

ПРИЛОЖЕНИЕ А

ТЕКСТ ПРОГРАММЫ

АННОТАЦИЯ

В данном документе представлен исходный код информационной системы генерации краткосрочных пропускных карт «РЭУ им. Г.В. Плеханова»

Для разработки использовалась среда разработки Microsoft Visual Studio 2019.

Всего было разработано 13 программных модулей.

СОДЕРЖАНИЕ

1. ПЕРЕЧЕНЬ МОДУЛЕЙ.....	3
2. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ VIBOR_BD.CS.....	4
3. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ AVTORIZ.CS	7
4. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ REG_SOTR.CS.....	11
5. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ FORM.CS	15
6. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ APFNEL.CS.....	18
7. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ SPISOK_NA_PROHOD.CS	24
8. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ ADDZAIVKA.CS	29
9. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ SPISOK_POSETITELEI.CS	34
10. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ SPISOK_ZAIVKA.CS	37
11. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ PROCEDURE.CS	41
12. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ USING_BASE.CS.....	48

1. ПЕРЕЧЕНЬ МОДУЛЕЙ

В таблице ниже (Таблица 1.1 - Модули) представлен список авторских модулей и их описание.

Таблица 1.1 - Модули

Название модуля	Функция модуля
Vibor_BD.cs	Отвечает за выбор базы данных
Avtoriz.cs	Отвечает за авторизацию пользователей
Reg_sotr.cs	Отвечает за регистрацию пользователей
Form1.cs	Главная форма
APanel.cs	Панель администратора, где происходит управление пользователями и отслеживаются входы пользователей
Spisok_na_Prohod.cs	Обрабатываются заявки на проход и генерирует краткосрочную пропускную карту
AddZaivka.cs	Отвечает за создание заявки на проход и управлением
Spisok_posetitelei.cs	Отвечает за управление списком посетителей
Spisok_Zaivka.cs	Хранит все обработанные заявки
Procedure.cs	Класс в которые находятся методы используемые в программе
Using_Base.cs	Класс отвечает за подключение к базе данных

2. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ VIBOR_BD.CS

Ниже представлен код программного модуля Vibor_BD.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.Data.SqlClient;
using System.Data.Sql;
using Microsoft.SqlServer.Server;
using System.IO;

namespace Prohod
{
    public partial class Vibor_BD : Form
    {
        Using_Base _UB = new Using_Base();

        public Vibor_BD()
        {
            InitializeComponent();

            public void getInst()
            {
                try
                {
                    //Создание таблицы в которой хранятся все локальные сервера
                    using (DataTable SQLSources =
                        SqlDataSourceEnumerator.Instance.GetDataSources())
                    {
                        foreach (DataRow source in SQLSources.Rows) //цикл
предоставляющий строку
                        {
                            string instanceName =
                                source["InstanceName"].ToString();

                            if (!string.IsNullOrEmpty(instanceName))
                            {
                                //отображает список локальных серверов
                                comboBox1.Items.Add(source["ServerName"] + "\\\"
+ source["InstanceName"]);
                            }
                        }
                    }
                }
                catch (Exception ex)
                {
                    MessageBox.Show("Ошибка выбора сервера!");
                    //MessageBox.Show(ex.Message);
                }
            }
        }
    }
}
```

```
    }

}

private void Vibor_BD_Load(object sender, EventArgs e)
{
    getInst();
}

private void Button1_Click(object sender, EventArgs e)
{
    try
    {
        if (comboBox1.Text == null || textBox1.Text == null ||
textBox2.Text == null)
        {
            MessageBox.Show("Выберите сервер");
        }
        else
        {
            _UB.bdhost = comboBox1.Text;
            _UB.bdlog = textBox1.Text;
            _UB.bdpass = textBox2.Text;
            _UB.bd = comboBox2.Text;
            SqlConnection dbconn = new SqlConnection("Data Source="
+ comboBox1.Text + ";User ID=" + textBox1.Text + ";Password=" +
textBox2.Text + "");
            dbconn.Open();
            /*отображение списка баз данных исходя из введенных
данных в поля указанные выше*/
            SqlCommand getDb = new SqlCommand("SELECT name FROM
master.dbo.sysdatabases;", dbconn);
            SqlDataReader reader = getDb.ExecuteReader();

            if (reader.HasRows) // если есть данные
            {
                comboBox2.Enabled = true;
                button2.Enabled = true;
                // выводим названия столбцов
                while (reader.Read()) // построчно считываем данные
                {
                    comboBox2.Items.Add(reader.GetString(0));
                }
            }
            dbconn.Close();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Выберите сервер");
        //MessageBox.Show(ex.Message);
    }
}
```

```
private void Button2_Click(object sender, EventArgs e)
{
    try
    {
        if (comboBox2.Text == null)
        {
            MessageBox.Show("Выберите источник данных");
        }
        else
        {
            _UB.bd = comboBox2.Text;
            _UB.SetConn();
            Avtoriz _A = new Avtoriz();
            this.Hide();
            _A.Show();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Выберите источник данных");
        //MessageBox.Show(ex.Message);
    }
}

private void PictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    //если кнопка нажата пароль виден
    textBox2.UseSystemPasswordChar = false; // пароль раскрыт
    pictureBox1.BackgroundImage =
Properties.Resources.Пароль_виден;
}

private void PictureBox1_MouseUp(object sender, MouseEventArgs e)
{
    //если кнопка отпущена пароль не виден
    textBox2.UseSystemPasswordChar = true; // пароль раскрыт
    pictureBox1.BackgroundImage =
Properties.Resources.Пароль_не_виден;
}

