МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ»

(КНИТУ-КАИ)

Институт компьютерных технологий и защиты информации

Кафедра автоматизированных систем обработки информации и управления

Направление подготовки 09.03.02: «Информационные системы и технологии»

**Курсовая работа**

по дисциплине: «Технологии программирования»

на тему: «Клиент-серверное приложение. Remoting, ADO.NET и ASP.NET»

|  |  |  |
| --- | --- | --- |
| Обучающийся группы 4309 | (подпись, дата) | Погасий Кирилл Всеволодович |
| Обучающийся группы 4309 | (подпись, дата) | Гайнулов Айдар |

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель | Ст. препод.  (должность) | (подпись, дата) | Вафин Радик Рашитович  (Ф.И.О.) |

|  |  |
| --- | --- |
| Курсовая работа зачтена с оценкой | (оценка) |
|  | (подпись, дата) |

Казань, 2018

MINISTRY OF EDUCATION AND SCIENCE OF THE RUSSIAN FEDERATION

Federal State Budget Educational Institution of Higher Education «Kazan National Research Technical University. A.N. Tupolev-KAI »

(KNRTU-KAI)

Institute of Computer Technologies and Information Protection

Department of Automated Information Processing and Control Systems

Direction of training 09.03.02: «Information systems and technologies»

**Course work**

on discipline: «Programming Technologies»

on the topic: «Client-server application. Remoting and ASP.NET»

|  |  |  |
| --- | --- | --- |
| Learning group 4309 | (signature, date) | Pogasiy Kirill Vsevolodovich |
| Learning group 4309 | (signature, date) | Gainulov Aidar |

|  |  |  |  |
| --- | --- | --- | --- |
| Head of | S. Lecturer  (position) | (signature, date) | Vafin Radik Rashitovich  (S.N.P.) |

|  |  |
| --- | --- |
| Course work is read off with an assessment | (assessment) |
|  | (signature, date) |

Kazan, 2018

**Содержание**

[Аннотация 4](#_Toc514583166)

[Задание 6](#_Toc514583167)

[Анализ предметной области 7](#_Toc514583168)

[Структура системы 8](#_Toc514583169)

[Описание модулей системы 10](#_Toc514583170)

[Инструкция по установке системы 10](#_Toc514583171)

[Инструкция по использованию системы 11](#_Toc514583172)

[Заключение 13](#_Toc514583173)

[Список литературы 14](#_Toc514583174)

[Приложения 15](#_Toc514583175)

Аннотация

В данной курсовой работе приведен пример разработки клиент-серверного приложения, использующего технологии Remoting, ADO.NET и ASP.NET для обеспечения связи между клиентским и серверным приложением.

Серверное приложение разработано на языке программирования C#. Серверное приложение создает на своей стороне удаленный объект, который хранит в себе всю информацию об пользователе, а также позволяет пользователю работать с базой данных.

Клиентское приложение разработано на языке программирования C#. Он визуализирует элементы управления и передает серверу команды о совершении действий пользователем.

Веб приложение разработано на языках C#, HTML и JavaScript оно создает динамическую вэб-форму и передает ее пользователю через http протокол для открытия в браузере.

**Annotation**

This course work provides an example of developing a client-server application that uses Remoting, ADO.NET and ASP.NET technologies to provide communication between a client and a server application.

The server application is developed in the C # programming language. The server application creates on its side a remote object that stores all the information about the user, and also allows the user to work with the database.

The client application is developed in the C # programming language. It visualizes the controls and sends commands to the server to perform actions by the user.

A web application is developed in C #, HTML and JavaScript languages; it creates a dynamic web form and sends it to the user via the http protocol for opening in a browsers.

Задание

Необходимо разработать клиент-серверное приложение, использующее технологии Remoting, ADO.NET и ASP.NET для обеспечения связи между клиентом и сервером.

Архитектура: клиент – сервер.

В курсовой работе должно быть разработано 4 модуля:

1. Сервис – сервер для создания распределенного объекта;
2. Клиент – оконное приложение;
3. Библиотек классов.
4. Сервис – ASP.NET

Сервер – приложение Windows Forms. Он создает объект, который хранит в себе информацию о процессе выполнения запросов, и дает доступ к нему всем пользователям по каналу связи.

Клиент – приложение Windows Forms, взаимодействующее с сервером, с помощью технологии Remoting и ASP.NET, устанавливая с сервером TCP/IP либо HTTP соединение. Он должен визуализировать элементы управления. Клиентское приложение не должно хранить в себе реализацию удаленного объекта, а только интерфейс на методы данного объекта.

Библиотека классов – библиотека распределенного объекта, который обеспечивает выполнение запросов пользователя.

Необходимо использовать конфигурационные файлы для настройки соединения.

