**Федеральное государственное образовательное бюджетное учреждение  высшего образования**

**«Финансовый университет при Правительстве РФ»**

КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

Лабораторная работа № 8

по дисциплине инфокоммуникационные системы и сети

Выполнил студент

Гаджалиев И. А.

Группа 3ПКС-420

Вариант:    \_\_\_\_

Проверил

Сибирев И. В

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

оценка подпись

Москва 2023

Цель: разобраться и переписать консольное HTTP приложение под WPF

XAML

<Window x:Class="WpfApp1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:WpfApp1"

mc:Ignorable="d"

Title="Gadzhaliev\_420 smski" Height="500" Width="800" Background="LightGreen" ResizeMode="CanMinimize">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="30"/>

<ColumnDefinition Width="765"/>

</Grid.ColumnDefinitions>

<TextBlock Text="Собеседник" TextWrapping="Wrap" Margin="409,21,91,349" Foreground="White" FontSize="40" FontWeight="Bold" TextAlignment="Center" FontFamily="Times New Roman" Grid.Column="1" RenderTransformOrigin="0.5,0.5"></TextBlock>

<TextBlock Text="Вы" TextWrapping="Wrap" Margin="106,21,543,349" Foreground="White" FontSize="40" FontWeight="Bold" TextAlignment="Center" FontFamily="Times New Roman" Grid.Column="1"/>

<TextBox VerticalContentAlignment="Center" Name="SendMessage" Text="" TextWrapping="Wrap" Margin="24,377,495,29" Background="White" Foreground="Black" FontFamily="Times New Roman" FontSize="18" Grid.Column="1"></TextBox>

<ListBox Name="Messages" Margin="24,79,400,123" Background="White" Foreground="Black" FontFamily="Times New Roman" FontSize="16" Grid.Column="1"></ListBox>

<Button Content=">" Margin="276,377,400,29" Click="Send\_Message" VerticalContentAlignment="Center" Background="White" Foreground="Black" FontSize="45" FontWeight="Bold" Cursor="Hand" Grid.Column="1" FontFamily="Bahnschrift"></Button>

<ListBox x:Name="Messages\_Copy" Margin="371,79,53,123" Background="White" Foreground="Black" FontFamily="Times New Roman" FontSize="16" Grid.Column="1"></ListBox>

<Button Content=">" Margin="623,377,53,29" Click="Send\_Message\_Copy" VerticalContentAlignment="Center" Background="White" Foreground="Black" FontSize="45" FontWeight="Bold" Cursor="Hand" Grid.Column="1" FontFamily="Bahnschrift"></Button>

<TextBox VerticalContentAlignment="Center" x:Name="SendMessage\_Copy" Text="" TextWrapping="Wrap" Margin="371,377,148,29" Background="White" Foreground="Black" FontFamily="Times New Roman" FontSize="18" Grid.Column="1"></TextBox>

</Grid>

</Window>

xaml.cs

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Net;

using System.Net.Http;

using System.IO;

namespace Лабораторная\_8

{

public static class Ext\_HttpListener

{

public static System.Net.HttpListener Set\_Prefixes\_Add(this System.Net.HttpListener \_this, string \_str = "http://127.0.0.1:8888/connection/", System.Boolean \_IsOpen = false)

{

\_this.Prefixes.Add(\_str);

if (\_IsOpen) System.Diagnostics.Process.Start(\_str);

return \_this;

}

public static System.Net.HttpListener Set\_Start(this System.Net.HttpListener \_this)

{ \_this.Start(); return \_this; }

public static System.Net.HttpListener Get\_ContextAsync(this System.Net.HttpListener \_this, System.Action<HttpListenerContext> A)

{ A(\_this.GetContextAsync().GetAwaiter().GetResult()); return \_this; }

public static System.Net.HttpListener Get\_ContextAsync\_WhileTrue(this System.Net.HttpListener \_this, System.Action<HttpListenerContext> A)

{ while (true) \_this.Get\_ContextAsync(A); }

}

public static class Ext\_String

{

public static byte[] Get\_Encoding\_UTF8\_Bytes(this System.String \_this) { return Encoding.UTF8.GetBytes(\_this); }

}

public static class Ext\_HttpListenerResponse

{

public static System.Net.HttpListenerResponse Set\_Bytes(this System.Net.HttpListenerResponse \_this, byte[] \_Bytes)

{

\_this.ContentLength64 = \_Bytes.Length;

Stream output = \_this.OutputStream;

output.WriteAsync(\_Bytes, 0, \_Bytes.Length);

output.FlushAsync();

return \_this;

}

}

public partial class MainWindow : Window

{

string msg1 = "";

string msg2 = "";

public MainWindow()

{

InitializeComponent();

System.Threading.Tasks.Task \_TaskA = new Task(() =>

new HttpListener()

.Set\_Prefixes\_Add("http://127.0.0.1:8881/connection/", \_IsOpen: false)

.Set\_Start()

.Get\_ContextAsync\_WhileTrue(a => {

System.String \_strResponse = msg1;

msg1 = "";

Dispatcher.InvokeAsync(() => this.Messages\_Copy.Items.Add(\_strResponse));

a.Response.Set\_Bytes(\_strResponse.Get\_Encoding\_UTF8\_Bytes());

})

);

