БПОУ РА «Горно-Алтайский государственный политехнический колледж им. М.З. Гнездилова»

**ОТЧЕТ**

Выполнил\_\_ студент\_\_\_\_ \_\_\_3\_\_ курса \_\_\_\_\_1212 группы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Максимов Никита\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Выполнен с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024г.

Горно-Алтайск

2024

Задание 1

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

if (!string.IsNullOrWhiteSpace(txtName.Text) && !lstNames.Items.Contains(txtName.Text))

{

lstNames.Items.Add(txtName.Text);

}

}

void Form1\_KeyPress(object sender, KeyPressEventArgs e)

{

MessageBox.Show($"Form.KeyPress: '{e.KeyChar}' pressed.");

switch(e.KeyChar)

{

case (char)49:

case (char)52:

case (char)55:

MessageBox.Show($"Form.KeyPress: '{e.KeyChar}' consumed.");

e.Handled = true;

break;

}

}

private void button1\_MouseEnter(object sender, EventArgs e) =>

Cursor.Hide();

private void button1\_MouseLeave(object sender, EventArgs e) =>

Cursor.Show();

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

Cursor.Position = PointToScreen(btnAdd.Location);

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

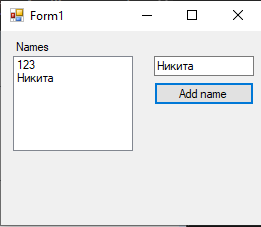
Cursor.Position = PointToScreen(btnAdd.Location);

private void Form1\_Load(object sender, EventArgs e)

{

}

}



Задание 2 и 2.2

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

this.Close();

}

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

{

int Summa = Int32.Parse(txtA.Text) +

Int32.Parse(txtB.Text) + Int32.Parse(txtC.Text);

txtSumma.Text = Summa.ToString();

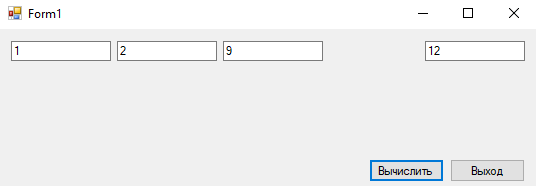
}

private void Form1\_Load(object sender, EventArgs e)

{

}

}



public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

this.txt.BackColor = System.Drawing.Color.Black;

}

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

{

this.txt.BackColor = System.Drawing.Color.Red;

}

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

{

this.txt.BackColor = System.Drawing.Color.Blue;

}

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

{

this.txt.BackColor = System.Drawing.Color.Green;

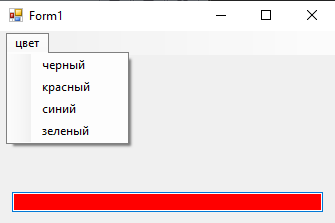
}

private void Form1\_Load(object sender, EventArgs e)

{

}

}



Задание 3

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

if(openFileDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.Load(openFileDialog1.FileName);

}

}

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

{

pictureBox1.Image = null;

}

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

{

if(colorDialog1.ShowDialog() == DialogResult.OK)

{

pictureBox1.BackColor = colorDialog1.Color;

}

}

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

{

this.Close();

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox1.Checked)

{

pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;

}

else

{

pictureBox1.SizeMode = PictureBoxSizeMode.Normal;

}

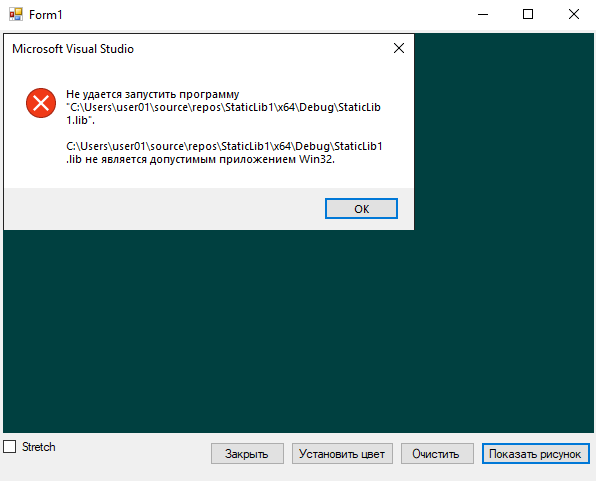
}

private void Form1\_Load(object sender, EventArgs e)

{

}

}



Задание 4

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

file1.Filter = "(\*.JPG)|\*.JPG";

}

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

{

}

private void saveFileDialog1\_FileOk(object sender, CancelEventArgs e)