Каналы соединения должны быть защищены.

Должно быть установлено изначальное время жизни удаленного объекта в 300 секунд, а также реализован спонсор.

Должен быть реализован зашифрованный вход для администратора.

Анализ предметной области

Разработка данного приложения подразумевает его использование в бизнес-среде, например, в качестве приложения для кассовых автоматов, оно позволяет легко администрировать масштабируемую базу данных.

Проанализировав данное приложение, мы пришли к выводу, что в окне клиента должны присутствовать следующие элементы:

* Элементы управления (кнопки)
* Поле вывода информации

Для серверного и клиентского приложений мы выбрали язык программирования C#, т.к. на этом языке удобно работать с технологиями Remoting и ASP.NET и строить визуальные приложения, т.к. Microsoft Visual Studio дает возможность быстро разместить на нужных местах элементы управления и назначить на них необходимые события с помощью конструктора Windows Forms приложений.

Структура системы

**Технология Remoting**

Технология Remoting была разработана для создания распределенных приложений. С ее помощью можно обращаться к экземплярам классов .Net, находящимся за пределами собственного домена (application domain). Это может быть другое приложение внутри одного процесса, другой процесс на той же машине, или процесс на другой машине (в том числе подключенной через Internet).

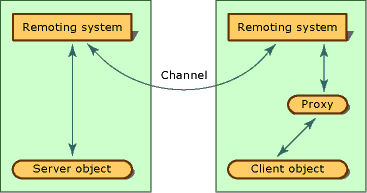


Рис. 1. Схема взаимодействия модулей в технологии Remoting

Система состоит из двух частей сервера и клиента.

Сервер при запуске настраивает Remoting для использования определенного протокола. При этом указывается транспортный протокол и протокол доступа. Затем сервер регистрирует все классы, к которым он предоставляет доступ.

Клиент при необходимости доступа к удаленным объектам также указывает протоколы доступа и посылает запрос на сервер. Сервер в ответ на этот запрос создает у себя затребованный объект и передает его идентификатор клиенту. Клиент создает у себя специальный прокси-класс, который затем и использует, как если бы это был объект в его собственном домене.

**Технология ASP.NET**

ASP.NET (Active Server Pages для .NET) — платформа разработки веб-приложений, в состав которой входит: веб-сервисы, программная инфраструктура, модель программирования, от компании Майкрософт. ASP.NET входит в состав платформы .NET Framework и развитием более старой технологии Microsoft ASP.



Рис. 2 Схема взаимодействия модулей в технологии ASP.NET

Система состоит из двух частей сервера и клиента.

Сервер служит для создания и отправки html страницы по заранее заданному скрипту и обработки запросов клиентской программы с последующим формированием и отправкой клиенту html страницы.

Клиент служит для отображения информации и создания запросов на сервер для обработки данных клиента. Чаще всего в качестве клиента выступает WEB-браузер.

Описание модулей системы

1. Библиотека классов (ClassLibrary)
   * + Class1.cs – класс работы с базой данных. Создается подключение к базе данных. Метод записи значений в базу данных. Метод считывания значений из базы данных.
     + Interface1.cs – класс-интерфейс, через который пользователь взаимодействует с распределенным объектом.
2. Сервер (Server – Program.cs)

* Создает информационное окно;
* Выполняет проверку и очищает таблицу от записей с истекшим сроком

1. Клиент (Client – Form1.cs)
   * + Выбирает способ создания канала связи с сервером (с помощью конфигурационного файла или кода).
     + В зависимости от выбранного способа создается канал связи.
     + Получает доступ к распределенному объекту.
     + Назначает распределенному объекту спонсора.
2. Сервер (ASP.NET)
   * + Дублирует возможности клиента, позволяя использовать браузер на любой платформе

Инструкция по установке системы

Для работы серверного и клиентского приложений необходимо наличие операционной системы Windows на рабочей машине. Для запуска необходимо поочередно запустить Server.exe и Client.exe.

Инструкция по использованию системы

Для использования системы нужно в клиентском окне выбрать способ подключения (TCP/HTTP(конф. файл)).

