmasını sağlar.***

public void ekranabas(string s)

{

if (this.InvokeRequired)

{

ricdegis degis = new ricdegis(ekranabas);

this.Invoke(degis, s);

}

else

{

s = "Server: " + s;

richTextBox1.AppendText(s + "\n");

}

}

***// İstenilen IP'ye istenen Port üzerinden baglanılmayı sağlıyor***

public void baglanti\_kur()

{

try

{

baglantikur = new TcpClient(textBox1.Text, 1024);

t = new Thread(new ThreadStart(okumayabasla));

t.Start();

richTextBox1.AppendText(DateTime.Now.ToString() + " Baglanti kuruldu..."+"\n");

}

catch (Exception) // istisnai durum yönetimi

{

MessageBox.Show("Server ile baglanti kurulurken hata olustu !");

}

}

***// Bağlantı kurulacak olan bilgisayarın IP adrsini tanıtıyor.***

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

{

baglanti\_kur();

}

***// “Bağlan” butonuna bastığımızda son metot içindeki kodları çalıştırarak IP adresi verilmiş bilgisayara bağlanır.***

private void button2\_Click(object sender, EventArgs e)

{

baglantikur.Client.Close();

}

***// “Bağlantıyı Kes” butonuna basıldığında iki bilgisayar rsındaki bağlantıyı sonlandırır.***

private void button3\_Click(object sender, EventArgs e)

{

if (textBox2.Text == "")

***// Burda bos alan göndermeyi önlüyoruz***

return;

else

{

yaz = new StreamWriter(ag);

yaz.WriteLine(textBox2.Text);

yaz.Flush();

richTextBox1.AppendText("Client: "+textBox2.Text + "\n");

textBox2.Text = "";

}

}

***// “Gönder” butonuna basıldığında texBoxa yazdığımız gönderiyi bağlantı kurulan bilgisayara gönderir.***

