Содержание

[3](#_Toc59656555)

[**Введение** 4](#_Toc59656556)

[**1.** **Анализ задачи** 5](#_Toc59656557)

[**2.** **Теоретическая часть** 6](#_Toc59656558)

[**3.** **Выбор средств реализации** 7](#_Toc59656559)

[**4.** **Проектирование приложения** 9](#_Toc59656560)

[4.1 Создание сервера 9](#_Toc59656561)

[**5.** **Реализация приложения** 13](#_Toc59656562)

[**6.** **Тестирование приложения** 19](#_Toc59656563)

[**Литература** 25](#_Toc59656564)

# 

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

Разраб.

Анисимов А.В.

Провер.

Бизюк А. Н.

Реценз.

Н. Контр.

Утверд.

Содержание

Лит.

Листов

УО «ВГТУ» каф. ИСИТ гр. Ит-10

# **Введение**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

Разраб.

Машковский Е.В.

Провер.

Бизюк А. Н.

Реценз.

Н. Контр.

Утверд.

Анализ задачи

Лит.

Листов

УО «ВГТУ» каф. ИСАП гр. Ит-6

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

Разраб.

Машковский Е.В.

Провер.

Бизюк А. Н.

Реценз.

Н. Контр.

Утверд.

Введение

Лит.

Листов

УО «ВГТУ» каф. ИСАП гр. Ит-6

# **Анализ задачи**

Целью данного курсового проекта является создание веб – приложения для управления файлами.

Веб – приложение должно будет реализовывать следующие функции:

1. Регистрация пользователя;
2. Авторизация пользователя;
3. Создание папок;
4. Удаление папок и файлов;
5. Загрузка файлов;
6. Скачивание файлов;
7. Сортировка файлов по имени, дате создания, типу;
8. Поиск файлов;

Авторизованному пользователю на сервере создается отдельная папка, в которой будут храниться загруженные пользователем файлы. С клиентской части приложения пользователю будет предоставлен интерфейс для реализации всех выше перечисленных функций веб – приложения.

# **Теоретическая часть**

Приложение, разрабатываемое в данном проекте будет общаться с сервером по протоколу TCP. Рассмотрим поподробнее данный протокол.

**Transmission Control Protocol** (TCP, протокол управления передачей) — один из основных [протоколов передачи данных](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) интернета, предназначенный для управления [передачей данных](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Пакеты в TCP называются сегментами**.**

В [стеке протоколов TCP/IP](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D0%BE%D0%B2_%D1%82%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D1%83%D1%80%D0%BE%D0%B2%D0%BD%D1%8F) выполняет функции [транспортного уровня](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%BD%D1%8B%D0%B9_%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C) [модели OSI](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI).

Механизм TCP предоставляет [поток данных](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%BE%D0%BA_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) с предварительной установкой соединения, осуществляет повторный запрос данных в случае потери данных и устраняет дублирование при получении двух копий одного пакета, гарантируя тем самым, в отличие от [UDP](https://ru.wikipedia.org/wiki/UDP), целостность передаваемых данных и уведомление отправителя о результатах передачи.

Реализации TCP обычно встроены в [ядра](https://ru.wikipedia.org/wiki/%D0%AF%D0%B4%D1%80%D0%BE_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B9_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B) [ОС](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0). Существуют реализации TCP, работающие в [пространстве пользователя](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F).

Когда осуществляется передача от компьютера к компьютеру через Интернет, TCP работает на верхнем уровне между двумя конечными системами, например, [браузером](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) и веб-сервером. TCP осуществляет надёжную передачу потока байтов от одного процесса к другому. TCP реализует управление потоком, управление перегрузкой, рукопожатие, надёжную передачу.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

Разраб.

Машковский Е.В.

Провер.

Бизюк А. Н.

Реценз.

Н. Контр.

Утверд.

Теоретическая часть

Лит.

Листов

УО «ВГТУ» каф. ИСАП гр. Ит-6

# **Выбор средств реализации**

Созданное приложение будет отправлять запросы на сервер и принимать соответствующие ответы от него. В проекте предусмотрено создание сервера.