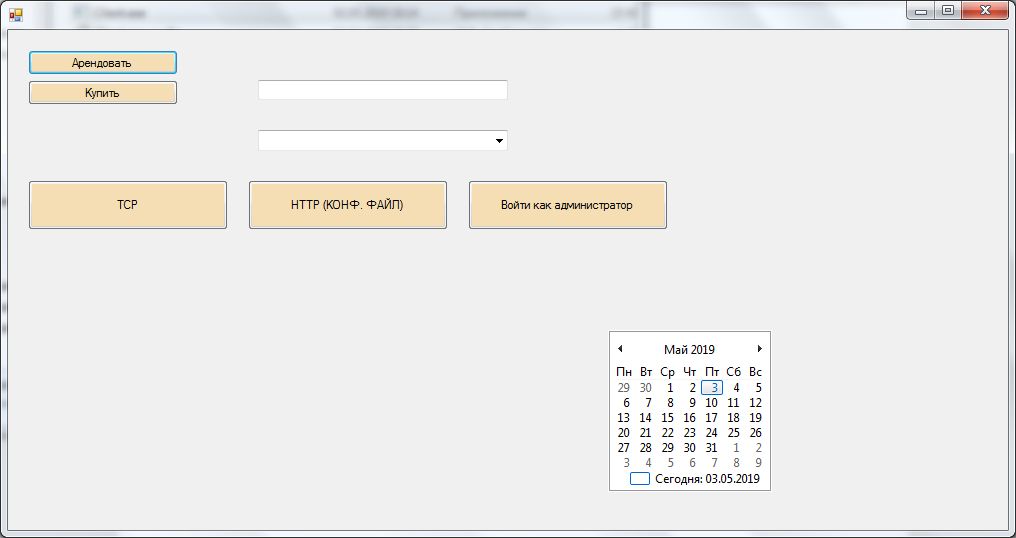


Рис. 3. Окно клиента (после запуска)

Затем в зависимости от желания выбрать дату при аренде, выбрать тип сервера из выпадающего списка, нажать на кнопку с тем видом заказа, который вам нужен. После нажатия вы получите IP адрес сервера.

При необходимости вы можете войти как администратор, нажав на соответствующую кнопку и введя данные в новой форме (Рис. 4)

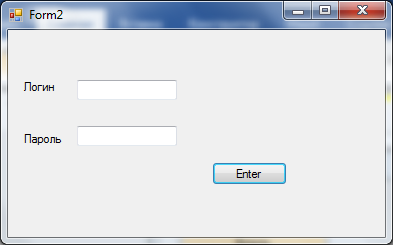


Рис. 4 Окно входа администратора

После входа в качестве администратора появится дополнительная кнопка «Добавить сервер», которая позволяет, вписав IP в соответствующее поле и выбрав тип сервера добавить в базу данных новый сервер (Рис.5)

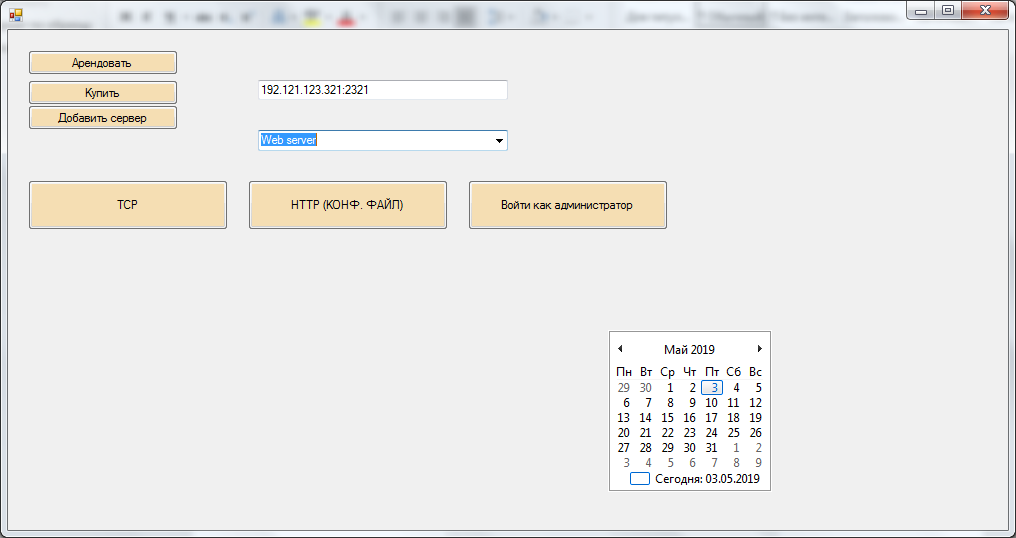


Рис. 5. Окно клиента при добавлении нового сервера

Стоит отметить, что спонсор перестанет продлевать время жизни удаленного объекта, если пользователь не будет кликать в течение 30 секунд на клиентское окно.

Заключение

В результате выполнения данной курсовой работы:

* была изучена методология построения клиент-серверных приложений с применением технологий Remoting, ADO.NET и ASP.NET на примере бизнес-программы по продаже и аренде серверов;
* изучены возможности языка и C# в разрезе клиент серверного взаимодействия;

В данном программном решении были реализованы все функции, предусмотренные в проекте.

