**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Факультет информационных технологий

Дисциплина «Программирование и основы алгоритмизации»

Отчет о лабораторно-практической работе № 2

«Обмен данными по сети Ethernet»

Выполнили: Карпеева Е. С.

Лексанова Е. В.

(группа 181-311)

Проверил: Т.Т. Идиатуллов

Москва, 2019

**Задание:**

Реализовать текстовый чат на UDP с использованием C# и WinForm путем создания приложения, обеспечивающего возможность настройки соединения (удаленный IP и порт, локальный порт), ввод и отправку сообщений (текстовых строк), получение и отображение сообщений в текстовом списке.  
Полученные сообщения должны дополняться информацией о том, с какого компьютера получено сообщение.

**Программный код:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Net;

using System.Threading;

using System.Net.Sockets;

namespace chat

{

public partial class Form1 : Form

{

bool alive = false; // будет ли работать поток для приема

IPAddress ip;

UdpClient udpClient;

UdpClient udpClientReceive;

Thread thread;

int localport; // Порт для приёма сообщений

int remoteport; // Порт для отправки сообщений

public Form1()

{

InitializeComponent();

}

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

{

ip = IPAddress.Parse(IP.Text.Trim()); //Удаленный IP для отправки сообщений

localport = Convert.ToInt32(LOCALPORT.Text); // порт для приема сообщений

remoteport = Convert.ToInt32(REMOTEPORT.Text); // порт для отправки сообщений

udpClient = new UdpClient();

udpClientReceive = new UdpClient(localport);

thread = new Thread(new ThreadStart(ReceiveMessage));

thread.Start();

try

{

connection.Text = "Connection";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void ReceiveMessage()

{

alive = true;

while (alive)

{

IPEndPoint remoteIPEndPoint = new IPEndPoint(IPAddress.Any, localport); ;

byte[] data = udpClientReceive.Receive(ref remoteIPEndPoint);

if (data.Length > 0)

{

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

IPEndPoint remoteFullIp = remoteIPEndPoint as IPEndPoint;

string source = remoteIPEndPoint.Address.ToString() + ":" + remoteIPEndPoint.Port.ToString();

string text = "Friend:" + receivedString.ToString();

this.Invoke(new MethodInvoker(() =>

{

chat.Items.Add(text.ToString());

}));

}

}

}

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

{

try

{

IPAddress ip = IPAddress.Parse(IP.Text.Trim());

IPEndPoint ipEndPoint = new IPEndPoint(ip, remoteport);

string message1 = message.Text;

byte[] data = Encoding.Unicode.GetBytes(message1);

udpClient.Send(data, data.Length, ipEndPoint);

string text = "I: " + message1;

message.Clear();

chat.Items.Add(text.ToString());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Form1\_FormClosed(object sender, FormClosedEventArgs e)

{

IPAddress ip = IPAddress.Parse(IP.Text.Trim());

IPEndPoint ipEndPoint = new IPEndPoint(ip, remoteport);

try

{

alive = false;

udpClient.Close();

udpClientReceive.Close();