private void Button3_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void Vibor_BD_FormClosed(object sender, FormClosedEventArgs
e)
{
    Application.Exit();
}
}
```

3. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ AVTORIZ.CS

Ниже представлен код программного модуля Avtoriz.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.Data.SqlClient;
using System.Data.Sql;
using Microsoft.SqlServer.Server;
using System.IO;

namespace Prohod
{
    public partial class Avtoriz : Form
    {
        Using_Base _UB = new Using_Base();
        Form1 _F = new Form1();
        Procedure _P = new Procedure();
        DataSet ds = new DataSet();
        ToolTip _T = new ToolTip();

        public Avtoriz()
        {
            InitializeComponent();
        }

        private void Avtoriz_FormClosed(object sender, FormClosedEventArgs
e)
        {
            Application.Exit();
            GC.Collect();
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            try
            {
                _UB.SQLconnect.Open();
                /*переменные которые показывают ФИО сотрудника, роль
сотрудника и его ID на основе написанного логина в textBox*/
                SqlCommand FIO_S = new SqlCommand("SELECT [FIO_S] FROM
[DBO].[Sotr] WHERE [Log_S]='\" + textBox1.Text + \"'", _UB.SQLconnect);
                SqlCommand Role = new SqlCommand("SELECT [Rolle] FROM
[DBO].[Rolli] WHERE [ID_role]=(SELECT [Role_ID] FROM [DBO].[Sotr] WHERE
[Log_S]='\" + textBox1.Text + \"')", _UB.SQLconnect);
```

```
SqlCommand ID = new SqlCommand("SELECT [ID_sotr] FROM
[DBO].[Sotr] WHERE [Log_S]='" + textBox1.Text + "'", _UB.SQLconnect);
/*Проверка всех логинов*/
SqlCommand getAcc = new SqlCommand("SELECT * FROM
[DBO].[Sotr] WHERE [Log_S]='" + textBox1.Text + "'", _UB.SQLconnect);
var Acc = getAcc.ExecuteScalar();
_UB.SQLconnect.Close();
if (Acc != null)
{
    _UB.SQLconnect.Open();
    /*Проверка пароля на основании логина*/
    SqlCommand getPass = new SqlCommand("SELECT [Pass_S]
FROM [DBO].[Sotr] WHERE [Log_S]='" + textBox1.Text + "'", _UB.SQLconnect);
    string password = getPass.ExecuteScalar().ToString();

    if (password == textBox2.Text)
    {
        /*Вывод сообщения с ФИО и ролью авторизованного
пользователя*/
        MessageBox.Show("Вы вошли как: " +
FIO_S.ExecuteScalar().ToString() + ", Роль: " +
Role.ExecuteScalar().ToString());
        Program.IsAdmin = Role.ExecuteScalar().ToString();
        _F.label1.Text = FIO_S.ExecuteScalar().ToString();
        _F.label2.Text = Role.ExecuteScalar().ToString();

        Properties.Settings.Default.L1 =
_F.label1.Text;//запись в системную переменную
        Properties.Settings.Default.L2 =
_F.label2.Text;//запись в системную переменную
        Properties.Settings.Default.Rol =
Program.IsAdmin;//запись в системную переменную
        Properties.Settings.Default.Save(); // Сохранение
системных переменных

        /*Добавление времени авторизации и сотрудника с
базу данных*/

        _P.Time_vhod_add(Convert.ToInt32(ID.ExecuteScalar().ToString()),
FIO_S.ExecuteScalar().ToString(), toolStripLabel1.Text);

        this.Hide();
        _F.Show();
    }
    else
    {
        MessageBox.Show("Неверный пароль!");
    }
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
    //MessageBox.Show("Неверный логин или пароль");
}

_UB.SQLconnect.Close();
```



```
}

private void Button3_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void Button2_Click(object sender, EventArgs e)
{
    Reg_Sotr _RG = new Reg_Sotr();
    this.Hide();
    _RG.Show();
}

private void PictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    textBox2.PasswordChar = '\0';
    textBox2.UseSystemPasswordChar = false; // пароль раскрыт
    pictureBox1.BackgroundImage =
Properties.Resources.Пароль_виден;
}

private void PictureBox1_MouseUp(object sender, MouseEventArgs e)
{
    textBox2.UseSystemPasswordChar = true; // пароль раскрыт
    pictureBox1.BackgroundImage =
Properties.Resources.Пароль_не_виден;
}

private void Timer1_Tick(object sender, EventArgs e)
{
    toolStripLabel1.Text = DateTime.Now.ToString("MM/dd/yy
H:mm:ss");
}

private void Button2_MouseMove(object sender, MouseEventArgs e)
{
    _T.SetToolTip(button2, "Если вы не зарегистрированы
\нпожалуйста нажмите сюда!");
}

private void TextBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    /*можно водить только те символы котрые указаны в кавычках*/
    if ((e.KeyChar >= 'A' && e.KeyChar <= 'Z') || (e.KeyChar >= 'a'
&& e.KeyChar <= 'z') || e.KeyChar == (char)Keys.Back)
    {
    }
    else
    {
        e.Handled = true;
    }
}

private void TextBox2_KeyPress(object sender, KeyPressEventArgs e)
```

```
        { /*можно водить только те символы котрые указаны в кавычках*/
            if ((e.KeyChar >= 'A' && e.KeyChar <= 'Z') || (e.KeyChar >= 'a'
&& e.KeyChar <= 'z') || (e.KeyChar >= '0' && e.KeyChar <= '9') || e.KeyChar
== (char)Keys.Back)
            {
            }
            else
            {
                e.Handled = true;
            }
        }
    }
}
```

4. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ REG_SOTR.CS

Ниже представлен код программного модуля Reg_Sotr.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.Data.SqlClient;
using System.Data.Sql;
using Microsoft.SqlServer.Server;
using System.IO;

namespace Prohod
{
    public partial class Reg_Sotr : Form
    {
        Procedure _P = new Procedure(); //класс методов
        ToolTip _T = new ToolTip();
        Using_Base _UB = new Using_Base(); //подключение к БД
        Shifr _Sh = new Shifr(); //шифрования

        public Reg_Sotr()
        {
            InitializeComponent();
            textBox5.MaxLength = 6;
            textBox6.MaxLength = 6;
            comboBox1.KeyPress += (sender, e) => e.Handled = true; //запрет
на ввод символов
        }

        private void Button2_Click(object sender, EventArgs e)
        {
            Avtoriz _A = new Avtoriz();
            this.Hide();
            _A.Show();
        }

        private void Reg_Sotr_FormClosed(object sender, FormClosedEventArgs
e)
        {
            Application.Exit();
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            try
            {
                if (textBox5.Text == textBox6.Text)
```