Список литературы

1. Албахари Джозеф, Албахари Бен – C# 6.0. Справочник. Полное описание языка / Изд.: Вильямс / 2016 г. / 1040 стр. / ISBN: 978-5-8459-2087-4
2. Ingo Rammer – Advanced .NET Remoting (C# Edition) / Apress / 2002 year / ISBN: 978-1-4302-0833-4
3. Симон Робинсон - C# для профессионалов. Том 2 / Cимон Робинсон, Олли Корнес, Джей Глинн, Бартон Харвей, Крейг Макквин, Джерод Моемека, Кристиан Нагель, Морган Скиннер, Карли Ватсон / Изд.: Лори / 2005 г. / 522 стр. / ISBN: 5-85582-187-0
4. Microsoft .NET Remoting: A Technical Overview / [Электронный ресурс] / URI: https://msdn.microsoft.com/en-us/library/ms973857.aspx

Приложения

Приложение 1. ClassLibrary – obj.cs

using OTCHET;

using System;

using System.Data.OleDb;

using System.Runtime.Remoting.Lifetime;

using System.Security.Cryptography;

using System.Text;

namespace Obj

{

public delegate void Del(string s);

public class UObj : MarshalByRefObject, Interface1

{

public event Del even;

string type\_serv;

public int x = 0, y = 0, z = 0;

public bool d = false, f = false, g = false;

public static string connectString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=BD.mdb;";

private OleDbConnection myConnection;

RSACryptoServiceProvider RSA;

public void Connection() //подключение к БД

{

try { myConnection.Close(); }

catch { }

myConnection = new OleDbConnection(connectString);

myConnection.Open();

//myConnection

}

//добавление нового сервера

public bool Add\_serv(string IP, string Type)

{

try

{

if (IP.Length < 13 || Type == null)

throw new Exception("ВВедите ip и тип правильно");

connectString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=BD.mdb;";

Connection();

string query = "INSERT INTO Таблица1 (IP, Type) VALUES('" + IP + "','" + Type + "')";

// создаем объект OleDbCommand для выполнения запроса к БД MS Access

OleDbCommand command = new OleDbCommand(query, myConnection);

// выполняем запрос к MS Access

if (command.ExecuteNonQuery() == 0)

return false;

else return true;

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

return false;

}

//поиск и запись арендованного сервера

public string Prihod(string Type\_serv, DateTime Exp\_date)

{

connectString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=BD.mdb;";

Connection();

string IP;

string query = "SELECT IP from Таблица1 WHERE Type = '" + Type\_serv + "' AND Zanyat = 0";

OleDbCommand command = new OleDbCommand(query, myConnection);

OleDbDataReader reader = command.ExecuteReader();

if (reader.Read())

{

IP = reader.GetString(0);

query = "Update Таблица1 SET Zanyat= 1, Buy\_date = '" + DateTime.Now.ToString("yyyy-MM-dd") + "', Expire\_date='" + Exp\_date.ToString("yyyy-MM-dd") + "'where IP ='" + IP + "'";

command = new OleDbCommand(query, myConnection);

command.ExecuteNonQuery();

return IP;

}

else return null;

}

//Покупка (удаление) сервера

public string Uhod(string Type\_serv)

{

connectString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=BD.mdb;";

Connection();

string IP;

string query = "SELECT IP from Таблица1 WHERE Type = '" + Type\_serv + "' AND Zanyat = 0";

OleDbCommand command = new OleDbCommand(query, myConnection);

OleDbDataReader reader = command.ExecuteReader();

if (reader.Read())

{

IP = reader.GetString(0);

query = "DELETE FROM Таблица1 where IP ='" + IP + "'";

command = new OleDbCommand(query, myConnection);

command.ExecuteNonQuery();

return IP;

}

else return null;

}

public RSAParameters Key\_send()

{

RSA = new RSACryptoServiceProvider();

return RSA.ExportParameters(false);

}

public bool Enter\_serv(byte[] Login, byte[] Pas)

{

connectString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=BD1.mdb;";

Connection();

string login, pas;

try

{

login = Encoding.UTF8.GetString(RSA.Decrypt(Login, false));

pas = Encoding.UTF8.GetString(RSA.Decrypt(Pas, false));

}

catch (ArgumentNullException)

{

Console.WriteLine("Encryption failed.");

return false;

}

OleDbDataReader reader;

try

{

OleDbCommand command = new OleDbCommand("SELECT Key from Таблица1 WHERE Login = '" + login + "'", myConnection);

reader = command.ExecuteReader();

if (reader.Read())

{

if (reader.GetString(0).Equals(pas))

return true;

else return false;

}

else return false;

}

catch

{

return false;

}

}

public bool Enter\_serv(string login, string pas)

{

connectString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=BD1.mdb;";

Connection();

