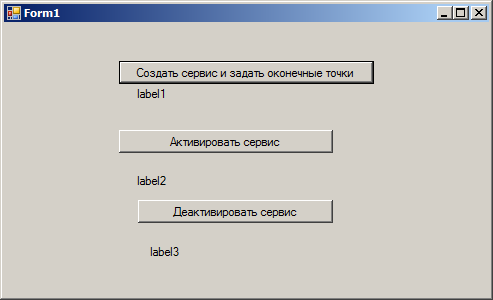
**Создание системы Клиент – сервер без точки MEX с использованием файлов конфигурации на клиенте и сервере**

Форма сервера



**Код сервера**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Data.OleDb;

using System.ServiceProcess;

using System.ServiceModel;

using System.ServiceModel.Description;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace W7F\_WCF\_SimpSerCnfg

{

[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 button1\_Click(object sender, EventArgs e)

{

serviceHost = new ServiceHost(typeof(StockService));

this.label1.Text = " Служба создана";

}

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

{

serviceHost.Open();

this.label2.Text = " Сервис активизирован";

}

private void button3\_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)

{

OleDbConnection cn = new OleDbConnection();

cn.ConnectionString = @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=D:\ASOER\_CB\ASOER\_DB\ASOER\_ML.mdb";

cn.Open();

OleDbCommand cmd = cn.CreateCommand();

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

OleDbDataReader dr = cmd.ExecuteReader();

DataTable dt = new DataTable("Table");

dt.Load(dr);

// MessageBox.Show((dt.Rows[2][1]).ToString() + "-" + (dt.Rows[2][2]).ToString() + "-" + (dt.Rows[2][3]).ToString());

return dt;

}

}

}

**Файл конфигурации сервера**

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />

</startup>

<system.serviceModel>

<services>

<service name ="W7F\_WCF\_SimpSerCnfg.StockService">

<host>

<baseAddresses>

<add baseAddress ="http://localhost:8002/W7F\_WCF\_SimpSerCnfg"/>

</baseAddresses>

</host>

<endpoint address ="" binding ="basicHttpBinding" contract ="W7F\_WCF\_SimpSerCnfg.IStockService" />

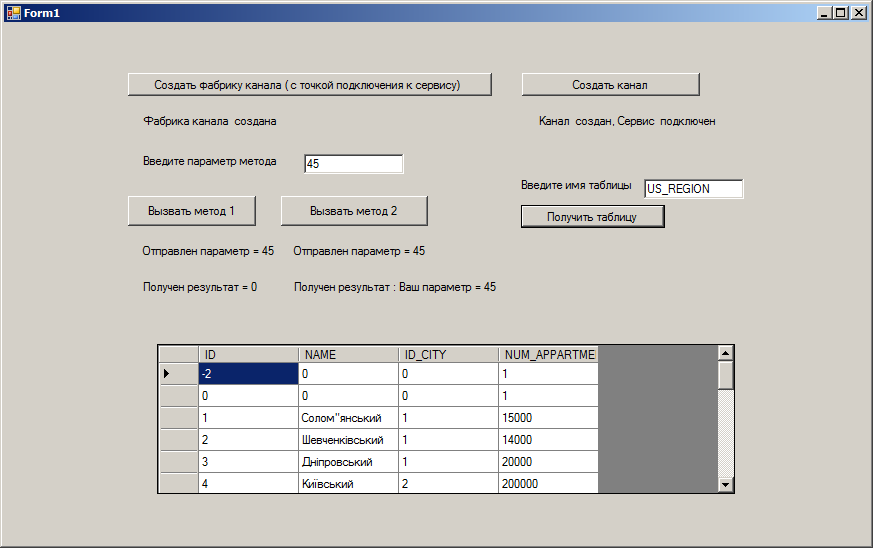
</service>

</services>

</system.serviceModel>

</configuration>

**Форма клиента**



**Код клиента**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.ServiceModel;

using System.ServiceModel.Configuration;

using System.ServiceModel.Description;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace W7F\_WCF\_SimpClientCnfg

{

[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)

{

myChannelFactory = new ChannelFactory<IStockService>("BasicHttpBinding\_IStockService");

// myChannelFactory = new ChannelFactory<IStockService>

// (new BasicHttpBinding(),new EndpointAddress("http://localhost:8002/W7F\_WCF\_SimpSerCnfg"));

if (myChannelFactory == null)

label2.Text = "Фабрика канала НЕ создана";

else

label2.Text = "Фабрика канала создана";

}

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

{

// StockService wcfClient = new StockService();

wcfClient = myChannelFactory.CreateChannel();

if (wcfClient == null)

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

else

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

}

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 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 button4\_Click(object sender, EventArgs e)

{

if (wcfClient != null)

{

string str1 = this.textBox2.Text;

DataTable rtab = new DataTable("Table");

rtab = wcfClient.GetTab(str1);

this.dataGridView1.DataSource = rtab;

}

else

label1.Text = " Сервис НЕ подключен";

}

}

}

**Файл конфигурации клиента**

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />

</startup>

<system.serviceModel>

<bindings>

<basicHttpBinding>

<binding name ="BasicHttpBinding\_IStockService"

closeTimeout ="00:01:00"

sendTimeout ="00:01:00"

/>

</basicHttpBinding>

</bindings>

<client>

<endpoint address ="http://localhost:8002/W7F\_WCF\_SimpSerCnfg"

binding ="basicHttpBinding"

bindingConfiguration ="BasicHttpBinding\_IStockService"

contract ="W7F\_WCF\_SimpClientCnfg.IStockService"

name ="BasicHttpBinding\_IStockService"

/>

</client>

</system.serviceModel>

</configuration>