САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАНИЕ   
«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

**ОТЧЕТ**

**по практической работе №14-16**

**«Организация чата»**

Выполнил студент 41 гр.

Березко В.В.

Преподаватель: Фомин А. В.

Оценка: \_\_\_\_\_\_\_\_\_

Санкт-Петербург 2016

**Оглавление**

[1. Цель работы 3](#_Toc467707416)

[2. Ход выполнения работы 4](#_Toc467707417)

[3. Заключение 8](#_Toc467707418)

# Цель работы

Получить навыки по созданию соединения серверного приложения с одним и несколькими клиентами, а также навыки по передаче сообщений между клиентскими приложениями.

# Ход выполнения работы

Для реализации чата сначала надо написать серверную часть.

Для создания сервера поместим на главную форму кнопку запуска сервера и textbox для отображения текущего состояния.

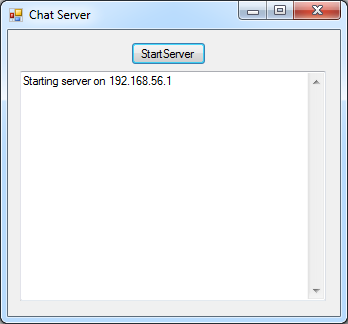


Рисунок - Главная форма сервера

Для создания клиента на главной форме мы разместили 2 textBox для ввода имени в чате и IP адреса сервера и кнопку для подключения к серверу. Для организации общения в чате используется многострочный textbox для отображения сообщений, обычный textbox для написания сообщений и кнопка для отправки сообщений.

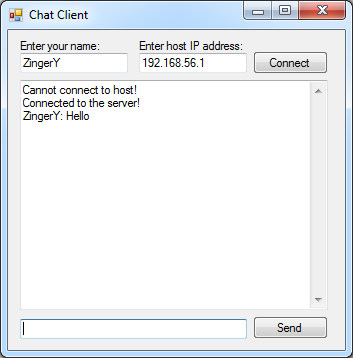


Рисунок - Главная форма клиента

Сервер поддерживает любое количество клиентов:

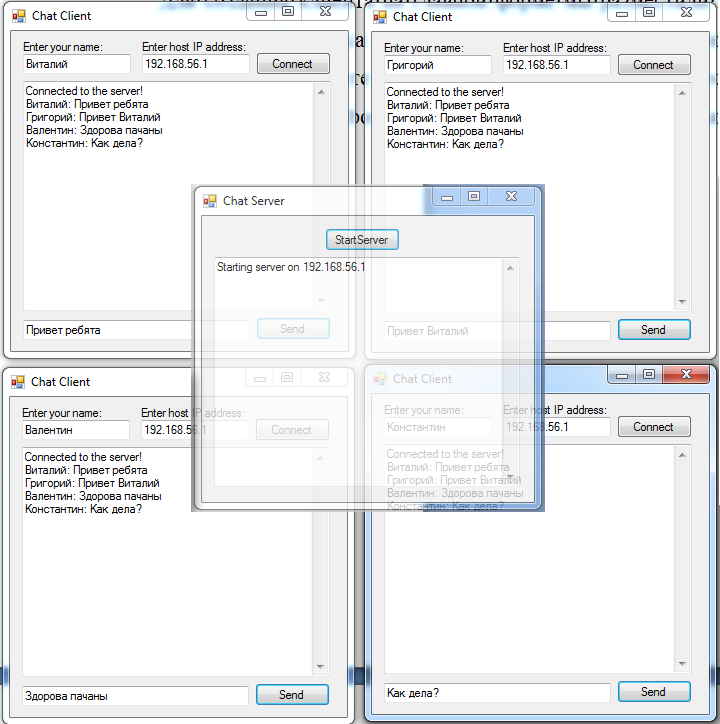


Рисунок - Четыре запущеных клиента и сервер

Для организации работы с клиентами был написан класс ClientData:

class ClientData

{

public Socket clientSocket;

public Thread clientThread;

public string id;

public ClientData()

{

id = Guid.NewGuid().ToString();

clientThread = new Thread(Data\_IN);

clientThread.Start(clientSocket);

SendRegistrationPacket();

}

public ClientData(Socket clientSocket)

{

this.clientSocket = clientSocket;

id = Guid.NewGuid().ToString();

clientThread = new Thread(Data\_IN);

clientThread.Start(clientSocket);

SendRegistrationPacket();

}

public void SendRegistrationPacket()

{

Packet p = new Packet(PacketType.Registration, "server");

p.Gdata.Add(id);

clientSocket.Send(p.ToBytes());

}

}

Экземпляры этого класса добавляются в список, благодаря этому сервер может обслуживать большое количество клиентов:

static List<ClientData> \_clients;

static void ListenThread()

{

while (true)

{

listenerSocket.Listen(0);

\_clients.Add(new ClientData(listenerSocket.Accept()));

}

}

Для организации обмена сообщениями между клиентами и сервером был написан класс Packet при помощи которого можно указывать тип пакета и преобразовывать строки в байтовые массивы для отправки по локальной сети.

public class Packet

{

public List<string> Gdata;

public int packetInt;

public bool packetBool;

public string senderID;

public PacketType packetType;

public Packet(PacketType type, string senderID)

{

Gdata = new List<string>();

this.senderID = senderID;

this.packetType = type;

}

public Packet(byte[] packetbytes)

{

BinaryFormatter bf = new BinaryFormatter();

MemoryStream ms = new MemoryStream(packetbytes);

Packet p = (Packet)bf.Deserialize(ms);

this.Gdata = p.Gdata;

this.packetInt = p.packetInt;

this.packetBool = p.packetBool;

this.senderID = p.senderID;

this.packetType = p.packetType;

}

public byte[] ToBytes()

{

BinaryFormatter bf = new BinaryFormatter();

MemoryStream ms = new MemoryStream();

bf.Serialize(ms, this);

byte[] bytes = ms.ToArray();

ms.Close();

return bytes;

}

public static string GetIP4Address()

{

IPAddress[] ips = Dns.GetHostAddresses(Dns.GetHostName());

foreach(IPAddress i in ips)

{

if (i.AddressFamily == System.Net.Sockets.AddressFamily.InterNetwork)

return i.ToString();

}

return "127.0.0.1";

}

}

# Заключение

Благодаря проделанной работе, я закрепил на конкретных примерах теоретические знания, полученные при изучении свойств, методов и событий стандартных компонентов построения графического пользовательского интерфейса. Получил навыки по созданию соединения серверного приложения с одним и несколькими клиентами, а также навыки по передаче сообщений между клиентскими приложениями. Было разработано прикладное приложение «Чат» для организации общения между несколькими компьютерами в локальной сети.