OleDbDataReader reader;

try

{

OleDbCommand command = new OleDbCommand("SELECT Key from Таблица1 WHERE Login = '" + login + "'", myConnection);

reader = command.ExecuteReader();

if (reader.Read())

{

if (reader.GetString(0).Equals(pas))

return true;

else return false;

}

else return false;

}

catch

{

return false;

}

}

public int Check()

{

try

{

connectString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=BD.mdb;";

Connection();

string query = "Update Таблица1 SET Zanyat= 0, Buy\_date = null, Expire\_date=null where Expire\_date < date()";

OleDbCommand command = new OleDbCommand(query, myConnection);

return command.ExecuteNonQuery();

}

catch

{

return -1;

}

}

public override Object InitializeLifetimeService() //переопределяем время жизни каналов на 30 секунд

{

ILease lease = (ILease)base.InitializeLifetimeService();

if (lease.CurrentState == LeaseState.Initial)

{

lease.InitialLeaseTime = TimeSpan.FromSeconds(300);

lease.RenewOnCallTime = TimeSpan.FromSeconds(30);

lease.SponsorshipTimeout = TimeSpan.FromSeconds(45);

}

return lease;

}

}

}

Приложение 2. ClassLibrary interface – OTCHET.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Runtime.Remoting.Lifetime;

using System.Security.Cryptography;

namespace OTCHET

{

public interface Interface1

{

string Prihod(string Type\_serv, DateTime Exp\_date);

string Uhod(string Type\_serv);

bool Add\_serv(string IP, string Type);

RSAParameters Key\_send();

bool Enter\_serv(byte[] Login, byte[] Pas);

Object InitializeLifetimeService();

Object GetLifetimeService();

}

}

Приложение 3. Server – Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data;

using System.Data.OleDb;

using System.Runtime.Remoting;

using System.Runtime.Remoting.Channels;

using System.Runtime.Remoting.Channels.Tcp;

using System.Runtime.Remoting.Channels.Http;

using Obj;

using System.Drawing;

using System.IO;

using System.Runtime.Serialization.Formatters;

namespace Server

{

class Server : Form

{

public delegate void AddMessageDelegate(string message, string s);

UObj obj;

private TextBox txt, txt2;

public int x = 0, y = 0, z = 0;

public Server()

{

obj = new UObj();

//Создание канала

BinaryServerFormatterSinkProvider srvPrvd = new BinaryServerFormatterSinkProvider();

srvPrvd.TypeFilterLevel = TypeFilterLevel.Full;

BinaryClientFormatterSinkProvider clntPrvd = new BinaryClientFormatterSinkProvider();

Dictionary<string, string> proprt = new Dictionary<string, string>();

proprt["port"] = "8086";

TcpChannel channel = new TcpChannel(proprt, clntPrvd, srvPrvd);

//Регистация канала

ChannelServices.RegisterChannel(channel, true);

RemotingConfiguration.Configure("C:\\Users\\Kirushka\\Desktop\\vaflin\\config.config", false);

RemotingServices.Marshal(obj, "UdOb"); // задаем удаленный объект и слово через которое будет вестись связь

Create();

System.Threading.Timer timer = new System.Threading.Timer(tick, this, 0, 3600000);

}

void tick(object state)

{

obj.Check();

}

public void Create()

{

Label lab = new Label();

lab.Location = new Point(20, 10);

lab.Size = new Size(1000, 25);

lab.Text = "Сервер работает";

this.Controls.Add(lab);

}

public void AddTxt(string message, string s)

{

switch (s)

{

case "1":

txt.Text = message;

break;

case "2":

txt2.Text = message;

break;

}

}

static class Programm

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Server());

}

}

private void InitializeComponent()

{

this.SuspendLayout();

//

// Server

//

this.ClientSize = new System.Drawing.Size(284, 261);

this.Name = "Server";

this.ResumeLayout(false);

}

private void Server\_Load(object sender, EventArgs e)

{

}

}

}

Приложение 4. Client – Form1.Designer.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Runtime.Remoting;

using System.Runtime.Remoting.Channels;

using System.Runtime.Remoting.Channels.Tcp;

using System.Runtime.Remoting.Channels.Http;

using System.Drawing;

using System.IO;

using System.Runtime.Remoting.Lifetime;

using System.Runtime.Serialization.Formatters;

using System.Runtime.Remoting.Activation;

using OTCHET;

using System.Threading;

namespace Client

{

class Client : Form

{

public static DateTime Click\_time;

Interface1 obj;

public MonthCalendar cal;

private Button Prihod, Uhod, But1, But2, But3, But4; // Кнопки

public TextBox txt;

public ComboBox cmb, cmb1;

Form2 ent\_f;

ILease lease;

public bool is\_adm=false;

public Client()