```
        {
            if (textBox1.TextLength == 0 || textBox4.TextLength ==
0 || textBox5.TextLength == 0 || comboBox1.Text == null ||
textBox7.TextLength == 0)
            {
                MessageBox.Show("Занесите все данные!!!");
            }
            else
            {
                /*Добавление нового сотрудника*/
                _P.Sotr_add(textBox1.Text, textBox4.Text,
textBox5.Text, comboBox1.SelectedIndex + 1, textBox7.Text);
                MessageBox.Show("Вы зарегистрированы, \nпожалуйста
не забудьте введённые данные!");
                Avtoriz _A = new Avtoriz();
                this.Hide();
                _A.Show();
            }
        }
        else
        {
            MessageBox.Show("Пароли не совпадают!");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
        //MessageBox.Show("Пользователь с таким логином уже
существует!");
    }
}

private void Reg_Sotr_Load(object sender, EventArgs e)
{
    _P.Rol_Load(comboBox1); //вывод ролей
}

private void Label8_MouseMove(object sender, MouseEventArgs e)
{
    _T.SetToolTip(label8, "Укажите отдел, факультет или место \нк
которому вы принадлежите");
}

private void PictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    textBox5.UseSystemPasswordChar = false; // пароль раскрыт
    textBox6.UseSystemPasswordChar = false; // паротль раскрыт
    pictureBox1.BackgroundImage =
Properties.Resources.Пароль_виден;
}

private void PictureBox1_MouseUp(object sender, MouseEventArgs e)
{
    textBox5.UseSystemPasswordChar = true; // пароль раскрыт
```

```
        textBox6.UseSystemPasswordChar = true;// пароль раскрыт
        pictureBox1.BackgroundImage =
Properties.Resources.Пароль_не_виден;
    }

    private void TextBox1_KeyPress(object sender, KeyPressEventArgs e)
    { /*можно водить только те символы котрые указаны в кавычках*/
        if ((e.KeyChar >= 'А' && e.KeyChar <= 'Я') || (e.KeyChar >= 'а'
&& e.KeyChar <= 'я') || e.KeyChar == '_' || e.KeyChar ==
(char)Keys.CapsLock || e.KeyChar == (char)Keys.Space || e.KeyChar ==
(char)Keys.Back)
        {
        }
        else
        {
            e.Handled = true;
        }
    }

    private void TextBox4_KeyPress(object sender, KeyPressEventArgs e)
    { /*можно водить только те символы котрые указаны в кавычках*/
        if ((e.KeyChar >= 'А' && e.KeyChar <= 'Z') || (e.KeyChar >= 'а'
&& e.KeyChar <= 'z') || e.KeyChar == (char)Keys.Back)
        {
        }
        else
        {
            e.Handled = true;
        }
    }

    private void TextBox5_KeyPress(object sender, KeyPressEventArgs e)
    { /*можно водить только те символы котрые указаны в кавычках*/
        if ((e.KeyChar >= 'А' && e.KeyChar <= 'Z') || (e.KeyChar >= 'а'
&& e.KeyChar <= 'z') || (e.KeyChar >= '0' && e.KeyChar <= '9') || e.KeyChar
== (char)Keys.Back)
        {
        }
        else
        {
            e.Handled = true;
        }
    }

    private void TextBox7_KeyPress(object sender, KeyPressEventArgs e)
    { /*можно водить только те символы котрые указаны в кавычках*/
        if ((e.KeyChar >= 'А' && e.KeyChar <= 'Я') || (e.KeyChar >= 'а'
&& e.KeyChar <= 'я') || (e.KeyChar >= '0' && e.KeyChar <= '9') ||
e.KeyChar == '_' || e.KeyChar == ',' || e.KeyChar ==
(char)Keys.Back)
        {
        }
        else
        {

```

```
        e.Handled = true;
    }
}
}
```

5. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ FORM.CS

Ниже представлен код программного модуля Form.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;

namespace Prohod
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        { /*Разграничение прав на основе роли содержащейся в label2*/
            switch (label2.Text)
            {
                case "Администратор"           ":
                {
                    //button1.Visible = false;
                    break;
                }
                case "Оператор"                 ":
                {
                    button1.Visible = false;
                    button4.Visible = false;
                    break;
                }
                case "Декан"                    ":
                {
                    button1.Visible = false;
                    button5.Visible = false;
                    button4.Visible = false;
                    break;
                }
            }
        }

        private void Form1_FormClosed(object sender, FormClosedEventArgs e)
        {
            Application.Exit(); //закрытие программы
        }

        private void Button3_Click(object sender, EventArgs e)
```

```
    { /*Переход на форму авторизации*/  
        Avtoriz _A = new Avtoriz();  
        this.Hide();  
        Properties.Settings.Default.Reset(); //переменные по умолчанию  
        _A.Show();  
    }  
  
    private void Button1_Click(object sender, EventArgs e)  
    { /*переход на форму создания заявки*/  
        AddZaivka _AZ = new AddZaivka();  
        Properties.Settings.Default.L1 = label1.Text;  
        Properties.Settings.Default.L2 = label2.Text;  
        Properties.Settings.Default.Save(); // Сохранение системных  
переменных  
        this.Hide();  
        _AZ.Show();  
    }  
  
    private void Button4_Click(object sender, EventArgs e)  
    { /*переход на панель администратора*/  
        APanel _AP = new APanel();  
        Properties.Settings.Default.L1 = label1.Text;  
        Properties.Settings.Default.L2 = label2.Text;  
        Properties.Settings.Default.Save(); // Сохранение системных  
переменных  
        this.Hide();  
        _AP.Show();  
    }  
  
    private void Button2_Click(object sender, EventArgs e)  
    { /*переходи на форму список заявок на проход*/  
        Spisok_na_Prohod _SP = new Spisok_na_Prohod();  
        Properties.Settings.Default.L1 = label1.Text;  
        Properties.Settings.Default.L2 = label2.Text;  
        Properties.Settings.Default.Save(); // Сохранение системных  
переменных  
        this.Hide();  
        _SP.Show();  
    }  
  
    private void Button5_Click(object sender, EventArgs e)  
    { /*переход на форму список оформленных заявок*/  
        Spisok_Zaivka _SZ = new Spisok_Zaivka();  
        Properties.Settings.Default.L1 = label1.Text;  
        Properties.Settings.Default.L2 = label2.Text;  
        Properties.Settings.Default.Save(); // Сохранение системных  
переменных  
        this.Hide();  
        _SZ.Show();  
    }  
  
    private void Button6_Click(object sender, EventArgs e)  
    {  
        Application.Exit(); //закрытие программы  
    }
```


}
}
}

6. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ APANEL.CS

Ниже представлен код программного модуля APanel.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.Data.SqlClient;
using System.Data.Sql;
using System.Diagnostics;

namespace Prohod
{
    public partial class APanel : Form
    {
        Procedure _P = new Procedure(); //класс методов
        ToolTip _T = new ToolTip(); // всплывающие подсказки

        public APanel()
        {
            InitializeComponent();
            textBox5.MaxLength = 6;
            /*если выбрать dropdownlist то в combobox не будут показываться
            данные при событии DataGridView_CellContentClick*/
            comboBox1.KeyPress += (sender, e) => e.Handled = true; //запрет
            на ввод символов
        }

        private void APanel_Load(object sender, EventArgs e)
        {
            textBox5.MaxLength = 6;
            //Grid_Load();

