

## Отчет лабораторной работы № 2.

### Создание клиент-серверного приложения «Протокол».

**Задание.** Создать клиент-серверное приложение со следующим функционалом:

1. Клиент подключается к серверу, обменивается с ним сообщениями и отключается по кодовому слову (задать самостоятельно). В это время остальные клиенты ждут своей очереди.
2. После завершения обслуживания одного клиента, сервер принимает следующего клиента.

### Описание работы

В ходе работы были написаны и отлажены два приложения на языке C#: клиент и сервер. Клиент и сервер обмениваются сообщениями с помощью протокола TCP/IP.

На рисунках 1-3 показано работы сервера и двух клиентов, два клиента отправляют сообщения серверу, по очереди, сервер не получит сообщение от второго клиента пока клиент первый не закроет соединение с сервером.

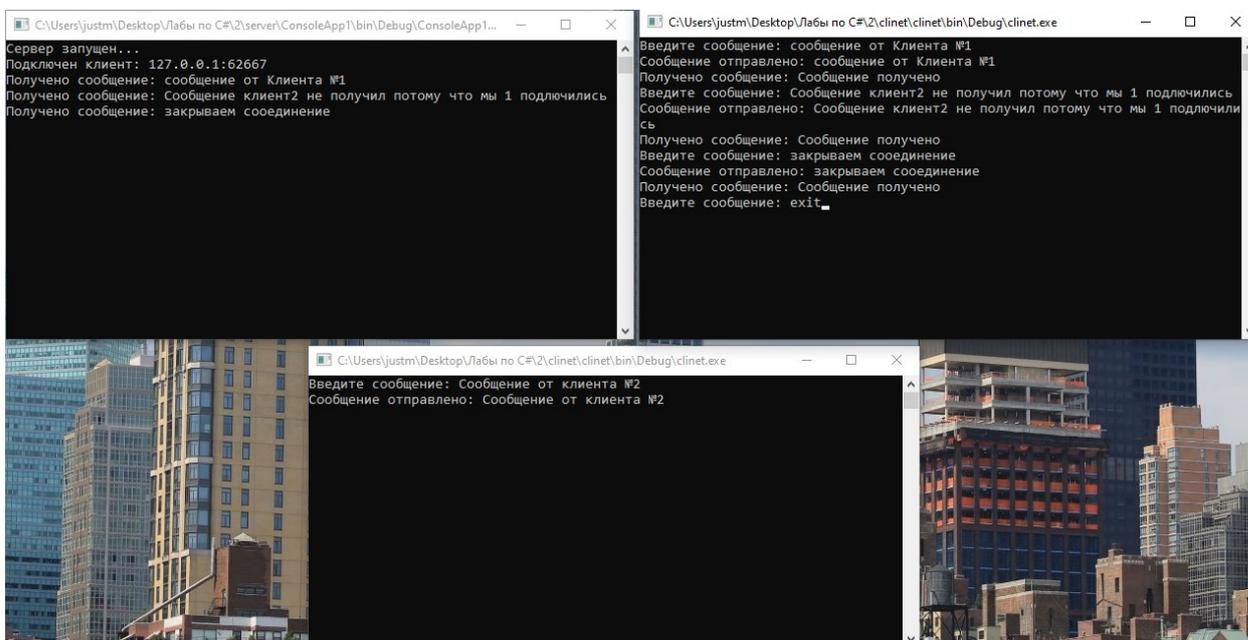


Рисунок 1

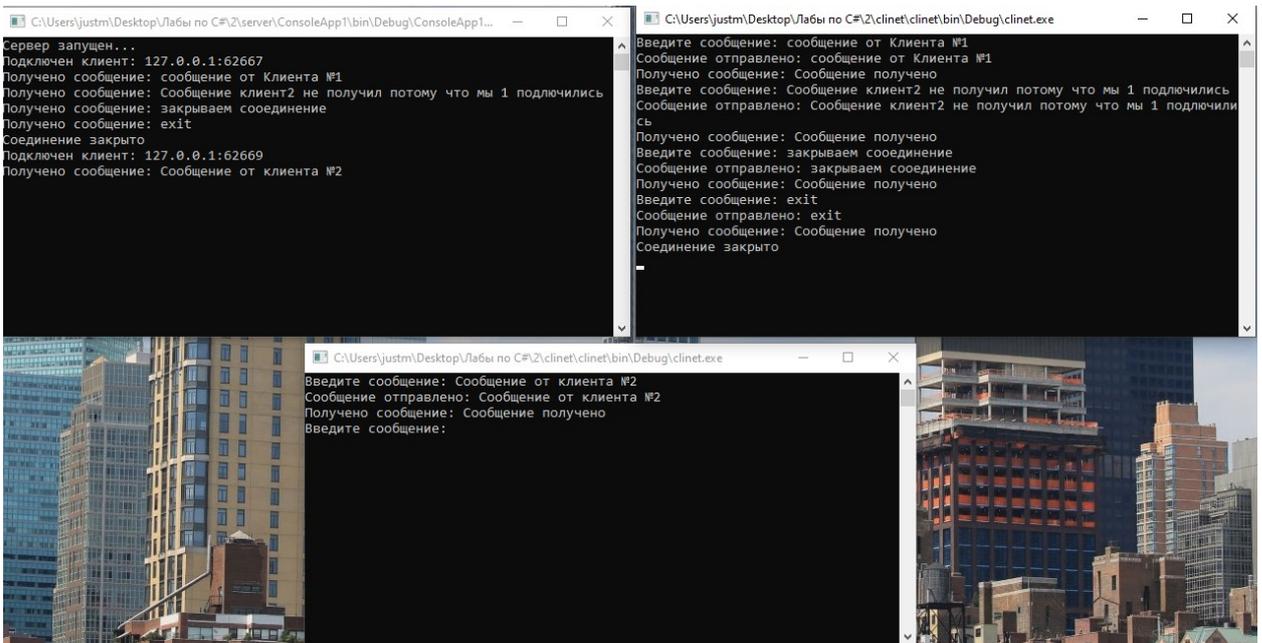


Рисунок 2

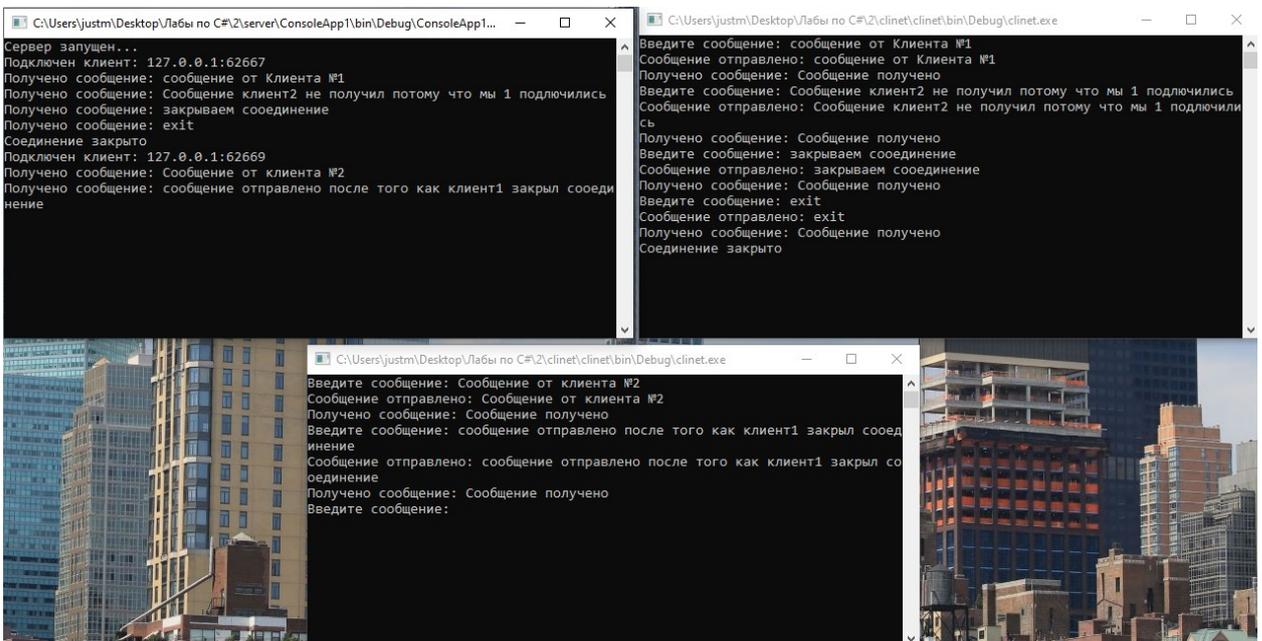


Рисунок 3

Исходный код клиента:

```

// Клиент
using System;
using System.Net.Sockets;
using System.Text;

class Program
{
    static void Main(string[] args)
    {
        byte[] bytes = new byte[1024];

        // Установка IP-адреса и порта сервера
    
```

```

string ipAddress = "127.0.0.1";
int port = 8080;

// Создание объекта TcpClient
TcpClient client = new TcpClient(ipAddress, port);

// Получение потока для обмена сообщениями с сервером
NetworkStream stream = client.GetStream();

string message = "";

while (message != "exit")
{
    Console.WriteLine("Введите сообщение: ");
    message = Console.ReadLine();

    // Отправка сообщения серверу
    byte[] msg = Encoding.UTF8.GetBytes(message);
    stream.Write(msg, 0, msg.Length);