{

Create();

Click\_time = DateTime.Now;

}

public void Create()

{

Color z = new Color(); //цвет кнопки

z = Color.Wheat;

this.ClientSize = new System.Drawing.Size(1000, 500);

Prihod = new Button(); // Создание кнопки

Prihod.Location = new Point(20, 20); // Расположение кнопки

Prihod.Size = new Size(150, 25); // Размер кнопки

Prihod.Name = "Prihod"; // Имя кнопки

Prihod.Text = "Арендовать"; // Текст на кнопки

Prihod.Click += new EventHandler(OnBut);

Prihod.BackColor = z;

this.Controls.Add(Prihod); // Добавление кнопки в окно клиента

Uhod = new Button(); // Создание кнопки

Uhod.Location = new Point(20, 50); // Расположение кнопки

Uhod.Size = new Size(150, 25); // Размер кнопки

Uhod.Name = "Uhod"; // Имя кнопки

Uhod.Text = "Купить"; // Текст на кнопки

Uhod.Click += new EventHandler(OnBut);

Uhod.BackColor = z;

this.Controls.Add(Uhod); // Добавление кнопки в окно клиента

But1 = new Button(); // Создание кнопки

But1.Location = new Point(20, 150); // Расположение кнопки

But1.Size = new Size(200, 50); // Размер кнопки

But1.Name = "But1"; // Имя кнопки

But1.Text = "TCP"; // Текст на кнопки

But1.Click += new EventHandler(Connection1);

But1.BackColor = z;

this.Controls.Add(But1); // Добавление кнопки в окно клиента

But2 = new Button(); // Создание кнопки

But2.Location = new Point(240, 150); // Расположение кнопки

But2.Size = new Size(200, 50); // Размер кнопки

But2.Name = "But2"; // Имя кнопки

But2.Text = "HTTP (КОНФ. ФАЙЛ)"; // Текст на кнопки

But2.Click += new EventHandler(Connection2);

But2.BackColor = z;

this.Controls.Add(But2); // Добавление кнопки в окно клиента

But3 = new Button(); // Создание кнопки

But3.Location = new Point(460, 150); // Расположение кнопки

But3.Size = new Size(200, 50); // Размер кнопки

But3.Name = "But3"; // Имя кнопки

But3.Text = "Войти как администратор"; // Текст на кнопки

But3.Click += new EventHandler(Admin\_ent);

But3.BackColor = z;

this.Controls.Add(But3); // Добавление кнопки в окно клиента

if (is\_adm)

{

But4 = new Button(); // Создание кнопки

But4.Location = new Point(20, 75); // Расположение кнопки

But4.Size = new Size(150, 25); // Размер кнопки

But4.Name = "Add"; // Имя кнопки

But4.Text = "Добаваить сервер"; // Текст на кнопки

But4.Click += new EventHandler(OnBut);

But4.BackColor = z;

this.Controls.Add(But4);

}

txt = new TextBox();

txt.Location = new Point(250, 50); // Расположение

txt.Size = new Size(250, 600); // Размер

txt.Name = "Txt"; // Имя кнопки

this.Controls.Add(txt);

cal = new MonthCalendar();

cal.Location = new Point(600, 300);

cal.Name = "Calendar";

Controls.Add(cal);

cmb = new ComboBox();

cmb.Location = new Point(250, 100);

cmb.Size = new Size(250, 600);

cmb.Name = "combo";

this.cmb.Items.AddRange(new object[]

{

"igrovoi",

"Web server",

"Application server"

});

this.Controls.Add(cmb);

this.Click += new System.EventHandler(Click\_on);

}

public void Connection1(object sender, EventArgs e)

{

try

{ //полный уровень десериализации

BinaryServerFormatterSinkProvider srvPrvd = new BinaryServerFormatterSinkProvider();

srvPrvd.TypeFilterLevel = TypeFilterLevel.Full; //уровень диссериализации

BinaryClientFormatterSinkProvider clntPrvd = new BinaryClientFormatterSinkProvider();

Dictionary<string, string> proprt = new Dictionary<string, string>();

proprt["port"] = "0";

TcpChannel channel = new TcpChannel(proprt, clntPrvd, srvPrvd);

ChannelServices.RegisterChannel(channel, true); //регистрация

obj = (Interface1)Activator.GetObject(typeof(Interface1), "tcp://localhost:8086/UdOb");

Sponsors();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

//http

public void Connection2(object sender, EventArgs e)

{

try

{

RemotingConfiguration.Configure("C:\\Users\\Kirushka\\Desktop\\vaflin\\confin.config", false);

obj = (Interface1)Activator.GetObject(typeof(Interface1), "http://localhost:8085/UdOb");