\_TaskA.Start();

System.Threading.Tasks.Task \_TaskB = new Task(() =>

new HttpListener()

.Set\_Prefixes\_Add("http://127.0.0.2:8882/connection/", \_IsOpen: false)

.Set\_Start()

.Get\_ContextAsync\_WhileTrue(a => {

System.String \_strResponse = msg2;

msg2 = "";

Dispatcher.InvokeAsync(() => this.Messages.Items.Add(\_strResponse));

a.Response.Set\_Bytes(\_strResponse.Get\_Encoding\_UTF8\_Bytes());

})

);

\_TaskB.Start();

}

private void Send\_Message(object sender, RoutedEventArgs e)

{

msg1 = SendMessage.Text.Trim();

SendMessage.Clear();

if (msg1.Length == 0)

return;

new HttpClient().GetStringAsync("http://127.0.0.1:8881/connection/").GetAwaiter().GetResult();

}

private void Send\_Message\_Copy(object sender, RoutedEventArgs e)

{

msg2 = SendMessage\_Copy.Text.Trim();

SendMessage\_Copy.Clear();

if (msg2.Length == 0)

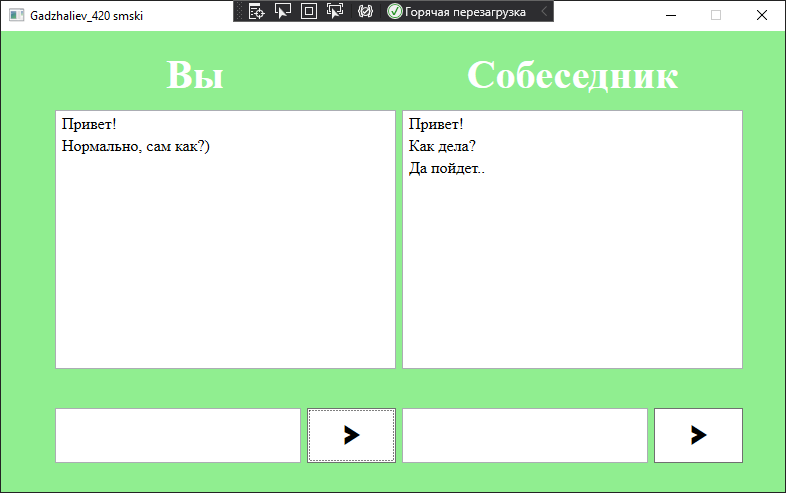
return;

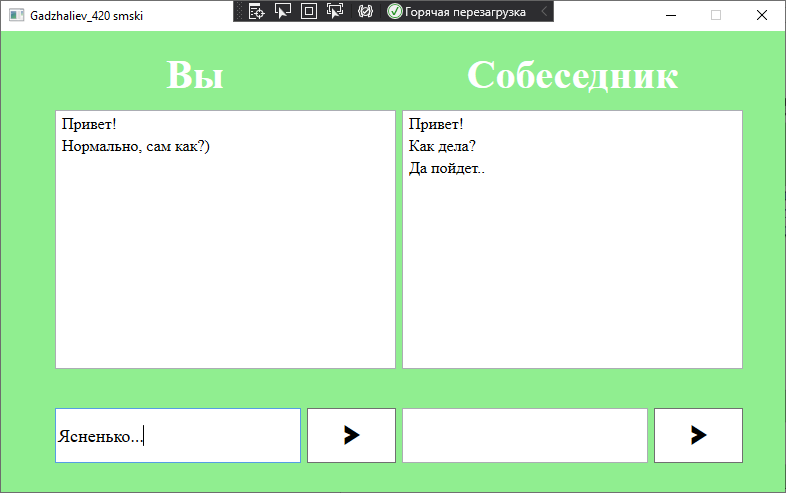
new HttpClient().GetStringAsync("http://127.0.0.2:8882/connection/").GetAwaiter().GetResult();

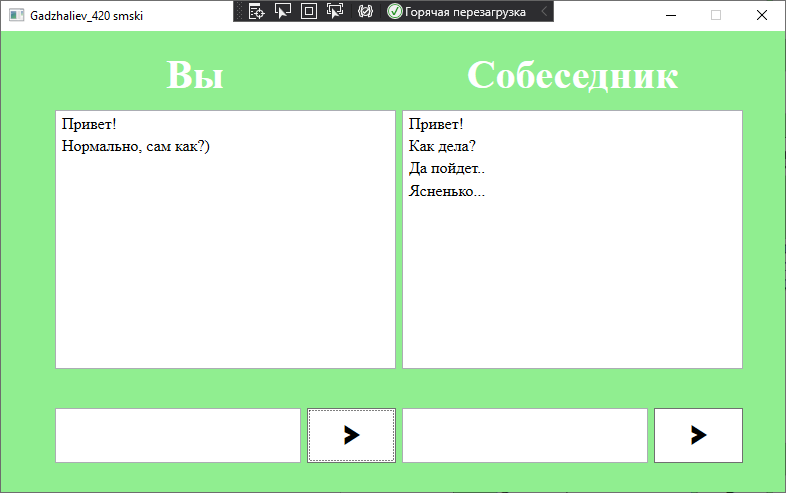
}

}

}







Вывод: я разобрался и переписал приложение под WPF.