    Console.WriteLine("Сообщение отправлено: {0}", message);

    // Получение сообщения от сервера
    int i = stream.Read(bytes, 0, bytes.Length);
    string data = Encoding.UTF8.GetString(bytes, 0, i);
    Console.WriteLine("Получено сообщение: {0}", data);
}
// Закрытие соединения с сервером
client.Close();
Console.WriteLine("Соединение закрыто");
Console.ReadLine();
}
}

```

Исходный код сервера:

```

using System;
using System.Net;
using System.Net.Sockets;
using System.Text;

class Program
{
    static void Main(string[] args)
    {
        byte[] bytes = new byte[1024];

        // Установка IP-адреса и порта для сервера
        IPAddress ipAddress = IPAddress.Parse("127.0.0.1");
        int port = 8080;

        // Создание объекта TcpListener
        TcpListener server = new TcpListener(ipAddress, port);

        // Начало прослушивания
        server.Start();
    }
}

```

```
Console.WriteLine("Сервер запущен...");

while (true)
{
    // Ожидание подключения клиента
    TcpClient client = server.AcceptTcpClient();
    Console.WriteLine("Подключен клиент: {0}",
client.Client.RemoteEndPoint);

    string data = "";
    string exitCode = "exit";

    while (data != exitCode)
    {
        // Получение сообщения от клиента
        NetworkStream stream = client.GetStream();
        int i = stream.Read(bytes, 0, bytes.Length);
        data = Encoding.UTF8.GetString(bytes, 0, i);
        Console.WriteLine("Получено сообщение: {0}", data);

        // Отправка сообщения клиенту
        byte[] msg = Encoding.UTF8.GetBytes("Сообщение получено");
        stream.Write(msg, 0, msg.Length);
    }

    // Закрытие соединения с клиентом
    client.Close();
    Console.WriteLine("Соединение закрыто");
}
}
```