{

string fname;

file1.ShowDialog();

fname= file1.FileName;

pct.Image = Image.FromFile(fname);

}

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

{

string fname;

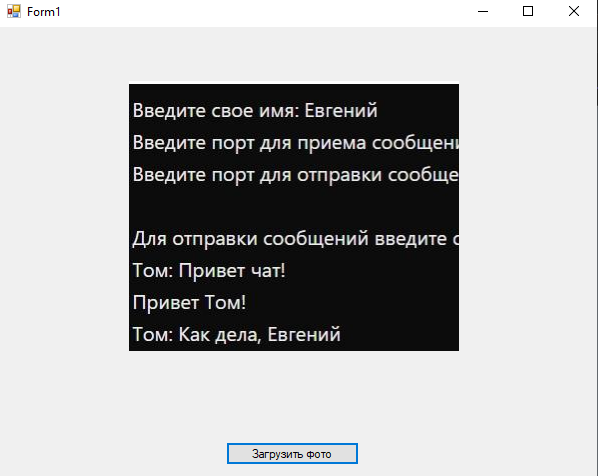
file1.ShowDialog();

fname = file1.FileName;

pct.Image = Image.FromFile(fname);

}

}



Задание 5

partial class Form1

{

private System.ComponentModel.IContainer components = null;

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

private void InitializeComponent()

{

pictureBox1 = new PictureBox();

openButton = new Button();

saveButton = new Button();

((System.ComponentModel.ISupportInitialize)pictureBox1).BeginInit();

SuspendLayout();

// pictureBox1

pictureBox1.Location = new Point(192, 35);

pictureBox1.Name = "pictureBox1";

pictureBox1.Size = new Size(388, 235);

pictureBox1.TabIndex = 0;

pictureBox1.TabStop = false;

pictureBox1.Click += pictureBox1\_Click;

// openButton

openButton.Location = new Point(192, 288);

openButton.Name = "openButton";

openButton.Size = new Size(75, 23);

openButton.TabIndex = 1;

openButton.Text = "Открыть";

openButton.UseVisualStyleBackColor = true;

openButton.Click += openButton\_Click\_1;

// saveButton

saveButton.Location = new Point(505, 288);

saveButton.Name = "saveButton";

saveButton.Size = new Size(75, 23);

saveButton.TabIndex = 2;

saveButton.Text = "Сохранить";

saveButton.UseVisualStyleBackColor = true;

saveButton.Click += saveButton\_Click\_1;

// Form1

AutoScaleDimensions = new SizeF(7F, 15F);

AutoScaleMode = AutoScaleMode.Font;

ClientSize = new Size(800, 450);

Controls.Add(saveButton);

Controls.Add(openButton);

Controls.Add(pictureBox1);

Name = "Form1";

Text = "Form1";

Load += Form1\_Load;

MouseDown += pictureBox1\_MouseDown;

MouseMove += pictureBox1\_MouseMove;

MouseUp += pictureBox1\_MouseUp;

((System.ComponentModel.ISupportInitialize)pictureBox1).EndInit();

ResumeLayout(false);

}

private void Form1\_Load(object sender, EventArgs e)

{

}

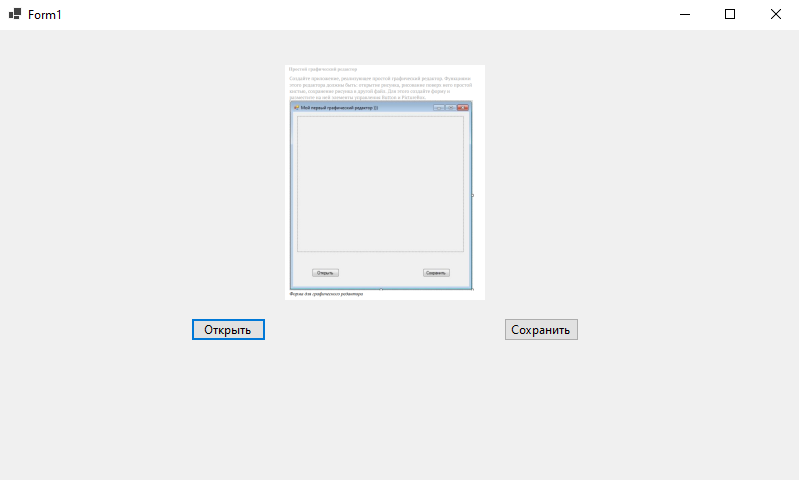
#endregion

private PictureBox pictureBox1;

private Button openButton;

private Button saveButton;

