**Создание CORBA - сервера**

**Объявление интерфейса**

using System; using System.Runtime.Remoting; using System.Drawing; using Ch.Elca.Iiop.Idl;

namespace CorbaDemo

{

public interface ITriangle

{

double Perimetr(Point a, Point b, Point c);

double Square(Point a, Point b, Point c);

}

}

**Класс -имплементации методов интерфейса (CORBA-объект)using System;**

using System.Drawing; using System.Runtime.Remoting; using System.Runtime.Remoting.Channels;

namespace CorbaDemo

{

public class Triangle: MarshalByRefObject, ITriangle

{

public override object InitializeLifetimeService()

{

return null;

}

private double Length(Point a, Point b)

{

return Math.Sqrt(Math.Pow(a.X - b.X, 2.0) + Math.Pow(a.Y - b.Y, 2.0));

}

public double Perimetr(Point a, Point b, Point c)

{

return Length(a, b) + Length(b, c) + Length(c, a);

}

public double Square(Point a, Point b, Point c)

{

double p = Perimetr(a, b, c)/2.0;

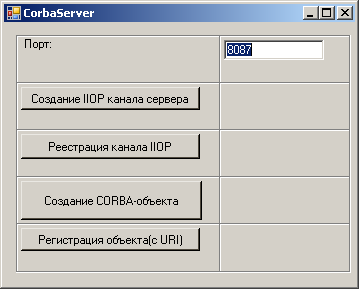
return Math.Sqrt(p \* (p - Length(a, b)) \* (p - Length(b, c)) \* (p - Length(c, a)));

}

}

}

**Форма сервера**



**Corba- сервер с сервантом**

using System; using System.Collections.Generic; using System.ComponentModel;

using System.Data; using System.Drawing; using System.Linq; using System.Text; using System.Windows.Forms;

using Ch.Elca.Iiop;

using System.Runtime.Remoting; using System.Runtime.Remoting.Channels; using System.Threading;

namespace CorbaDemo

{

public partial class CorbaServer : Form

{

private IiopChannel chan=null;

private int port = 8087;

private Triangle Otriangle = null;

public CorbaServer()

{

InitializeComponent();

}

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

{

try

{ // Создание канала

port = Int32.Parse(tPort.Text);

chan = new IiopChannel(port);

lCreate.Text = "Канал IIOP cоздан";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message,"Ошибка",MessageBoxButtons.OK,MessageBoxIcon.Error);

}

}

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

{

try

{ // Регистрация канала

ChannelServices.RegisterChannel(chan, false);

lRegister.Text = "Канал IIOP pарегистрирован";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

try

{ // Создание объекта

Otriangle = new Triangle();

lServant.Text = "Объект создан ";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

try

{ // Регистрация(публикация объекта)

RemotingServices.Marshal(Otriangle, "triangle");

lMarshal.Text = "Объект зарегистрирован(опубликован)";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**Создание клиента**

**Объявление интерфейса**

using System;

using System.Runtime.Remoting;

using System.Drawing;

using Ch.Elca.Iiop.Idl;

namespace CorbaDemo

{

public interface ITriangle

{

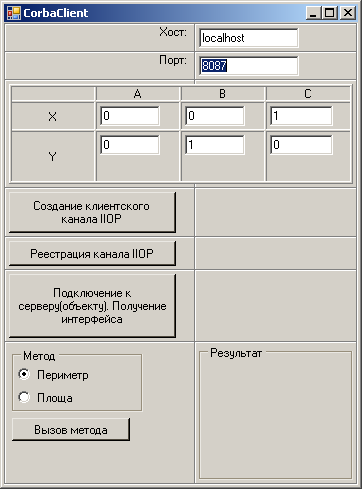
double Perimetr(Point a, Point b, Point c);

double Square(Point a, Point b, Point c);

}

}

Форма клиента



**Код клиента**

**Создание прокси и вызов методов**

using System; 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 CorbaDemo;

using Ch.Elca.Iiop;

using omg.org.CosNaming;

namespace CorbaClient

{

public partial class CorbaClient : Form

{

private string host="localhost", port="8087";

private IiopClientChannel channel = null;

private ITriangle Itriangle = null;

private Point a, b, c;

public CorbaClient()

{

InitializeComponent();

}

private void GetPoints()

{

try

{

a.X = Int32.Parse(tAx.Text); a.Y = Int32.Parse(tAy.Text); b.X = Int32.Parse(tBx.Text); b.Y = nt32.Parse(tBy.Text);

c.X = Int32.Parse(tCx.Text); c.Y = Int32.Parse(tCy.Text);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

try

{ // Создание клиентского канала

channel = new IiopClientChannel();

lCreate.Text = "Канал IIOP создан";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

double res;

GetPoints();

try

{

if (radioButton1.Checked == true)

{

res = Itriangle.Perimetr(a, b, c);

lResult.Text = String.Format("Периметр = {0} ед.", res);

}

else

{

res = Itriangle.Square(a, b, c);

lResult.Text = String.Format("Площадь = {0} кв.ед.", res);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка вызова метода, либо серванта", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

Try

{ // Регистрация клиентского канала

ChannelServices.RegisterChannel(channel, false);

lRegister.Text = "Канал IIOP зарегистрирован";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

try

{ // создание прокси и получение интерфейса

host = tHost.Text;

port = tPort.Text

Itriangle = (ITriangle)RemotingServices.Connect(typeof(ITriangle), "iiop://" + host + ":" + port + "/triangle");

lConnect.Text = "Прокси создан.Интерфейс получен";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}