

Лабораторная работа № 1

Технологии распределенной обработки данных

Тема: Создание простейшего распределённого приложения на
основе технологии .NET Remoting
Вариант № 5

Проверил:

Гай В. Е.

Выполнил:

Студент гр. 14-В-1

Кузнецова П.В.

Нижний Новгород 2016

Инв. № подл.	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата

1. Цель и порядок выполнения работы

Целью данной лабораторной работы является изучение и практическое освоение возможностей технологии «.NET Remoting».

Вариант индивидуального задания:

Разработайте распределённую систему вычисления синуса угла. В обмене данными участвуют один клиент и один сервер.

Для выполнения данной лабораторной работы были выполнены следующие шаги:

1. Разработка структуры клиент-сервер.
2. Разработка библиотеки, доступной как клиенту, так и серверу.
3. Реализация индивидуального задания в библиотеке.
4. Реализация подключения сервера.
5. Реализация подключения клиента к серверу.
6. Реализация пользовательских интерфейсов для клиента и сервера.
7. Реализация обращения клиента к библиотеке для подсчета синуса.

[illegible]

2. Выполнение лабораторной работы

2.1. Реализация библиотеки

Создана библиотека классов для удаленного объекта. Класс SampleObject унаследован от MarshalByRefObject.

В этом классе реализованы методы отправки сообщения на сервер, получение сообщения с сервера, а также реализован метод Sinus для подсчета синуса угла.

```
public void SendMsgToSvr(string chatMsgFromUsr)
{
    hTChatMsg.Add(++key, chatMsgFromUsr);
}
public string GetMsgFromSvr(int lastKey)
{
    if (key > lastKey)
        return hTChatMsg[lastKey + 1].ToString();
    else
        return "";
}

public double Sinus(double a)
{
    return ((double)(Math.Sin(a)));
}
```

2.2. Реализация серверного приложения

Для сервера создано новое консольное приложение RemoteServer. Чтобы использовать класс TcpChannel, сошлемся на сборку System.Runtime.Remoting. Также используем ссылку на сборку RemoteBase.

Создаем объект типа TcpChannel с портом номер 8080. Этот канал зарегистрирован с классом System.Runtime.Remoting.Channels.Tcp.TcpChannel для обеспечения его доступности удаленным объектам. Тип удаленного объекта зарегистрирован вызовом метода RemotingConfiguration.RegisterWellKnownServiceType().

Клиент использует тип URI класса удаленного объекта с указанием режима. Режим WellKnownObjectMode.Singleton означает, что каждое входящее сообщение обслуживается тем же экземпляром объекта.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;
using RemoteBase;
namespace RemoteServer
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        TcpChannel channel;
        private void btnStart_Click(object sender, EventArgs e)
        {
            if (channel == null)
            {
                channel = new TcpChannel(8080);
                ChannelServices.RegisterChannel(channel, false);
                RemotingConfiguration.RegisterWellKnownServiceType(typeof(SampleObject), "ChatRoom", WellKnownObjectMode.Singleton);
            }
        }
    }
}

```

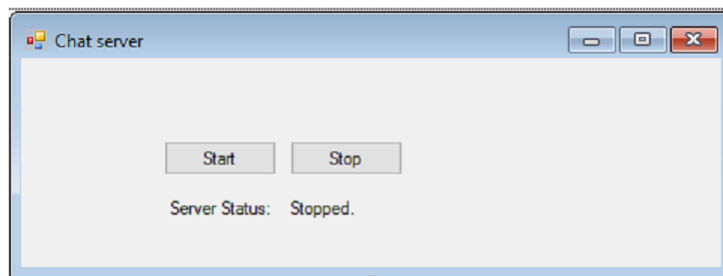


Рисунок 2.1. Форма chatServer

Инв. № подл	Подп. и дата	Инв. № дубл.	Взам. инв. №	Подп. и дата						Лист
										4
Ли	Изм.	№ докум.	Подп.	Дат	Лабораторная работа № 1					

2.3. Реализация клиентского приложения

Для клиента также создано консольное приложение RemotingClient. В этом проекте ссылаемся на сборку System.Runtime.Remoting, чтобы использовать класс TcpChannel. Также ссылаемся на сборку RemoteBase. Хотя объект будет создан на удаленном сервере, эта сборка необходима клиенту, чтобы прокси мог прочитать информацию о типе во время выполнения. В клиентской программе создаем объект TcpChannel, зарегистрированный в ChannelServices. Для TcpChannel можно использовать конструктор по умолчанию, выбирающий свободный порт. С помощью класса Activator возвращается прокси удаленного объекта. Этот объект выглядит как реальный объект, потому что предоставляет те же методы. Прозрачный прокси использует реальный прокси для отправки сообщений в канал. Это было реализовано в форме frmLogin.

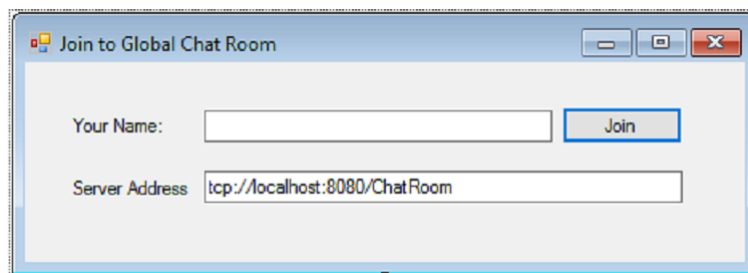


Рисунок 2.2. Форма frmLogin

В форме frmChatWin была создана функция SendMessage, в которой было реализовано получение введенного пользователем значения функцией Sinus и отправка результата вычислений на экран.

```
private void SendMessage()
{
    if (remoteObj != null && txtChatHere.Text.Trim().Length>0)
    {
        double d = double.Parse(txtChatHere.Text);
        double x = remoteObj.Sinus(d);
        label3.Text = x.ToString();
    }
}
```

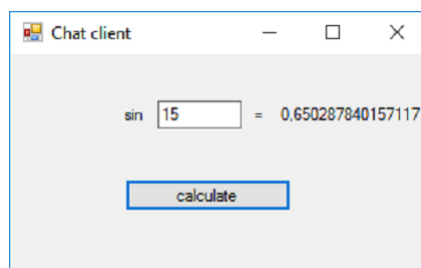


Рисунок 2.3. Форма frmChatWin

Выводы

Таким образом, в ходе данной лабораторной работы были изучены и освоены на практике технологии ".NET Remoting". Разработана распределенная система вычисления синуса угла. Реализована система клиент-сервер на основе взаимодействия клиента и сервера с общей библиотекой.