}



Задание 6

using System;

using System.IO;

using System.Net;

using System.Net.Sockets;

using System.Threading;

using System.Text;

using System.Diagnostics;

using System.Security.AccessControl;

public class Program

{

private static readonly string DirectoryPath = "H:\\";

private const int Port = 8000;

public static void Main(string[] args)

{

LoadHtmlAndDisplayRequests();

Console.WriteLine("Нажмите любую клавишу для завершения...");

Console.ReadKey();

}

static void LoadHtmlAndDisplayRequests()

{

string filePath = Path.Combine(DirectoryPath, "index.html");

string responseText = "";

try

{

responseText = File.ReadAllText(filePath, Encoding.UTF8);

}

catch (FileNotFoundException)

{

Console.WriteLine($"Ошибка: Файл {filePath} не найден.");

return;

}

Console.WriteLine("Содержимое responseText:");

Console.WriteLine(responseText);

Console.WriteLine(new string('-', 50));

Console.WriteLine($"Открываем браузер (скопируйте в адресную строку): http://localhost:{Port}/index.html");

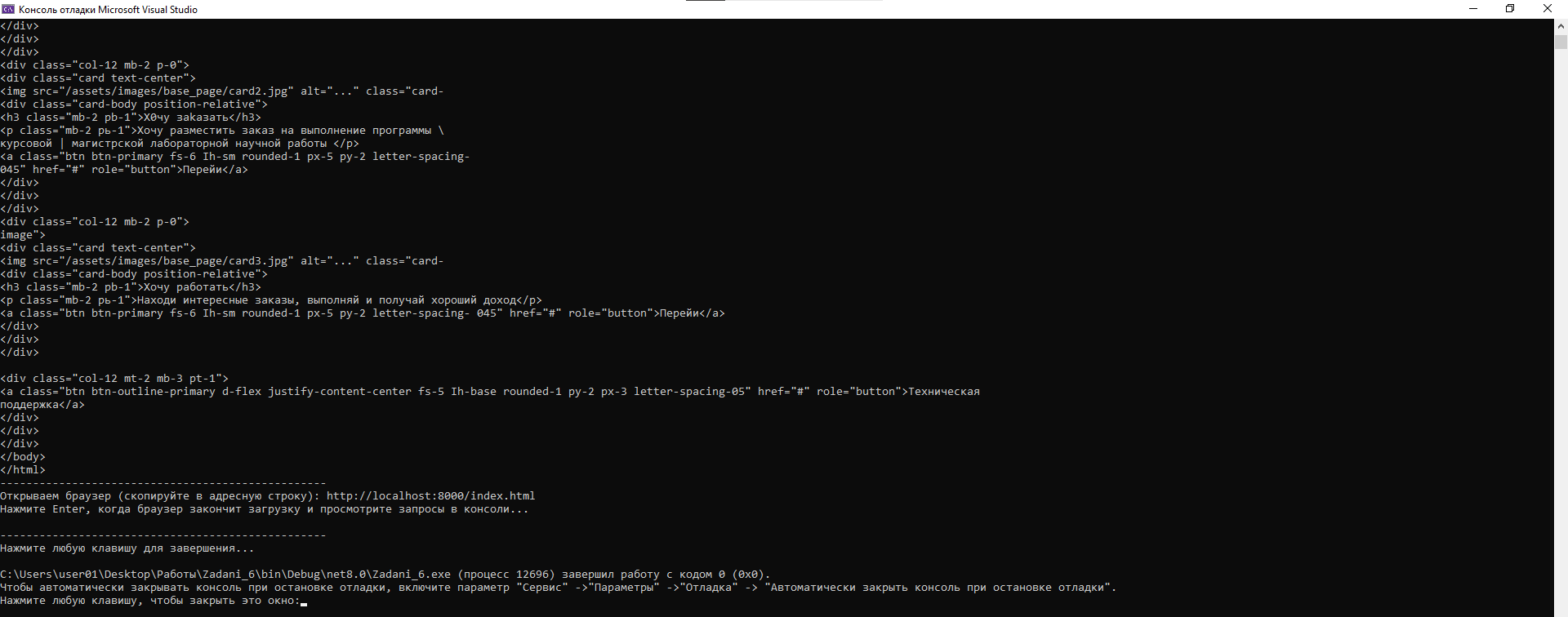
Console.WriteLine("Нажмите Enter, когда браузер закончит загрузку и просмотрите запросы в консоли...");

Console.ReadLine();

Console.WriteLine(new string('-', 50));

}

}



Задание 7

ClientChat

using System;

using System.Net;

using System.Net.Sockets;

using System.Text;

using System.Threading;

public class Client

{

private static TcpClient client;

private static NetworkStream stream;

private static string userName;

public static void Main(string[] args)