Клиентское приложение будет разработано на языке высокого уровня C#. Язык программирования С# был разработан в 1993—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft.

К 2000 году у Microsoft были готовы промышленные версии новых технологий и решений для обмена сообщениями и данными, а также для создания Internet-приложений. Была выпущена и новая платформа для разработки под новые решения — .NET. В ней объединились сразу несколько языков программирования, что было в новинку для того времени.

Ещё одним новшеством платформы .NET была технология активных серверных страниц ASP.NET (Active Server Page). С её помощью можно было относительно быстро разработать веб-приложения, взаимодействующие с базами данных. Специально для ASP.NET был создан язык программирования C#. Да и сама ASP.NET была полностью написана на нём.

 В качестве среды разработки была выбрана программа Microsoft Visual Studio2019Community.Линейка программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать продуктов компании [Microsoft](https://ru.wikipedia.org/wiki/Microsoft), программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать включающих [интегрированную среду программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) как [консольные](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) [приложения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), так и игры и приложения с [графическим интерфейсом](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F), в том числе с поддержкой технологии [Windows Forms](https://ru.wikipedia.org/wiki/Windows_Forms), а также [веб-сайты](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82), [веб-приложения](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), [веб-службы](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0) как в [родном](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4), так и в [управляемом](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D0%BC%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) кодах для всех платформ,поддерживаемых [Windows](https://ru.wikipedia.org/wiki/Windows), [Windows Mobile](https://ru.wikipedia.org/wiki/Windows_Mobile), [Windows CE](https://ru.wikipedia.org/wiki/Windows_CE), [.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework), [Xbox](https://ru.wikipedia.org/wiki/Xbox), [Windows Phone](https://ru.wikipedia.org/wiki/Windows_Phone)[.NET Compact Framework](https://ru.wikipedia.org/wiki/.NET_Compact_Framework) и [Silverlight](https://ru.wikipedia.org/wiki/Silverlight).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

Разраб.

Машковский Е.В.

Провер.

Бизюк А. Н.

Реценз.

Н. Контр.

Утверд.

Выбор средств реализации

Лит.

Листов

УО «ВГТУ» каф. ИСАП гр. Ит-6

Visual Studio включает в себя [редактор исходного кода](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0) с поддержкой технологии [IntelliSense](https://ru.wikipedia.org/wiki/IntelliSense) и возможностью простейшего [рефакторинга кода](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3).

Встроенный [отладчик](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер [классов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) и дизайнер [схемы базы данных](https://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Visual Studio позволяет создавать и подключать сторонние дополнения ([плагины](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%B3%D0%B8%D0%BD)) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем [контроля версий исходного кода](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8) (как, например, [Subversion](https://ru.wikipedia.org/wiki/Subversion" \o "Subversion) и [Visual SourceSafe](https://ru.wikipedia.org/wiki/Microsoft_Visual_SourceSafe" \o "Microsoft Visual SourceSafe)), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на [предметно-ориентированных языках программирования](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA)) или инструментов для прочих аспектов [процесса разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

# **Проектирование приложения**

Работа программы основана на отправке запросов на сервер при определённых действиях пользователя. Программа исполняет следующие функции:

1. Создание сервера;
2. Принятие клиентов;
3. Просмотр сообщений, присланных на сервер;
4. Подключение пользователей;
5. Ввод адреса и порта сервера;
6. Отправка сообщений на сервер;

## Создание сервера

При запуске программы будет создаваться сервер.