            Sotr_Rolli(); //таблицы сотрудников
            Grid_Load3(); //таблица авторизаций
            _P.Rol_Load(comboBox1); //роли
        }

        private void APanel_FormClosed(object sender, FormClosedEventArgs
e)
        {
            Form1 _F = new Form1(); // открытие главной формы
            _F.label1.Text = Properties.Settings.Default.L1;
            _F.label2.Text = Properties.Settings.Default.L2;
            this.Hide();
            _F.Show();
        }
    }
}
```

```
//public void Grid_Load()
//{
//    _P.Sotr_void();
//    dataGridView1.DataSource = Program.Sotr_Select;
//    //dataGridView1.Columns[0].Visible = false;
//    dataGridView1.Columns[0].HeaderText = "Номер сотрудника";
//    dataGridView1.Columns[1].HeaderText = "Фамилия";
//    dataGridView1.Columns[2].HeaderText = "Имя";
//    dataGridView1.Columns[3].HeaderText = "Отчество";
//    dataGridView1.Columns[4].HeaderText = "Логин";
//    dataGridView1.Columns[5].HeaderText = "Пароль";
//    dataGridView1.Columns[6].HeaderText = "Роль";
//    dataGridView1.Columns[7].HeaderText = "Дополнительная
информация";
//}

public void Grid_Load3()
{ /*Таблица главной формы*/
    _P.Time_vhod_void();
    dataGridView2.DataSource = Program.Time_vhod_Select;
    dataGridView2.Columns[0].Visible = false;
    dataGridView2.Columns[1].HeaderText = "Номер сотрудника";
    dataGridView2.Columns[2].HeaderText = "Сотрудник";
    dataGridView2.Columns[3].HeaderText = "Время авторизации";
    dataGridView2.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill;
}

private void Button1_Click(object sender, EventArgs e)
{ /*добавление новой записи, если одно поле пустое - ошибка*/
    try
    {
        if (textBox1.TextLength == 0 || textBox4.TextLength ==
0 || textBox5.TextLength == 0 || comboBox1.Text == null ||
textBox7.TextLength == 0)
        {
            MessageBox.Show("Занесите все данные!!!");
        }
        else
        {
            _P.Sotr_add(textBox1.Text, textBox4.Text,
textBox5.Text, comboBox1.SelectedIndex + 1, textBox7.Text);
            MessageBox.Show("Сотрудник добавлен");
            //Grid_Load();
            Sotr_Rolli();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Пользователь с таким логином уже
существует!");
    }
}
```

```
private void DataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    /*При выборе поля в datagridview данные дублируются в полях
    ввода*/
    textBox1.Text =
    dataGridView1.CurrentRow.Cells[1].Value.ToString();
    textBox4.Text =
    dataGridView1.CurrentRow.Cells[2].Value.ToString();
    textBox5.Text =
    dataGridView1.CurrentRow.Cells[3].Value.ToString();
    comboBox1.Text =
    dataGridView1.CurrentRow.Cells[4].Value.ToString();
    textBox7.Text =
    dataGridView1.CurrentRow.Cells[5].Value.ToString();
}

private void Button3_Click(object sender, EventArgs e)
{
    /*Удаление выбранного поля по id*/
    try
    {
        Using_Base _UB = new Using_Base();
        _UB.SQLconnect.Open();
        SqlCommand del = new SqlCommand("[DBO].[sotr_delete]",
        _UB.SQLconnect);
        del.CommandType = CommandType.StoredProcedure;
        SqlParameter id = new SqlParameter()
        {
            ParameterName = "@ID_sotr",
            Value = dataGridView1.CurrentRow.Cells[0].Value
        };
        del.Parameters.Add(id);
        del.ExecuteNonQuery();
        _UB.SQLconnect.Close();
        //Grid_Load();
        Sotr_Rolli();
        MessageBox.Show("Сотрудник удалён");
    }
    catch (Exception ex)
    {
        //MessageBox.Show(ex.Message);
        MessageBox.Show("Данный сотрудник используется в другой
таблице");
    }
}

private void Button2_Click(object sender, EventArgs e)
{
    /*команда обновления полей в таблице*/
    try
    {
        Using_Base _UB = new Using_Base();
        _UB.SQLconnect.Open(); /*Обновление поля*/
        SqlCommand UP = new SqlCommand("UPDATE Sotr SET " +
        "FIO_S = '" + textBox1.Text + "' , " +
        "Log_S = '" + textBox4.Text + "' , " +
```

```
        "Pass_S = '" + textBox5.Text + "' , " +
        "Role_ID = '" + (comboBox1.SelectedIndex + 1) + "' , " +
        "Dop_Inf = '" + textBox7.Text + "' WHERE ID_sotr = " +
dataGridView1[0, dataGridView1.CurrentRow.Index].Value, _UB.SQLconnect);
        UP.ExecuteNonQuery();
        _UB.SQLconnect.Close();
        //Grid_Load();
        Sotr_Rolli();
        MessageBox.Show("Обновлено");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void Button7_Click(object sender, EventArgs e)
{
    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        dataGridView1.Rows[i].Selected = false; //Фокус убирается
        for (int j = 0; j < dataGridView1.ColumnCount; j++)
            if (dataGridView1.Rows[i].Cells[j].Value != null) //если
таблица не равно нулю
            {
                if
                (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox8.Text)) //
ищет поле по значению, которое написано в textBox8
                {
                    dataGridView1.Rows[i].Selected = true; //если
найдено тогда выделяется синим
                    break;
                }
            }
    }
}

private void PictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    textBox5.UseSystemPasswordChar = false; // пароль раскрыт
    pictureBox1.BackgroundImage =
Properties.Resources.Пароль_виден;
}

private void PictureBox1_MouseUp(object sender, MouseEventArgs e)
{
    textBox5.UseSystemPasswordChar = true; // пароль раскрыт
    pictureBox1.BackgroundImage =
Properties.Resources.Пароль_не_виден;
}

private void Sotr_Rolli()
{
    try
    {
        Using_Base _UB = new Using_Base();
```