{

try

{

Console.Write("Введите свое имя: ");

userName = Console.ReadLine();

client = new TcpClient();

client.Connect(IPAddress.Loopback, 8888);

stream = client.GetStream();

byte[] nameData = Encoding.UTF8.GetBytes(userName);

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

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

Console.WriteLine("Для отправки сообщений введите сообщение и нажмите Enter");

Thread receiveThread = new Thread(ReceiveMessages);

receiveThread.Start();

while (true)

{

string message = Console.ReadLine();

byte[] data = Encoding.UTF8.GetBytes(message);

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

}

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка клиента: {ex.Message}");

}

finally

{

stream?.Close();

client?.Close();

}

}

static void ReceiveMessages()

{

try

{

while (true)

{

byte[] data = new byte[256];

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

if (bytes == 0)

break;

string message = Encoding.UTF8.GetString(data, 0, bytes);

Console.WriteLine(message);

}

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка при чтении сообщений с сервера: {ex.Message}");

}

}

}

ServerChat

using System;

using System.Net;

using System.Net.Sockets;

using System.Text;

using System.Collections.Generic;

using System.Threading;

public class Server

{

private static List<TcpClient> clients = new List<TcpClient>();

private static Dictionary<TcpClient, string> clientNames = new Dictionary<TcpClient, string>();

private static TcpListener listener;

public static void Main(string[] args)

{

try

{

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

listener.Start();

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

while (true)

{

TcpClient client = listener.AcceptTcpClient();

clients.Add(client);

Thread clientThread = new Thread(HandleClient);

clientThread.Start(client);

}

}

catch (Exception e)

{

Console.WriteLine($"Ошибка сервера: {e.Message}");

}

finally

{

listener?.Stop();

}

}

static void HandleClient(object obj)

{

TcpClient client = (TcpClient)obj;

try

{

NetworkStream stream = client.GetStream();

byte[] nameData = new byte[256];

int bytes = stream.Read(nameData, 0, nameData.Length);

string name = Encoding.UTF8.GetString(nameData, 0, bytes);

clientNames.Add(client, name);

Console.WriteLine($"{name} вошел в чат");

BroadcastMessage($"{name} вошел в чат", null);

while (true)

{

byte[] data = new byte[256];

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

if (bytes == 0)

{

break;

}

string message = Encoding.UTF8.GetString(data, 0, bytes);

Console.WriteLine($"{name}: {message}");

BroadcastMessage($"{name}: {message}", client);

}

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка при работе с клиентом: {ex.Message}");

}

finally

{

string leftName = "";

if (clientNames.ContainsKey(client))

{

leftName = clientNames[client];

}

Console.WriteLine($"{leftName} покинул чат");

clientNames.Remove(client);

clients.Remove(client);

BroadcastMessage($"{leftName} покинул чат", null);

client?.Close();

}

}

static void BroadcastMessage(string message, TcpClient excludeClient)

{

byte[] data = Encoding.UTF8.GetBytes(message);

foreach (TcpClient client in clients)

{

if (client != excludeClient && client.Connected)

{

try

{

NetworkStream stream = client.GetStream();

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

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка при отправке сообщения клиенту: {ex.Message}");