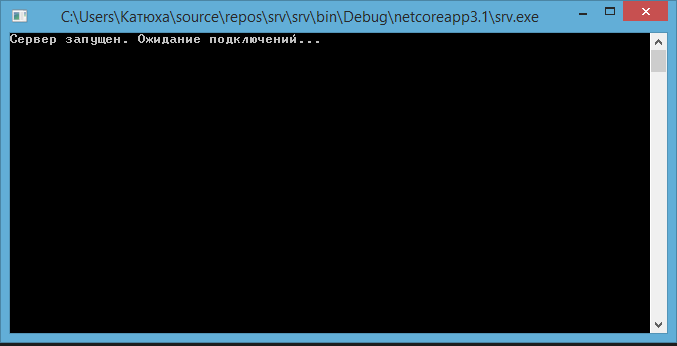


Рисунок 1. Пример сервера

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

Разраб.

Машковский Е.В.

Провер.

Бизюк А. Н.

Реценз.

Н. Контр.

Утверд.

Проектирование приложения

Лит.

Листов

УО «ВГТУ» каф. ИСАП гр. Ит-6

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

4.2 Принятие пользователей

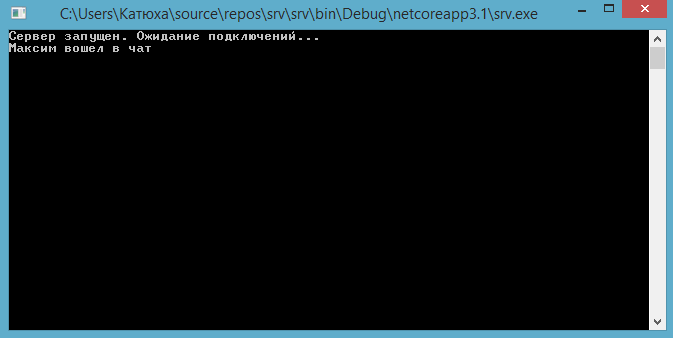


Рисунок 2. Пример принятия пользователя

4.3 Принятие сообщения от пользователя

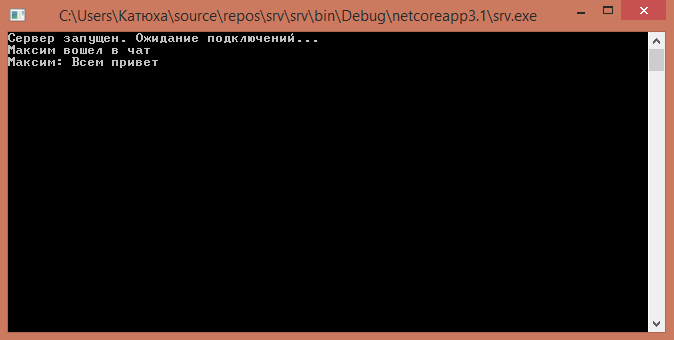


Рисунок 3. Пример принятия сообщения

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

4.4 Выход пользователя

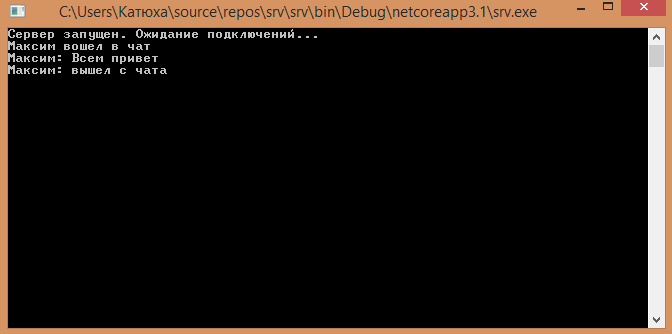


Рисунок 4.Пример выхода пользователя

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

4.5 Подключение пользователя

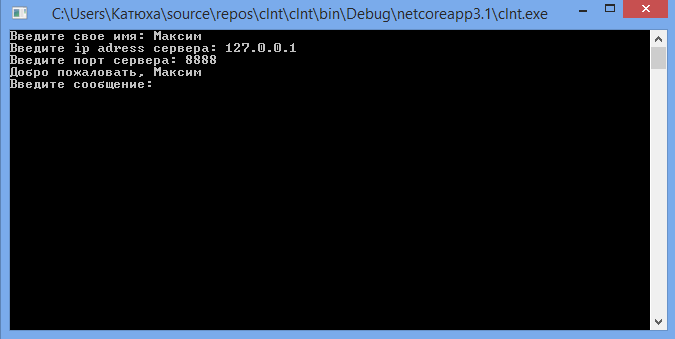


Рисунок 5. Пример подключения пользователя

4.6 Отправка сообщений

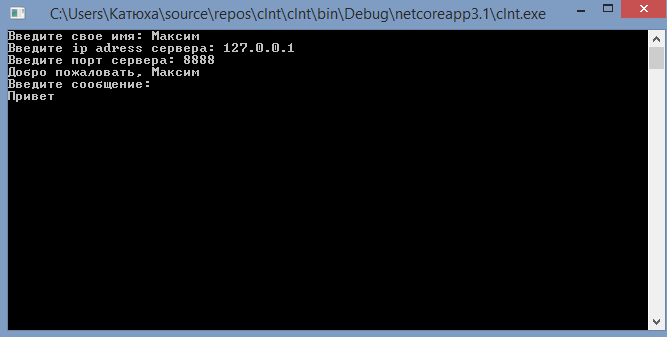


Рисунок 6. Пример отправки сообщения

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

4.7 Принятие сообщение от других пользователей

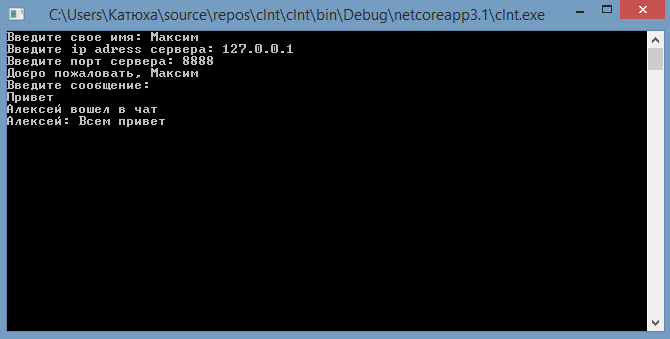


Рисунок 7. Пример принятия сообщения

# **Реализация приложения**

Приложение, разработанное в данном проекте может быть использовано любым пользователем для общения c любым другим пользователем через сервер.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

Разраб.

Машковский Е.В.

Провер.

Бизюк А. Н.

Реценз.

Н. Контр.

Утверд.

Реализация приложения

Лит.

Листов

УО «ВГТУ» каф. ИСАП гр. Ит-6