```
        _UB.SQLconnect.Open();
        /*вывод виртуальной таблицы*/
        SqlCommand _Sotr_Rolli = new SqlCommand("SELECT
Sotr.ID_Sotr, Sotr.FIO_S, Sotr.Log_S, Sotr.Pass_S, Rolli.Rolle,
Sotr.Dop_Inf " +
            "FROM [DBO].[Sotr], Rolli WHERE Sotr.Role_ID =
Rolli.ID_role", _UB.SQLconnect);
        SqlDataReader SR = _Sotr_Rolli.ExecuteReader();

        DataTable _S = new DataTable();
        _S.Load(SR);
        dataGridView1.DataSource = _S;
        //dataGridView1.Columns[0].Visible = false; /*Название
таблиц*/

        dataGridView1.Columns[0].HeaderText = "Номер сотрудника";
        dataGridView1.Columns[1].HeaderText = "Сотрудник";
        dataGridView1.Columns[2].HeaderText = "Логин";
        //dataGridView1.Columns[3].HeaderText = "Пароль";
        dataGridView1.Columns[3].Visible = false;
        dataGridView1.Columns[4].HeaderText = "Роль";
        dataGridView1.Columns[5].HeaderText = "Дополнительная
информация";

        dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill; //выравнивание таблицы по всей ширине
        _UB.SQLconnect.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void TextBox7_MouseMove(object sender, MouseEventArgs e)
{
    _T.SetToolTip(textBox7, "Укажите факультет или место, \n где Вы
работаете");
}

private void TextBox7_Enter(object sender, EventArgs e)
{
    _T.SetToolTip(textBox7, "Укажите факультет или место, \n где Вы
работаете");
}

private void Button9_Click(object sender, EventArgs e)
{
    Using_Base _UB = new Using_Base();
    _UB.SQLconnect.Open();
    SqlCommand del = new SqlCommand("[DBO].[Time_vhod_delete]",
_UB.SQLconnect); //создание переменной которая хранит таблицу
    del.CommandType = CommandType.StoredProcedure; //интерпритация
переменной как хранимая процедура
    try
    {
```

```
        SqlParameter id = new SqlParameter()// параметр id
        {
            ParameterName = "@ID_vhod",//выбор параметра
            Value = dataGridView2.SelectedRows.Count//задаёт
значение выбранной ячейки
        };
        del.Parameters.Add(id);//добавляет параметр
        del.ExecuteNonQuery();
        _UB.SQLconnect.Close();
        //Grid_Load3();
        Sotr_Rolli();
        MessageBox.Show("Запись удалена");
    }
    catch (Exception ex)
    {
        MessageBox.Show("Вы ничего не выбрали");
        //MessageBox.Show(ex.Message);
    }
}