Sponsors();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

public void Sponsors()

{

lease = (ILease)obj.GetLifetimeService(); //информация об аренде

MyClientSponsor sponsor = new MyClientSponsor(); //создание спонсора

lease.Register(sponsor); //регистрация спонсора в службе аренды

}

public void Admin\_ent(object sender, EventArgs e)

{

ent\_f = new Form2(obj,ref is\_adm);

ent\_f.Show();

ent\_f.form\_hide += (adm\_form);

}

public void Click\_on(object sender, EventArgs e)

{

Click\_time = DateTime.Now;

}

void adm\_form(bool Is\_adm)

{

if (Is\_adm)

is\_adm = true;

ent\_f.Close();

But4 = new Button(); // Создание кнопки

But4.Location = new Point(20, 75); // Расположение кнопки

But4.Size = new Size(150, 25); // Размер кнопки

But4.Name = "Add"; // Имя кнопки

But4.Text = "Добавить сервер"; // Текст на кнопки

But4.Click += new EventHandler(OnBut);

But4.BackColor = Color.Wheat;

this.Controls.Add(But4);

}

public void OnBut(object sender, EventArgs e) // Обработчик нажатия на кнопку

{

Button but = (Button)sender;

try

{

switch (but.Name) // Выбор нажатия на кнопки

{

case "Prihod":

txt.Text = obj.Prihod(Convert.ToString(cmb.SelectedItem), cal.SelectionEnd);

if (txt.Text == null) txt.Text = "сервера кончились, подождите";

break;

case "Uhod":

txt.Text = obj.Uhod(Convert.ToString(cmb.SelectedItem));

if (txt.Text == null) txt.Text = "сервера кончились, подождите";

break;

case "Add":

string a = txt.Text;

string b = Convert.ToString(cmb.SelectedItem);

if (obj.Add\_serv(a,b))

txt.Text = "Сервер добавлен";

else

txt.Text = "Ошибка";

break;

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

public class MyClientSponsor : MarshalByRefObject, ISponsor

{

public MyClientSponsor()

{

Console.WriteLine("\nСпонсор создан ");

}

public TimeSpan Renewal(ILease lease) //продление жизни

{

if ((Client.Click\_time - DateTime.Now) < TimeSpan.FromSeconds(30))

return TimeSpan.FromSeconds(60);

else return TimeSpan.Zero;

}

}

Приложение 5. Client – Form2.Designer.cs

}

namespace Client

{

partial class Form2

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

this.Enter = new System.Windows.Forms.Button();

this.Login = new System.Windows.Forms.TextBox();

this.Pas = new System.Windows.Forms.TextBox();

this.label1 = new System.Windows.Forms.Label();

this.label2 = new System.Windows.Forms.Label();

this.SuspendLayout();

//

// Enter

//

this.Enter.Location = new System.Drawing.Point(204, 132);

this.Enter.Name = "Enter";

this.Enter.Size = new System.Drawing.Size(75, 23);

this.Enter.TabIndex = 0;

this.Enter.Text = "Enter";

this.Enter.UseVisualStyleBackColor = true;

this.Enter.Click += new System.EventHandler(this.Enter\_Click);

//

// Login

//

this.Login.Location = new System.Drawing.Point(69, 50);

this.Login.Name = "Login";

this.Login.Size = new System.Drawing.Size(100, 20);

this.Login.TabIndex = 1;

//

// Pas

//

this.Pas.Location = new System.Drawing.Point(69, 96);

this.Pas.Name = "Pas";

this.Pas.Size = new System.Drawing.Size(100, 20);

this.Pas.TabIndex = 2;

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(13, 50);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(38, 13);

this.label1.TabIndex = 3;

this.label1.Text = "Логин";

//

// label2

//

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(13, 102);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(45, 13);

this.label2.TabIndex = 4;

this.label2.Text = "Пароль";

//

// Form2

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(377, 207);

this.Controls.Add(this.label2);

this.Controls.Add(this.label1);

this.Controls.Add(this.Pas);

this.Controls.Add(this.Login);

this.Controls.Add(this.Enter);

this.Name = "Form2";

this.Text = "Form2";

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.Button Enter;

private System.Windows.Forms.TextBox Login;

private System.Windows.Forms.TextBox Pas;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Label label2;

}

}

Приложение 6. Client – Form2.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Security.Cryptography;

using OTCHET;

namespace Client

{

public partial class Form2 : Form

{

bool aft;

Interface1 obj;

public delegate void Form\_hide(bool aft);

public event Form\_hide form\_hide;

public Form2(Interface1 Obj,ref bool Aft)

{

InitializeComponent();

obj = Obj;

}

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

{

byte[] Login, Pas;

using (RSACryptoServiceProvider RSA = new RSACryptoServiceProvider())