Приложение используется для отправки писем другим пользователям, которые находятся на данном сервере, а также для чтения присланных другими пользователями письма.

Код программы:

1.Создание сервера предоставлено в следующем коде:

protected internal void Listen()

{

try

{

tcpListener = new TcpListener(IPAddress.Any, 8888);

tcpListener.Start();

Console.WriteLine("Сервер запущен. Ожидание подключений...");

while (true)

{

TcpClient tcpClient = tcpListener.AcceptTcpClient();

ClientObject clientObject = new ClientObject(tcpClient, this);

Thread clientThread = new Thread(new ThreadStart(clientObject.Process));

clientThread.Start();

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

Disconnect();

}

}

Здесь происходит запуск сервера при помощи команды tcpListener.Start();

После этого с помощью команды Console.WriteLine("Сервер запущен. Ожидание подключений..."); Выводим сообщение о том что сервер запущен.

2.Принятие пользователя реализовано в следующем участке кода:

public void Process()

{

try

{

Stream = client.GetStream();

string message = GetMessage();

userName = message;

message = userName + " вошел в чат";

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

server.BroadcastMessage(message, this.Id);

Console.WriteLine(message);

while (true)

{

try

{

message = GetMessage();

message = String.Format("{0}: {1}", userName, message);

Console.WriteLine(message);

server.BroadcastMessage(message, this.Id);

}

catch

{

message = String.Format("{0}: вышел с чата", userName);

Console.WriteLine(message);

server.BroadcastMessage(message, this.Id);

break;