private void TextBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    /*можно водить только те символы котрые указаны в кавычках*/
    if ((e.KeyChar >= 'А' && e.KeyChar <= 'Я') || (e.KeyChar >=
'a' && e.KeyChar <= 'я') || e.KeyChar == '_' || e.KeyChar ==
(char)Keys.CapsLock || e.KeyChar == (char)Keys.Space || e.KeyChar ==
(char)Keys.Back)
    {
    }
    else
    {
        e.Handled = true;
    }
}
}
```

7. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ

SPIKOK_NA_PROHOD.CS

Ниже представлен код программного модуля Spisok_na_Prohod.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.Data.SqlClient;
using System.Data.Sql;
using System.Diagnostics;
using Word1 = Microsoft.Office.Interop.Word;
using Microsoft.Office;

namespace Prohod
{
    public partial class Spisok_na_Prohod : Form
    {
        Procedure _P = new Procedure();//класс методов
        Shifr _Sh = new Shifr();//класс шифрования
        Tooltip _T = new Tooltip();//подсказка
        Tooltip _T2 = new Tooltip();//вторая подсказка
        Word _W = new Word();//класс который генерирует таблицу
        Word1.Application oWord = new Word1.Application();

        public Spisok_na_Prohod()
        {
            InitializeComponent();
            /*если выбрать dropdownlist то в combobox не будут показываться
            данные при событии DataGridView_CellContentClick*/
            comboBox1.KeyPress += (sender, e) => e.Handled = true;//запрет
            на ввод символов
            comboBox2.KeyPress += (sender, e) => e.Handled = true;//запрет
            на ввод символовma
            maskedTextBox1.KeyPress += (sender, e) => e.Handled =
            true;//запрет на ввод символовma
            dateTimePicker1.KeyPress += (sender, e) => e.Handled =
            true;//запрет на ввод символовma
            dateTimePicker2.KeyPress += (sender, e) => e.Handled =
            true;//запрет на ввод символовma
            textBox9.KeyPress += (sender, e) => e.Handled = true;//запрет
            на ввод символовma
            maskedTextBox3.KeyPress += (sender, e) => e.Handled =
            true;//запрет на ввод символовma
        }

        //private void Grid_Load()
```



```
//{
//    _P.Zaivka_void();
//    dataGridView1.DataSource = Program.Zaivka_Select;
//    dataGridView1.Columns[0].Visible = false;
//    dataGridView1.Columns[1].HeaderText = "Сотрудник";
//    dataGridView1.Columns[2].HeaderText = "Дата оформления";
//    dataGridView1.Columns[3].HeaderText = "Посетитель";
//    dataGridView1.Columns[4].HeaderText = "Серия и номер";
//    dataGridView1.Columns[5].HeaderText = "Дата прихода";
//    dataGridView1.Columns[6].HeaderText = "Время прихода";
//    dataGridView1.Columns[7].HeaderText = "Место прихода";
//    dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill;
//}

private void Zaivka_load()
{
    Using_Base _UB = new Using_Base();
    _UB.SQLconnect.Open(); //открытие подключение
    /*Создание виртуальной таблицы */
    SqlCommand _Zaivka_Pos = new SqlCommand("SELECT
Zaivka.ID_zaiivka, Sotr.FIO_S, Zaivka.Data_oform, Posetitel.FIO_P,
Zaivka.Nomer_and_Seria_Pasport, Zaivka.Data_prihoda, Zaivka.Vremi_prihoda,
Zaivka.Mesto_prihoda FROM [DBO].[Zaivka], [DBO].[Posetitel], [DBO].[Sotr]
WHERE Zaivka.Sotr_ID = Sotr.ID_Sotr and Zaivka.Posetitel_ID =
Posetitel.ID_Posetitel", _UB.SQLconnect);
    SqlDataReader ZP = _Zaivka_Pos.ExecuteReader();

    DataTable _Z = new DataTable();

    _Z.Load(ZP);
    dataGridView1.DataSource = _Z;
    dataGridView1.Columns[0].Visible = false;
    //dataGridView1.Columns[0].HeaderText = "Номер сотрудника";
    dataGridView1.Columns[1].HeaderText = "Сотрудник";
    dataGridView1.Columns[2].HeaderText = "Дата оформления";
    dataGridView1.Columns[3].HeaderText = "Посетитель";
    dataGridView1.Columns[4].HeaderText = "Номер и серия паспорта";
    dataGridView1.Columns[5].HeaderText = "Дата прихода";
    dataGridView1.Columns[6].HeaderText = "Время прихода";
    dataGridView1.Columns[7].HeaderText = "Место прихода";
    dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill;
    _UB.SQLconnect.Close();
}

private void Spisok_na_Prohod_Load(object sender, EventArgs e)
{
    //Grid_Load();
    Zaivka_load(); //таблица
    _P.Sotr_Load(comboBox2); //вывод списка сотрудников
    _P.Pos_Load(comboBox1); //вывод списка посетителей
}
```

```
private void Spisok_na_Prohod_FormClosed(object sender,
FormClosedEventArgs e)
{
    Form1 _F = new Form1(); //открытие главной формы
    _F.label1.Text = Properties.Settings.Default.L1;
    _F.label2.Text = Properties.Settings.Default.L2;
    this.Hide();
    _F.Show();
}

private void DataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    //событие которое переносит данные выбранного поля в поля ввода
    maskedTextBox1.Text =
dataGridView1.CurrentRow.Cells[2].Value.ToString();
    comboBox1.Text =
dataGridView1.CurrentRow.Cells[3].Value.ToString();
    maskedTextBox3.Text =
dataGridView1.CurrentRow.Cells[4].Value.ToString();
    dateTimePicker1.Text =
dataGridView1.CurrentRow.Cells[5].Value.ToString();
    dateTimePicker2.Text =
dataGridView1.CurrentRow.Cells[6].Value.ToString();
    textBox9.Text =
dataGridView1.CurrentRow.Cells[7].Value.ToString();
    comboBox2.Text =
dataGridView1.CurrentRow.Cells[1].Value.ToString();
    maskedTextBox3.Text = _Sh.Codec(maskedTextBox3.Text, -5/*-
(int)numericUpDown1.Value*/);
}

private void Button1_Click(object sender, EventArgs e)
{
    /*происходит оформление заявки на проход*/
    try
    {
        /*Добавляется информация в таблицы из полей*/
        _P.Ozaivka_add(textBox2.Text, maskedTextBox1.Text,
comboBox1.SelectedIndex + 1, maskedTextBox3.Text =
_Sh.Codec(maskedTextBox3.Text, 5/*Смещение*/), dateTimePicker1.Text,
dateTimePicker2.Text, textBox9.Text, comboBox2.SelectedIndex + 1);
        /*Действующая заявка на проход удаляется*/
        Using_Base _UB = new Using_Base();
        _UB.SQLconnect.Open();
        /*команда хранит хранимую процедуру на удаление заявки*/
        SqlCommand del = new
SqlCommand("[DBO].[Zaivka_delete]", _UB.SQLconnect);
        del.CommandType = CommandType.StoredProcedure;
        SqlParameter id = new SqlParameter() //перечисление
параметров
        {
            ParameterName = "@ID_zaiivka",
            Value = dataGridView1.CurrentRow.Cells[0].Value
        };
        del.Parameters.Add(id);
        del.ExecuteNonQuery();
    }
}
```

```
        _UB.SQLconnect.Close();
        MessageBox.Show("Оформлено");
        //Grid_Load();
        Zaivka_load();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void Button3_Click(object sender, EventArgs e)
{
    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        dataGridView1.Rows[i].Selected = false; //все выделения
убираются
        for (int j = 0; j < dataGridView1.ColumnCount; j++)
            if (dataGridView1.Rows[i].Cells[j].Value != null) //если
таблица е пуста
            {
                if
(dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text)) //
если данные записанные в поле есть
                {
                    dataGridView1.Rows[i].Selected = true; //Тогда
происходит выделение
                    break;
                }
            }
    }

    private void PictureBox2_MouseMove(object sender, MouseEventArgs e)
    {
        _T2.SetToolTip(pictureBox2, "Распечатать краткосрочную
карту"); //всплывающая подсказка
    }

    private void PictureBox1_Click(object sender, EventArgs e)
    {
        try
        {
            Word _WD = new Word(); //класс, который генерирует таблицу в
документ word
            SaveFileDialog sfd = new SaveFileDialog(); //создание
компонента SaveFileDialog

            sfd.Filter = "Word Documents (*.docx)|*.docx"; //расширение
файла

            sfd.FileName = "Список.docx"; //имя файла

            if (sfd.ShowDialog() == DialogResult.OK) //окно сохранения
            {
                _WD.MS_Export_Table(dataGridView1, sfd.FileName);
            }
        }
    }
}
```

```
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void PictureBox2_Click(object sender, EventArgs e)
{
    if (comboBox2.Text == null || comboBox1.Text == null)
    {
        MessageBox.Show("Заполните данные");
    }
    else
    {
        try
        {
            Word1.Application app = new
Word1.Application(); //создание документа
            Word1.Document doc =
app.Documents.Add("C:\\Users\\petya\\Desktop\\Шаблон.docx"); //выбор шаблона
/*для того что бы поля записались надо добавить
закладки в меню вставка>закладка*/
            doc.Bookmarks["sotr"].Range.Text = comboBox2.Text;
            doc.Bookmarks["pos"].Range.Text = comboBox1.Text;
            doc.Bookmarks["date"].Range.Text =
DateTime.Now.ToString("dd.MM.yy");
            doc.SaveAs(FileName:
"C:\\Users\\petya\\Desktop\\Краткосрочная карта.doc"); //сохранение
документа

            app.Visible = true; //открытие
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

private void PictureBox1_MouseMove(object sender, MouseEventArgs e)
{
    /*подсказка*/
    _T.SetToolTip(pictureBox1, "Распечатать выделенные поля");
}
}
```

8. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ ADDZAIVKA.CS

Ниже представлен код программного модуля AddZaivka.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.Data.SqlClient;
using System.Data.Sql;
using System.Diagnostics;

namespace Prohod
{
    public partial class AddZaivka : Form
    {
        Procedure _P = new Procedure(); //класс методов
        Shifr _Sh = new Shifr(); //класс шифрования
        Using_Base _UB = new Using_Base(); //класс подключения к БД

        public AddZaivka()
        {
            InitializeComponent();
            /*если выбрать dropdownlist то в combobox не будут показываться
            данные при событии DataGridView_CellContentClick*/
            comboBox1.KeyPress += (sender, e) => e.Handled = true; //запрет
            на ввод символов
            comboBox2.KeyPress += (sender, e) => e.Handled = true; //запрет
            на ввод символов
            maskedTextBox1.KeyPress += (sender, e) => e.Handled =
            true; //запрет на ввод символов
        }

        private void AddZaivka_FormClosed(object sender,
        FormClosedEventArgs e)
        {
            Form1 _F = new Form1(); //главное окно
            _F.label1.Text = Properties.Settings.Default.L1;
            _F.label2.Text = Properties.Settings.Default.L2;
            this.Hide();
            _F.Show();
        }

        //private void Grid_Load()
        //{
        //    _P.Zaivka_void();
        //    dataGridView1.DataSource = Program.Zaivka_Select;
        //    dataGridView1.Columns[0].Visible = false;
        //    dataGridView1.Columns[1].HeaderText = "Сотрудник";
        //}
```

```
//      dataGridView1.Columns[2].HeaderText = "Дата оформления";
//      dataGridView1.Columns[3].HeaderText = "Посетитель";
//      dataGridView1.Columns[4].HeaderText = "Серия и номер
паспорта";
//      dataGridView1.Columns[5].HeaderText = "Дата прихода";
//      dataGridView1.Columns[6].HeaderText = "Время прихода";
//      dataGridView1.Columns[7].HeaderText = "Место прихода";
//      dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill;
//}

private void AddZaivka_Load(object sender, EventArgs e)
{
    //Grid_Load();
    Zaivka_Load();//таблица заявок
    _P.Sotr_Load(comboBox2);//список сотрудников
    _P.Pos_Load(comboBox1);//список посетителей
}

private void Button4_Click(object sender, EventArgs e)
{
    /*при открытии окна посетителей окно заявок не закрывается*/
    Spisok_posetitelei _SP = new Spisok_posetitelei();//окно список
    посетителей
    _SP.Show();
}

private void DataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    /*При выборе поля в datagridview данные дублируются в полях ввода
    */
    comboBox2.Text =
dataGridView1.CurrentRow.Cells[1].Value.ToString();
    maskedTextBox1.Text =
dataGridView1.CurrentRow.Cells[2].Value.ToString();
    comboBox1.Text =
dataGridView1.CurrentRow.Cells[3].Value.ToString();
    maskedTextBox2.Text =
dataGridView1.CurrentRow.Cells[4].Value.ToString();
    dateTimePicker2.Text =
dataGridView1.CurrentRow.Cells[5].Value.ToString();
    dateTimePicker1.Text =
dataGridView1.CurrentRow.Cells[6].Value.ToString();
    textBox7.Text =
dataGridView1.CurrentRow.Cells[7].Value.ToString();
    maskedTextBox2.Text = _Sh.Codec(maskedTextBox2.Text, -5/*-
(int)numericUpDown1.Value*/);
}

private void Button1_Click(object sender, EventArgs e)
{
    /*Добавление записи используя метод из класса*/
    try
    {
```

```
        if (comboBox2.Text == null || maskedTextBox1.TextLength ==
0 || comboBox1.Text == null || maskedTextBox2.TextLength == 0 ||
textBox7.TextLength == 0)
        {
            MessageBox.Show("Занесите все данные!!!");
        }
        else
        {
            _P.Zaivka_add(/*Convert.ToInt32(comboBox2.Text)*/comboBox2.SelectedIndex +
1, maskedTextBox1.Text, comboBox1.SelectedIndex + 1, maskedTextBox2.Text =
_Sh.Codec(maskedTextBox2.Text, 5/*Смещение*/), dateTimePicker2.Text,
dateTimePicker1.Text, textBox7.Text);
            MessageBox.Show("Заявка добавлена добавлен");
            //Grid_Load();
            Zaivka_Load();
        }
    }
    catch (Exception ex)
    {
        //MessageBox.Show("Произошла ошибка при добавлении заявки,
\нпожалуйста оповестите об этом администратора!");
        MessageBox.Show(ex.Message);
    }
}