{

RSA.ImportParameters(obj.Key\_send());

Login = RSA.Encrypt(Encoding.UTF8.GetBytes(this.Login.Text), false);

Pas = RSA.Encrypt(Encoding.UTF8.GetBytes(this.Pas.Text), false);

// string a = obj.Enter\_serv1(Login, Pas);

if (obj.Enter\_serv(Login, Pas))

form\_hide(true);

else form\_hide(false);

this.Hide();

}

}

}

Приложение 7. ASP.NET – WebForm1.aspx (Разметка)

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="Asp.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:LinkButton ID="Ent\_ad" runat="server" OnClientClick="window.open('WebForm2.aspx')" >Admin Enter</asp:LinkButton>

</div>

<asp:TextBox ID="txt" runat="server" Width="152px"></asp:TextBox>

<div style="margin-left:20px; display:inline"></div>

<asp:DropDownList ID="type\_serv" runat="server">

<asp:ListItem>igrovoi</asp:ListItem>

<asp:ListItem>Web server</asp:ListItem>

<asp:ListItem>Application server</asp:ListItem>

</asp:DropDownList>

<div style="margin-left:20px; display:inline"></div>

<asp:Button ID="Arend" runat="server" Text="Арендовать " OnClick="Arend\_Click" />

<div style="margin-left:20px; display:inline"></div>

<asp:Button ID="buy" runat="server" Text="Купить" OnClick="buy\_Click" />

<div style="align-items:center; display:inline"> <asp:Calendar ID="Cal" runat="server"></asp:Calendar> </div>

</form>

</body>

</html>

Приложение 8. ASP.NET – WebForm1.aspx.cs (Исполняемая часть)

using System;

using System.Collections.Generic;

using System.Runtime.Remoting.Lifetime;

using System.Runtime.Serialization.Formatters;

using System.Runtime.Remoting.Activation;

using System.Runtime.Remoting.Channels;

using System.Runtime.Remoting.Channels.Tcp;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using Obj;

namespace Asp

{

public partial class WebForm1 : System.Web.UI.Page

{

Obj.UObj obj;

ILease lease;

public WebForm1()

{

}

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Arend\_Click(object sender, EventArgs e)

{

using (obj = new UObj())

{

txt.Text = obj.Prihod(Convert.ToString(this.type\_serv.SelectedItem), this.Cal.SelectedDate);

if (txt.Text == null) txt.Text = "сервера кончились, подождите";

}

}

protected void buy\_Click(object sender, EventArgs e)

{

using (obj = new UObj())

{

txt.Text = obj.Uhod(Convert.ToString(this.type\_serv.SelectedItem));

if (txt.Text == null) txt.Text = "сервера кончились, подождите";

}

}

}

}

Приложение 9. ASP.NET – WebForm2.aspx (Разметка)

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="Asp.WebForm2" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

</div>

<asp:TextBox ID="Login" runat="server"></asp:TextBox>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Button ID="Ent" runat="server" OnClick="Ent\_Click" Text="Enter" />

<p>

<input id="Pass" runat="server" type="password" /></p>

</form>

</body>

</html>

Приложение 10. ASP.NET – WebForm2.aspx.cs (Исполняемая часть)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using Obj;

namespace Asp

{

public partial class WebForm2 : System.Web.UI.Page

{

UObj obj;

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Ent\_Click(object sender, EventArgs e)

{

using (obj = new UObj())

{

string a = Pass.Value;

if (obj.Enter\_serv(Login.Text, a))

Response.Redirect("WebForm3.aspx");

else Login.Text = "нельзя";

}

}

}

}

Приложение 11. ASP.NET – WebForm3.aspx (Разметка)

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm3.aspx.cs" Inherits="Asp.WebForm3" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:TextBox ID="IP" runat="server"></asp:TextBox>

<br />

<asp:DropDownList ID="Type" runat="server">

<asp:ListItem>igrovoi</asp:ListItem>

<asp:ListItem>Web server</asp:ListItem>

<asp:ListItem>Application server</asp:ListItem>

</asp:DropDownList>

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<asp:Button ID="Add" runat="server" OnClick="Add\_Click" Text="Add Server" />

</div>

</form>

</body>

</html>

Приложение 12. ASP.NET – WebForm3.aspx.cs (Исполняемая часть)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using Obj;

namespace Asp

{

public partial class WebForm3 : System.Web.UI.Page

{

UObj obj;

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Add\_Click(object sender, EventArgs e)

{

try

{

using (obj = new UObj())

{

if( obj.Add\_serv(IP.Text, Convert.ToString(Type.SelectedItem)))

IP.Text="сервер добавлен";

else

IP.Text = "ошибка";

}

}

catch { }

}

}

}