}

}

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

finally

{

server.RemoveConnection(this.Id);

Close();

}

}

public class ServerObject

{

static TcpListener tcpListener;

List<ClientObject> clients = new List<ClientObject>();

protected internal void AddConnection(ClientObject clientObject)

{

clients.Add(clientObject);

}

protected internal void RemoveConnection(string id)

{

ClientObject client = clients.FirstOrDefault(c => c.Id == id);

if (client != null)

clients.Remove(client);

Здесь с помощью запроса clients.Add(clientObject); идет присоединение пользователя и с помощью запроса

Stream = client.GetStream();

string message = GetMessage();

userName = message;

message = userName + " вошел в чат";  
Выводим сообщение с именем пользователя ,который присоединился.

3.Принятие сообщений .Данный метод реализован в следующем участке кода:

private string GetMessage()

{

byte[] data = new byte[64];

StringBuilder builder = new StringBuilder();

int bytes = 0;

do

{

bytes = Stream.Read(data, 0, data.Length);

builder.Append(Encoding.Unicode.GetString(data, 0, bytes));

}

while (Stream.DataAvailable);

return builder.ToString();

}

protected internal void Close()

{

if (Stream != null)

Stream.Close();

if (client != null)

client.Close();

}

}

4.Выход пользователя реализован в коде:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

protected internal void Disconnect()

{

tcpListener.Stop();

for (int i = 0; i < clients.Count; i++)

{

clients[i].Close();

}

Environment.Exit(0);

}

}

5.Подключение пользователя. Данный метод реализуется в приложении клиент следующим кодом:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

static void Main(string[] args)

{

Console.Write("Введите свое имя: ");

userName = Console.ReadLine();

Console.Write("Введите ip adress сервера: ");

host = Console.ReadLine();

Console.Write("Введите порт сервера: ");

port = Int32.Parse(Console.ReadLine());

client = new TcpClient();

try

{

client.Connect(host, port);

stream = client.GetStream();

string message = userName;

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

stream.Write(data, 0, data.Length);

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

receiveThread.Start();

Console.WriteLine("Добро пожаловать, {0}", userName);

SendMessage();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

finally

{

Disconnect();

}

}

Здесь пользователь вводит свои данные, ip address и порт сервера с помощью команд: Console.Write("Введите свое имя: ");

userName = Console.ReadLine();

Console.Write("Введите ip adress сервера: ");

host = Console.ReadLine();

Console.Write("Введите порт сервера: ");

port = Int32.Parse(Console.ReadLine());

Далее идёт подключение client.Connect(host, port);

6.Отправка сообщений реализована в этом коде:

static void SendMessage()

{

Console.WriteLine("Введите сообщение: ");

while (true)

{

string message = Console.ReadLine();

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

stream.Write(data, 0, data.Length);

}

}

7.Принятие сообщение реализована в коде сервера и клиента:

public void Process()

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

{

try

{

Stream = client.GetStream();

string message = GetMessage();

userName = message;

message = userName + " вошел в чат";

server.BroadcastMessage(message, this.Id);

Console.WriteLine(message);

while (true)

{

try

{

message = GetMessage();

message = String.Format("{0}: {1}", userName, message);

Console.WriteLine(message);

server.BroadcastMessage(message, this.Id);

}

catch

{

message = String.Format("{0}: вышел с чата", userName);

Console.WriteLine(message);

server.BroadcastMessage(message, this.Id);

break;

}

}

}

Здесь сервер обрабатывает сообщения пользователя и отсылает их другим пользователям:

try

{

message = GetMessage();

message = String.Format("{0}: {1}", userName, message);

Console.WriteLine(message);

server.BroadcastMessage(message, this.Id);

}

static void ReceiveMessage()

{

while (true)

{

try

{

byte[] data = new byte[64];

StringBuilder builder = new StringBuilder();

int bytes = 0;

do

{

bytes = stream.Read(data, 0, data.Length);

builder.Append(Encoding.Unicode.GetString(data, 0, bytes));