thread.Abort();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

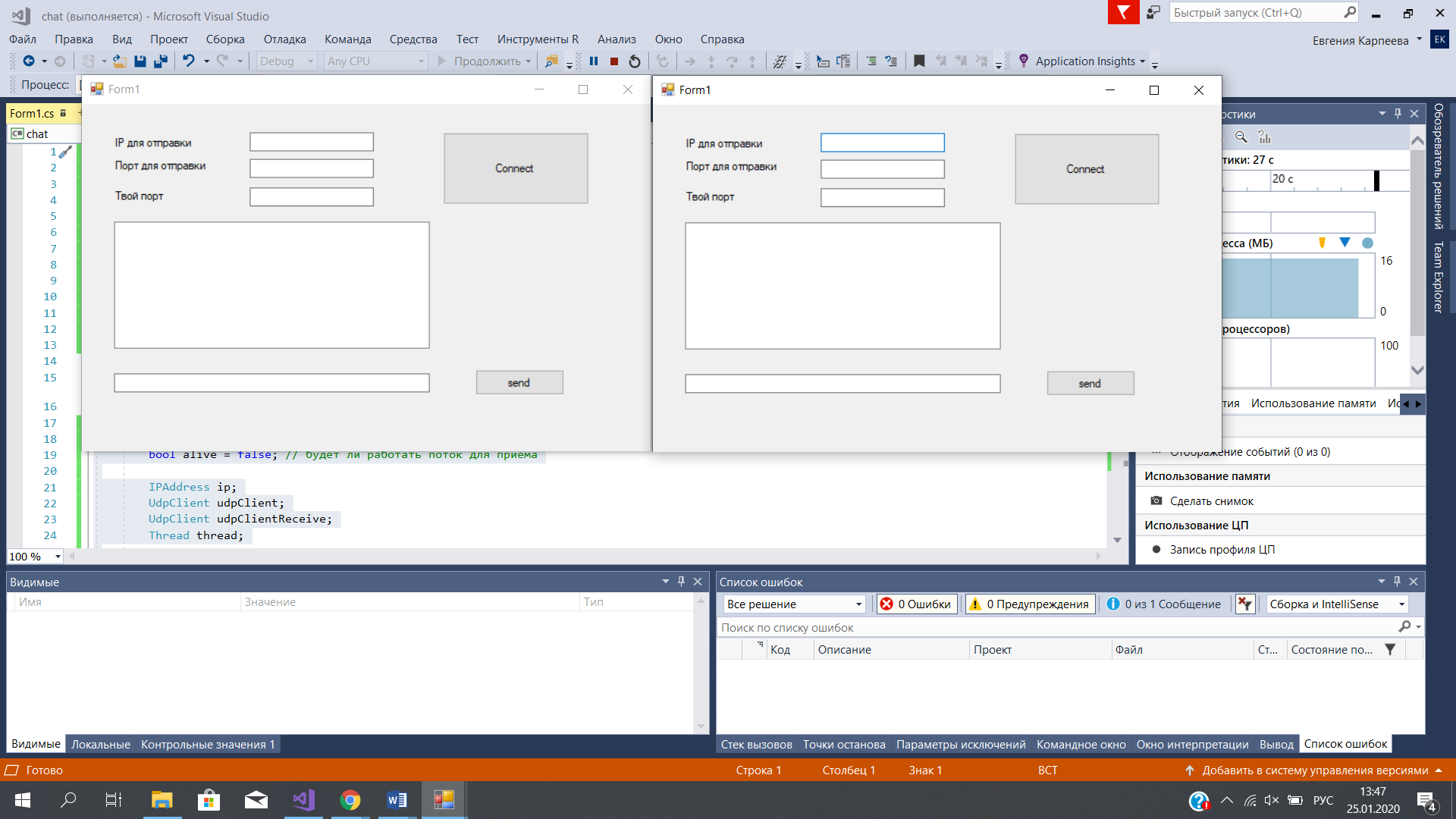
}

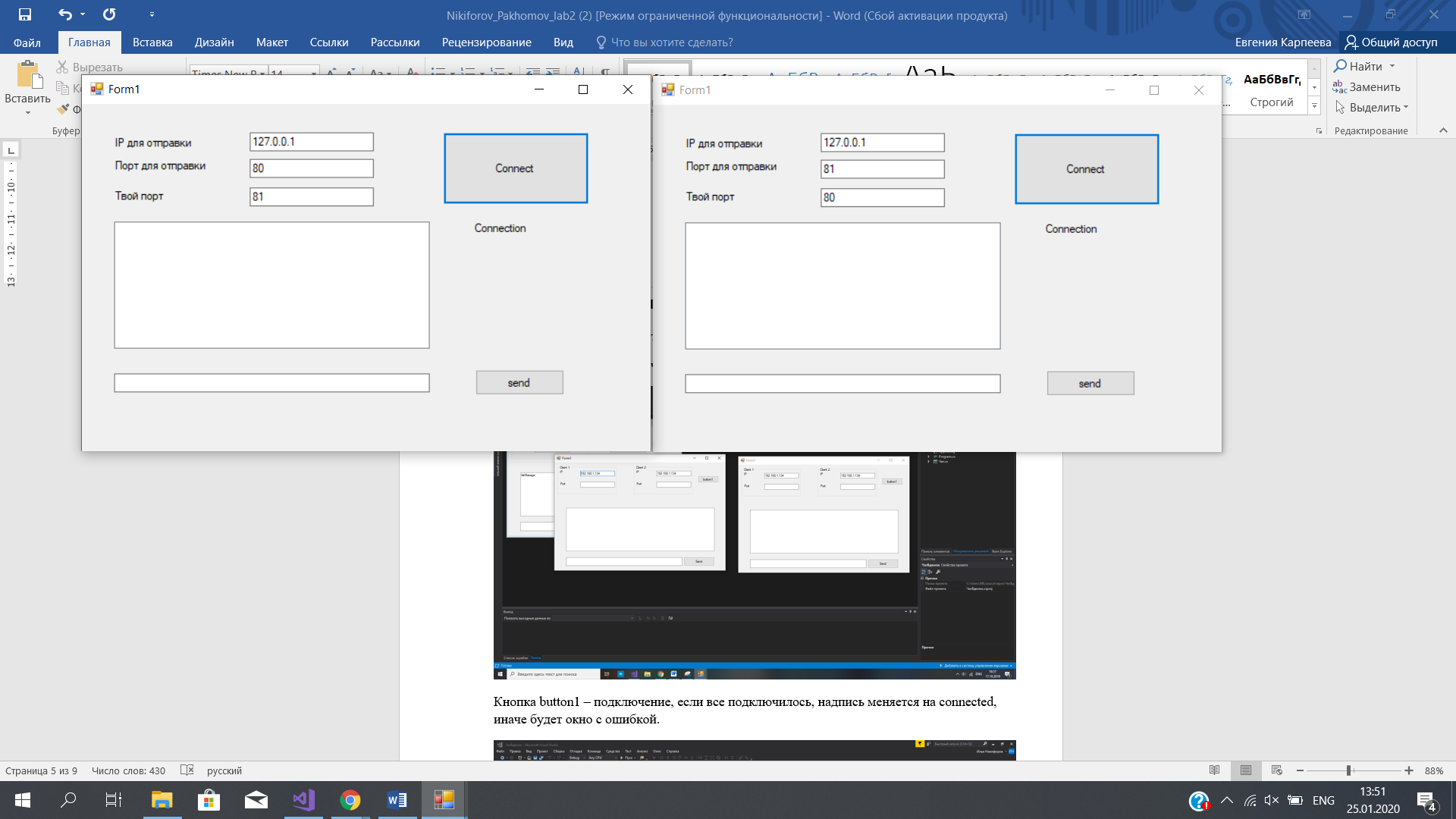
}

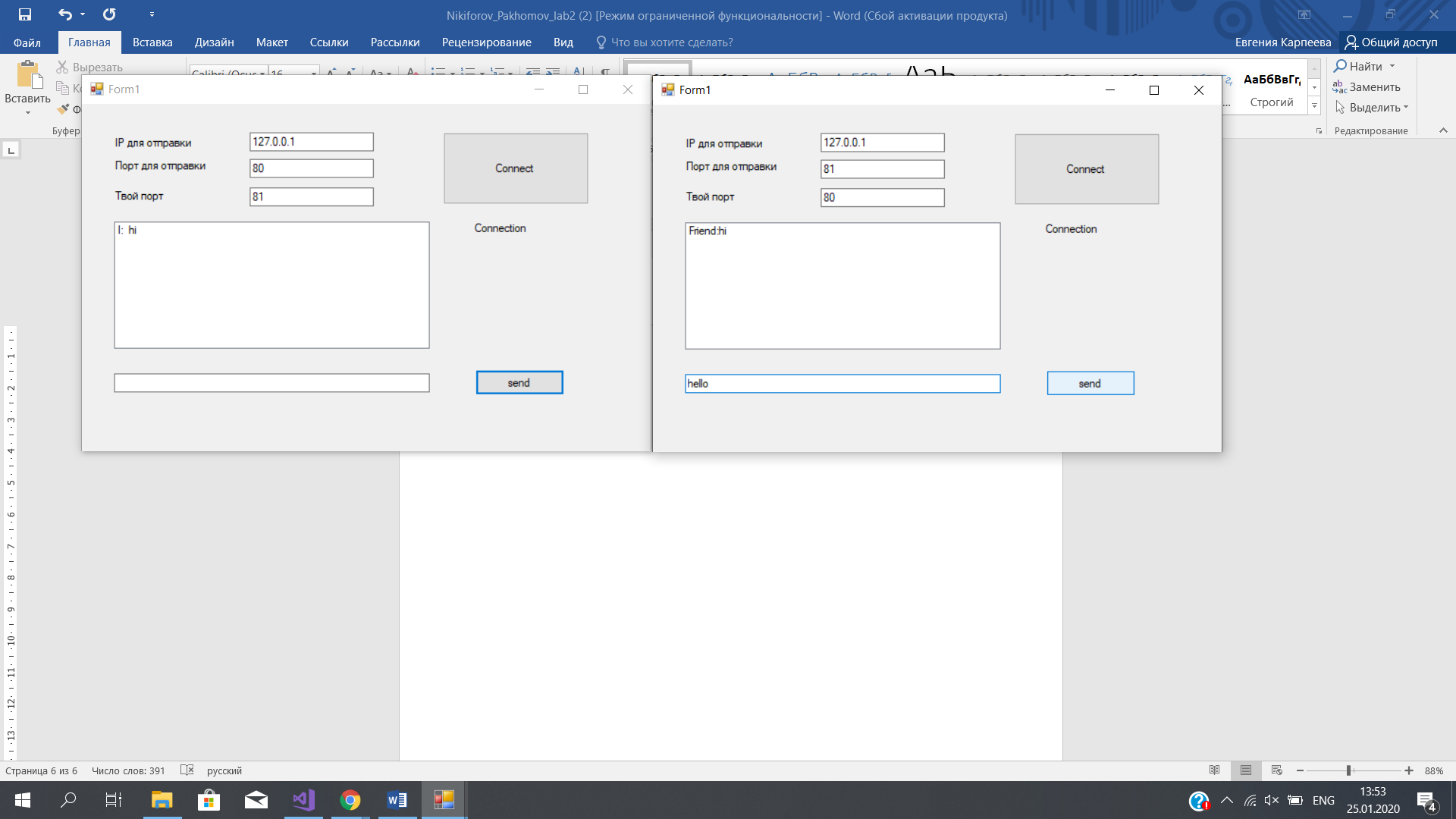
}

}

**Ход работы**

Запускаем чат, вводим Ip и порты, нажимаем кнопку Connect.

После чего отображается сигнал соединения, чатом можно пользоваться.

Мы вводим текст сообщения в нижней строке, нажимаем на кнопку Send. В большом окне в центре отображается вся переписка.