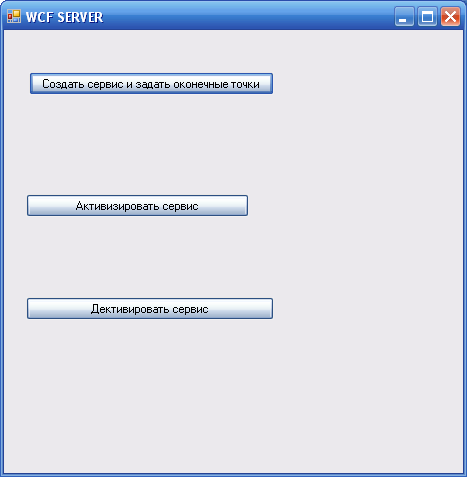
**Пример реализации системи WCF**

**1. Форма и код WCF – сервера ( службы)**

****

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.ServiceProcess;

using System.ServiceModel;

using System.Text;

using System.Windows.Forms;

namespace WAppWCFServ

{

[ServiceContract] // определение контракта

public interface IStockService

{

[OperationContract] // определение методов контракта

double GetPrice(int ticket);

[OperationContract] // определение методов контракта

string GetStr(string str);

[OperationContract] // определение методов контракта

DataTable GetTab(string str);

}

public partial class Form1 : Form

{

public ServiceHost serviceHost;

public Form1()

{

InitializeComponent();

}

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

{

// создаем сервис(службу) и указываем её размещение

//ServiceHost serviceHost = new ServiceHost(класс реализации, Uri(адрес поступления запросов)

serviceHost = new ServiceHost(typeof(StockService), new Uri("http://localhost:8001/WAppWCFServ"));

// далее определяем конечную точку сервиса

// serviceHost.AddServiceEndPoint(контракт, соединение(привязка), адрес);

// serviceHost.ChannelDispatchers.

// SecurityMode smod = new SecurityMode();

// smod = SecurityMode.None;

// NetTcpBinding bind = new NetTcpBinding(smod,false);

serviceHost.AddServiceEndpoint(typeof(IStockService), new WSHttpBinding(), "");

this.label1.Text = " сервис создан, точка определена";

}

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

{

}

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

{

serviceHost.Open();

this.label2.Text = " Сервис активизирован";

}

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

{

serviceHost.Close();

this.label3.Text = " Сервис деактивизирован и уничтожен";

}

}

public class StockService : IStockService // класс реализации методов контракта

{

public double GetPrice(int ticket)

{

MessageBox.Show( " Получен параметр double = " + ticket.ToString());

if (ticket == 1) {return 100.25;}

if (ticket == 2) {return 125.5;}

return 0;

}

public string GetStr(string str)

{

MessageBox.Show(" Получен параметр = " + str);

return "Ваш параметр = " + str;

}

public DataTable GetTab(string str)

{

DataTable restab;

OleDbConnection conn10;

OleDbCommand com10;

OleDbDataAdapter da;

MessageBox.Show(" Получен параметр = " + str);

conn10 = new OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=D:\ASOER\_CB\ASOER\_DB\ASOER\_ML.mdb");

conn10.Open();

com10 = conn10.CreateCommand();

com10.CommandText = "Select \* from " +str;

da = new OleDbDataAdapter(com10.CommandText, conn10);

restab = new DataTable();

restab.TableName = str;

da.Fill(restab);

MessageBox.Show(restab.Rows[2][1].ToString());

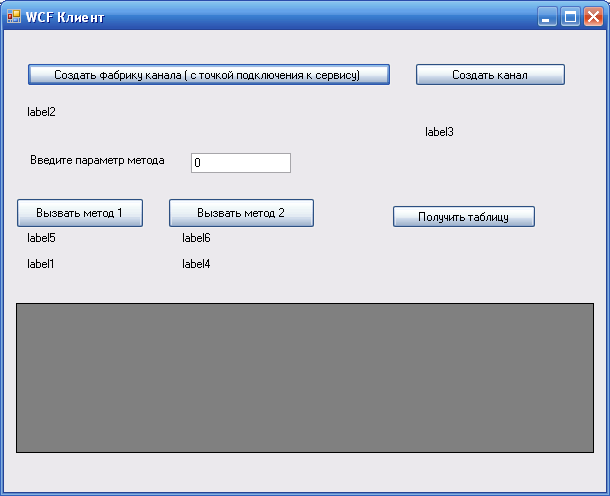
return restab;