}

while (stream.DataAvailable);

string message = builder.ToString();

Console.WriteLine(message);

}

catch

{

Console.WriteLine("Подключение прервано!");

Console.ReadLine();

Disconnect();

}

}

}

Здесь клиент принимает и выводит сообщение:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

while (stream.DataAvailable);

string message = builder.ToString();

Console.WriteLine(message);

# **Тестирование приложения**

В данном примере будет запущено клиентское и серверное приложение, будет показано: запуск сервера,вход клиентов на сервер,принятие клиентов,отправка сообщений клиентами ,принятие их на сервер и отправка другим клиентам.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

Разраб.

Машковский Е.В.

Провер.

Бизюк А. Н.

Реценз.

Н. Контр.

Утверд.

Тестирование приложения

Лит.

Листов

УО «ВГТУ» каф. ИСАП гр. Ит-6

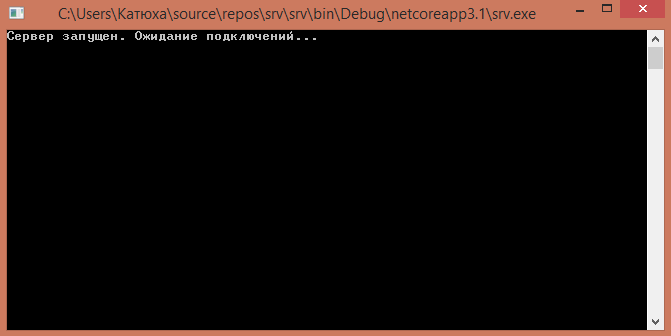


Рисунок 8. Запуск сервера

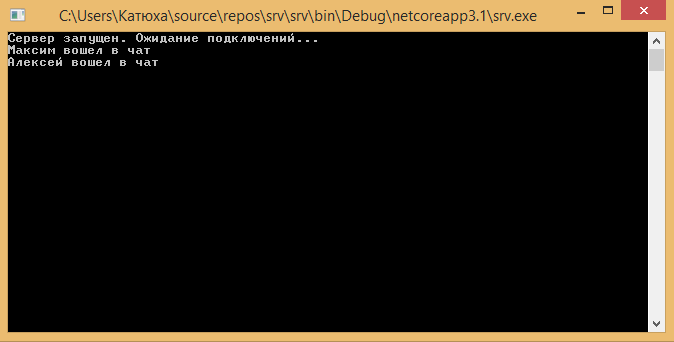
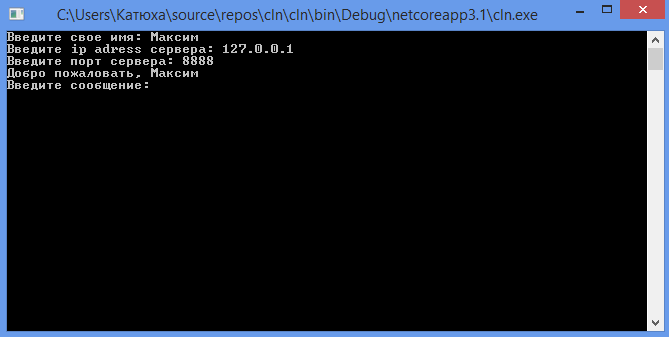


Рисунок 9.Вход пользователей



Изм.

Лист

№ докум.

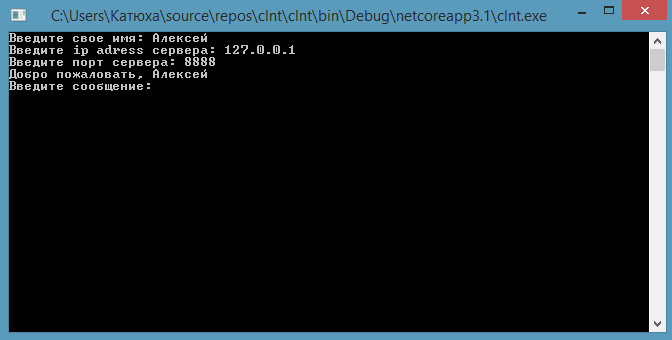
Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