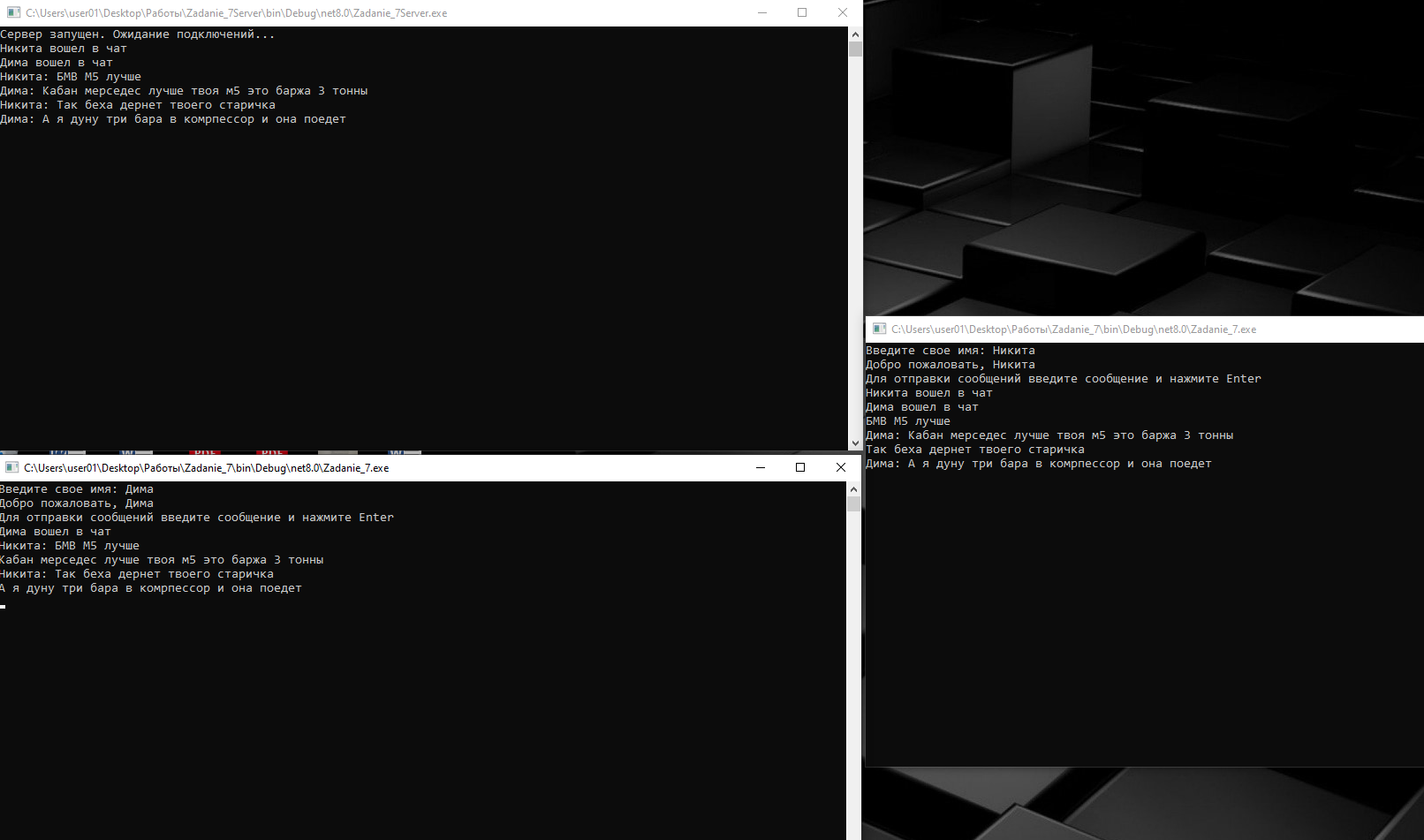
}

}

}

}

}



Задание 8

using System;

using System.Net;

using System.Net.Sockets;

using System.Text;

using System.Threading.Tasks;

public class UdpChat

{

private static string userName;

private static int receivePort;

private static int sendPort;

private static UdpClient receiveClient;

private static UdpClient sendClient;

private static bool \_exit = false;

public static async Task Main(string[] args)

{

Console.Write("Введите свое имя: ");

userName = Console.ReadLine();

Console.Write("Введите порт для приема сообщений: ");

receivePort = int.Parse(Console.ReadLine());

Console.Write("Введите порт для отправки сообщений: ");

sendPort = int.Parse(Console.ReadLine());

try

{

receiveClient = new UdpClient(receivePort);

\_ = ReceiveMessages();

sendClient = new UdpClient();

sendClient.EnableBroadcast = true;

Console.WriteLine("Для отправки сообщений введите сообщение и нажмите Enter");

while (!\_exit)

{

string message = Console.ReadLine();

if (string.IsNullOrEmpty(message)) continue;

SendMessage(message);

}

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка при работе с UDP: {ex.Message}");

}

finally

{

if (receiveClient != null)

{

receiveClient.Close();

receiveClient = null;

}

if (sendClient != null)

{

sendClient.Close();

sendClient = null;

}

}

}

static void SendMessage(string message)

{

string formattedMessage = $"{userName}: {message}";

byte[] data = Encoding.UTF8.GetBytes(formattedMessage);

try

{

IPEndPoint endPoint = new IPEndPoint(IPAddress.Broadcast, sendPort);

sendClient.Send(data, data.Length, endPoint);

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка при отправке сообщения: {ex.Message}");

}

}

static async Task ReceiveMessages()

{

try

{

while (!\_exit)

{

UdpReceiveResult result = await receiveClient.ReceiveAsync();

byte[] data = result.Buffer;

string message = Encoding.UTF8.GetString(data);

Console.WriteLine(message);

}

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка при получении сообщения: {ex.Message}");

}

}

}