}

}

}

**2. WCF – desktop Клиент**

****

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.ServiceModel;

using System.ServiceModel.Configuration;

using System.ServiceModel.Description;

using System.Runtime.Serialization;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace WindowsApplication1

{ [ServiceContract] // определение контракта

public interface IStockService

{

[OperationContract] // определение методов контракта

double GetPrice(int ticket);

[OperationContract] // определение методов контракта

string GetStr(string str);

[OperationContract] // определение методов контракта

DataTable GetTab(string str);

}

public partial class Form1 : Form

{

public ChannelFactory<IStockService> myChannelFactory;

public IStockService wcfClient;

public double p;

public int ticket;

public string str,strr;

public Form1()

{

InitializeComponent();

}

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

{

//Client client = new Client();

// wcfCl = client.getwcfClient();

WSHttpBinding binding = new WSHttpBinding();

// WSDualHttpBinding binding = new WSDualHttpBinding();

// binding.Name = "WSDualHttpBinding\_IStockService";

// binding.ClientBaseAddress = new Uri("http://localhost:8001/WAppWCFServ");

binding.ReliableSession.Ordered = true;

binding.ReliableSession.InactivityTimeout = new TimeSpan(0, 20, 0);

binding.ReceiveTimeout = new TimeSpan(0, 20, 0);

binding.SendTimeout = new TimeSpan(0, 00, 10);

myChannelFactory = new ChannelFactory<IStockService>

(binding, new EndpointAddress("http://localhost:8001/WAppWCFServ"));

wcfClient = myChannelFactory.CreateChannel();

if (myChannelFactory == null)

label2.Text = "Фабрика канала НЕ создана";

else

label2.Text = "Фабрика канала создана";

}

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

{

if (wcfClient != null)

{

this.label1.Text = "";

ticket = Convert.ToInt32(this.textBox1.Text);

label5.Text = "Отправлен параметр = " + ticket.ToString();

p = wcfClient.GetPrice(ticket);

this.label1.Text = "Получен результат = " + p.ToString();

}

else

label1.Text = " Сервис НЕ подключен";

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

if (wcfClient != null)

{

this.label4.Text = " ";

str = this.textBox1.Text;

// t = &ticket;

label6.Text = "Отправлен параметр = " + str ;

// double p = wcfCl.GetPrice(ticket);

// p = wcfClient.GetPrice( \*t);

strr = wcfClient.GetStr(str);

this.label4.Text = "Получен результат : " + strr;

}

else

label1.Text = " Сервис НЕ подключен";

}

private void label6\_Click(object sender, EventArgs e)

{

}

private void button4\_Click(object sender, EventArgs e)

{

if (wcfClient != null)

{

string str1 = this.textBox1.Text;

DataTable rtab = new DataTable();

rtab = wcfClient.GetTab(str1);

this.dataGridView1.DataSource = rtab;

}

else

label1.Text = " Сервис НЕ подключен";

}

private void button5\_Click(object sender, EventArgs e)

{

wcfClient = myChannelFactory.CreateChannel();

if (wcfClient == null)

label3.Text = " Канал НЕ создан, Сервис НЕ подключен";

else

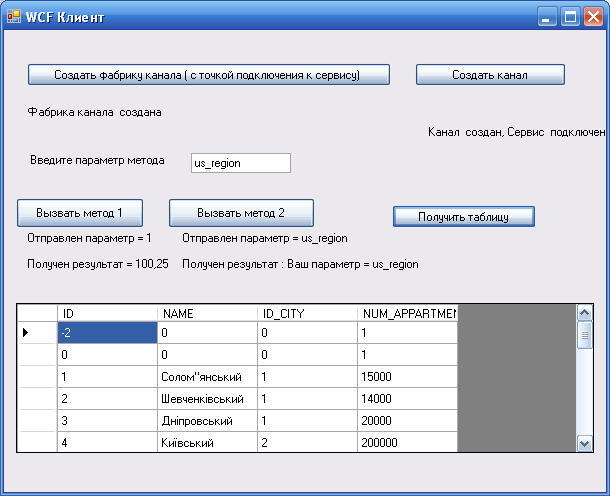
label3.Text = " Канал создан, Сервис подключен";

}

}

}

Результаты вызова

****