Рисунок 10. Вход пользователя Максима



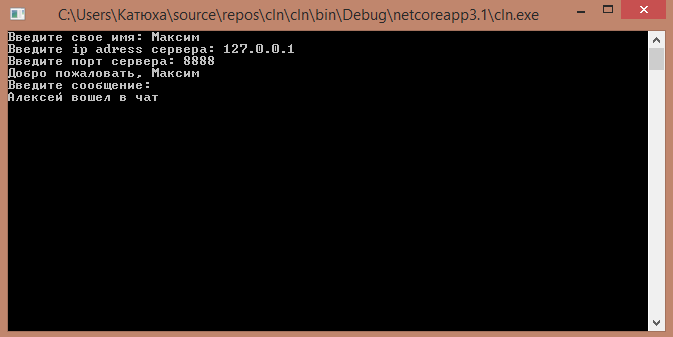
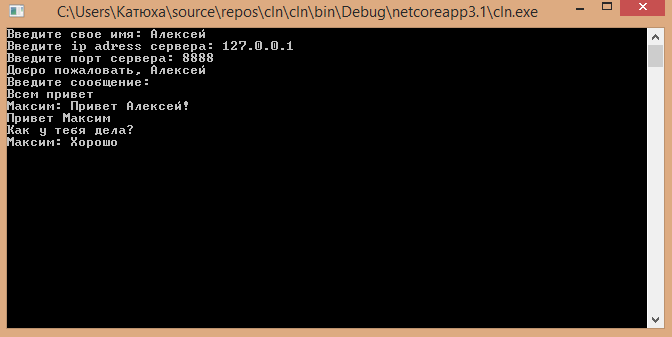


Рисунок 11-12. Вход пользователя Алексея



Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

Рисунок 13. Отправка сообщений пользователем Алексей

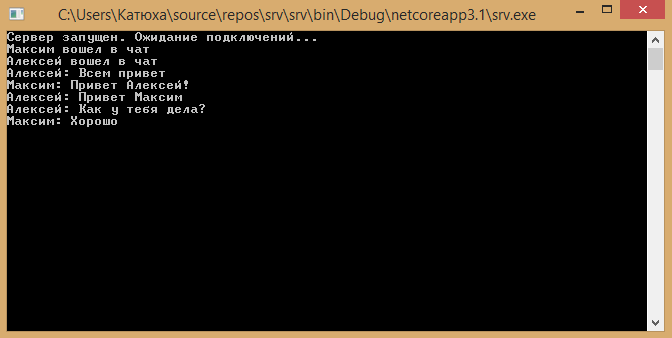
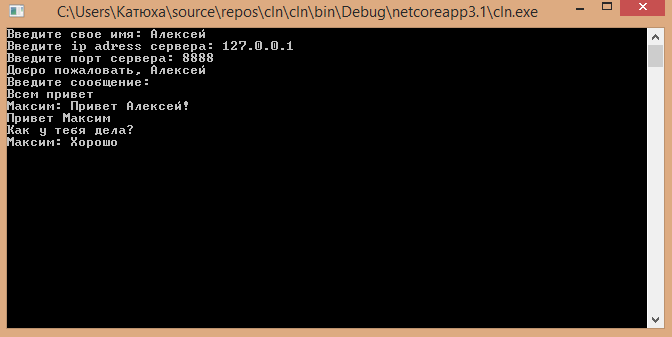


Рисунок 14. Принятие сообщений Сервером



Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

Рисунок 15. Принятие сообщений пользователем Максим

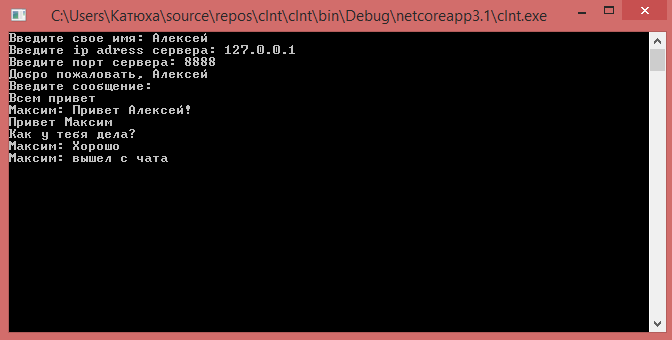


Рисунок 16. Отображение выхода пользователя Максим у Алексея

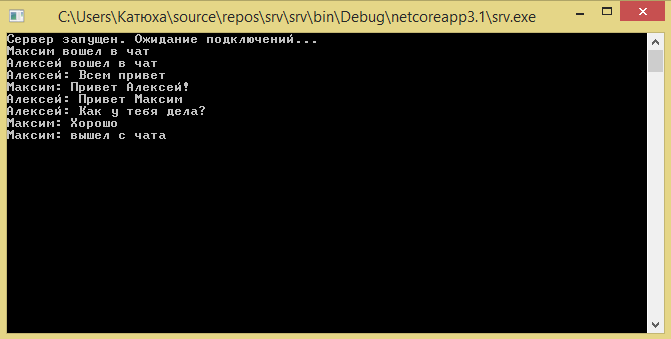


Рисунок 17. Отображение выхода пользователя Максим на сервере

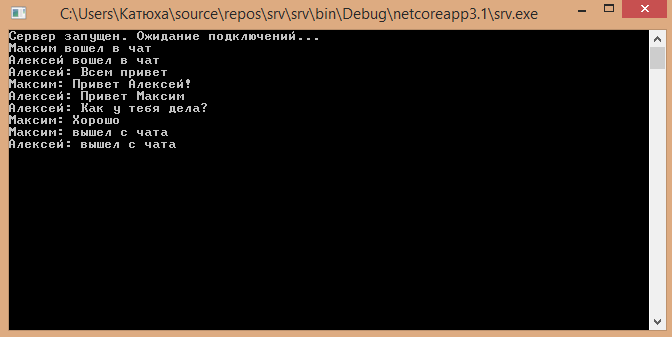


Рисунок 18. Отображение выхода пользователей на сервере

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

**Заключение**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

Разраб.

Машковский Е.В.

Провер.

Бизюк А. Н.

Реценз.

Н. Контр.

Утверд.

Заключение

Лит.

Листов

УО «ВГТУ» каф. ИСАП гр. Ит-6

В данной курсовой работе было создано клиентское и серверное приложение, отправляющее сообщения на сервер и другим клиентам. В приложении были реализованы следующие функции:

1. Создание сервера;
2. Принятие клиентов;
3. Просмотр сообщений, присланных на сервер;
4. Подключение пользователей;
5. Ввод адреса и порта сервера;
6. Отправка сообщений на сервер;
7. Принятие сообщение от других пользователей;

Можно сделать вывод что все задачи, поставленные во введении к курсовой работе, были выполнены, следовательно, курсовую работу можно считать успешно завершённой.

# **Литература**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» KР.007 1-53 01 01-05 ПЗ

Разраб.

Машковский Е.В.

Провер.

Бизюк А. Н.

Реценз.

Н. Контр.

Утверд.

Литература

Лит.

Листов

УО «ВГТУ» каф. ИСАП гр. Ит-6

1. TCP – что это? [Электронный ресурс].URL: <https://ru.wikipedia.org/wiki/Transmission_Control_Protocol> (дата обращения: 08.12.2020);
2. Язык программирования C#.

[Электронный ресурс]. URL: https://geekbrains.ru/posts/yazyk-programmirovaniya-c-sharp-istoriya-specifika-mesto-na-rynke (дата обращения: 10.12.2020);

1. Описание продукта Microsoft Visual Studio 2019 Community. [Электронный ресурс].URL: https://m.moreigr.com/soft/9488-microsoft-visual-studio-2019.html (дата обращения: 10.12.2020);