private void Button2_Click(object sender, EventArgs e)
{
    /*изменение записи*/
    try
    {
        Using_Base _UB = new Using_Base();
        _UB.SQLconnect.Open();
        SqlCommand UP = new SqlCommand("UPDATE Zaivka SET " +
"Sotr_ID = '" + (comboBox2.SelectedIndex + 1) + "' ," +
"Data_oform = '" + maskedTextBox1.Text + "' ," +
"Posetitel_ID = '" + (comboBox1.SelectedIndex + 1) + "' ,"
+
"Nomer_and_Seria_Pasport = '" + (maskedTextBox2.Text =
_Sh.Codec(maskedTextBox2.Text, 5/*Смещение*/)) + "' ," +
"Data_prihoda = '" + dateTimePicker2.Text + "' ," +
"Vremi_prihoda = '" + dateTimePicker1.Text + "' ," +
"Mesto_prihoda = '" + textBox7.Text + "' WHERE ID_zaiivka =
" + dataGridView1[0, dataGridView1.CurrentRow.Index].Value,
_UB.SQLconnect);
        UP.ExecuteNonQuery();
        _UB.SQLconnect.Close();
        //Grid_Load();
        Zaivka_Load();
        MessageBox.Show("Обновлено");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

```
}

private void Button3_Click(object sender, EventArgs e)
{
    /*Удаление выбранной записи*/
    Using_Base _UB = new Using_Base();
    _UB.SQLconnect.Open();
    /*создание коанды которая хранит зранимую процедуру*/
    SqlCommand del = new SqlCommand("[DBO].[Zaivka_delete]",
    _UB.SQLconnect);
    del.CommandType = CommandType.StoredProcedure;//тип команды
процедура
    try
    {
        SqlParameter id = new SqlParameter();//создание параметра
        {
            ParameterName = "@ID_zaiivka",//объявление параметра
            Value = dataGridView1.CurrentRow.Cells[0].Value
        };
        del.Parameters.Add(id);
        del.ExecuteNonQuery();
        _UB.SQLconnect.Close();
        Zaivka_Load();
        MessageBox.Show("Заявка удалена");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

public void Zaivka_Load()
{
    try
    {
        Using_Base _UB = new Using_Base();
        _UB.SQLconnect.Open();//подключение к БД
        /*Виртуальная таблица*/
        SqlCommand _Posetitel = new SqlCommand("SELECT
Zaivka.ID_zaiivka, Sotr.FIO_S, Zaivka.Data_oform, Posetitel.FIO_P,
Zaivka.Nomer_and_Seria_Pasport, Zaivka.Data_prihoda, Zaivka.Vremi_prihoda,
Zaivka.Mesto_prihoda FROM [DBO].[Zaivka], [DBO].[Sotr], [DBO].[Posetitel]
WHERE Zaivka.Sotr_ID = Sotr.ID_sotr and Zaivka.Posetitel_ID =
Posetitel.ID_Posetitel", _UB.SQLconnect);
        SqlDataReader Posetitel = _Posetitel.ExecuteReader();

        DataTable _P = new DataTable();
        _P.Load(Posetitel);
        dataGridView1.DataSource = _P;
        dataGridView1.Columns[0].Visible = false;
        dataGridView1.Columns[1].HeaderText = "Сотрудник";
        dataGridView1.Columns[2].HeaderText = "Дата оформления";
        dataGridView1.Columns[3].HeaderText = "Посетитель";
        dataGridView1.Columns[4].HeaderText = "Серия и номер
паспорта";
    }
}
```



```
        dataGridView1.Columns[5].HeaderText = "Дата прихода";
        dataGridView1.Columns[6].HeaderText = "Время прихода";
        dataGridView1.Columns[7].HeaderText = "Место прихода";
        dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill;
        _UB.SQLconnect.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void Button5_Click(object sender, EventArgs e)
{
    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        dataGridView1.Rows[i].Selected = false;
        for (int j = 0; j < dataGridView1.ColumnCount; j++)
            if (dataGridView1.Rows[i].Cells[j].Value != null)//если
таблица не пуста
                if
(dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))//
если данные записанные в поле есть
                {
                    dataGridView1.Rows[i].Selected = true;//Тогда
происходит выделение
                    break;
                }
    }
}

private void TextBox7_KeyPress(object sender, KeyPressEventArgs e)
{
    /*можно водить только те символы котрые указаны в кавычках*/
    if ((e.KeyChar >= 'А' && e.KeyChar <= 'Я') || (e.KeyChar >= 'а'
&& e.KeyChar <= 'я') || /*(e.KeyChar >= '0' && e.KeyChar <= '9') ||*/
        e.KeyChar == (char)Keys.CapsLock || e.KeyChar ==
(char)Keys.Space || e.KeyChar == '_' || e.KeyChar == ',' || e.KeyChar ==
(char)Keys.Back)
    {
    }
    else
    {
        e.Handled = true;
    }
}

private void Timer1_Tick(object sender, EventArgs e)
{
    /*запись текущей даты для того что бы дату не подделали*/
    maskedTextBox1.Text = DateTime.Now.ToString("dd/MM/yyyy");
}
}
```

9. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ

SPIKOK_POSETITELEI.CS

Ниже представлен код программного модуля Spisok_posetitelei.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.Data.SqlClient;
using System.Data.Sql;
using System.Diagnostics;

namespace Prohod
{
    public partial class Spisok_posetitelei : Form
    {
        Procedure _P = new Procedure(); //класс методов
        Shifr _Sh = new Shifr(); //класс шифрование

        public Spisok_posetitelei()
        {
            InitializeComponent();
        }

        private void Grid_Load()
        { /*таблица посетителей*/
            _P.Posetitel_void();
            dataGridView1.DataSource = Program.Posetitel_Select;
            dataGridView1.Columns[0].Visible = false;
            dataGridView1.Columns[1].HeaderText = "Посетитель";
            dataGridView1.Columns[2].HeaderText = "Номер и серия паспорта";
            dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill;
        }

        private void DataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        { /*При выборе поля в datagridview данные дублируются в полях
ввода*/
            textBox1.Text =
dataGridView1.CurrentRow.Cells[1].Value.ToString();
            maskedTextBox1.Text =
dataGridView1.CurrentRow.Cells[2].Value.ToString();
            maskedTextBox1.Text = _Sh.Codec(maskedTextBox1.Text, -5/*-
(int)numericUpDown1.Value*/);
        }
    }
}
```

```
private void Spisok_posetitelei_FormClosed(object sender,
FormClosedEventArgs e)
{
    AddZaivka _AD = new AddZaivka(); //форма создания заявки
    this.Hide();
    _P.Sotr_Load(_AD.comboBox2); //список сотрудников
    _P.Pos_Load(_AD.comboBox1); //список посетителей
    _AD.Zaivka_Load(); //таблица заявок
}

private void Button1_Click(object sender, EventArgs e)
{ /*Добавление нового пользователя*/
    if (textBox1.TextLength == 0 || maskedTextBox1.TextLength == 0)
    {
        MessageBox.Show("Занесите все данные!!!");
    }
    else
    {
        _P.Posetitel_add(textBox1.Text, maskedTextBox1.Text =
        _Sh.Codec(maskedTextBox1.Text, 5/*Смещение*/)/*Шифрование*/);
        MessageBox.Show("Посетитель добавлен");
        Grid_Load();
    }
}

private void Spisok_posetitelei_Load(object sender, EventArgs e)
{
    Grid_Load();
    this.Activate(); //форма активна
}

private void Button2_Click(object sender, EventArgs e)
{ /*обновление записи*/
    try
    {
        Using_Base _UB = new Using_Base();
        _UB.SQLconnect.Open();
        SqlCommand UP = new SqlCommand("UPDATE Posetitel SET " +
        "FIO_P = '" + textBox1.Text + "' , " +
        "Nomer_and_Seria_Pasport = '" + (maskedTextBox1.Text =
        _Sh.Codec(maskedTextBox1.Text, 5/*Смещение*/)) + "' WHERE ID_Posetitel = "
        + dataGridView1[0, dataGridView1.CurrentRow.Index].Value, _UB.SQLconnect);
        UP.ExecuteNonQuery();
        _UB.SQLconnect.Close();
        Grid_Load();
        MessageBox.Show("Обновлено");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void Button3_Click(object sender, EventArgs e)
```

```
        { /*удаление записи*/
            Using_Base _UB = new Using_Base();
            _UB.SQLconnect.Open();
            /*команда содержащая хранимую процедуру на удаление*/
            SqlCommand del = new SqlCommand("[DBO].[Posetitel_delete]",
            _UB.SQLconnect);
            del.CommandType = CommandType.StoredProcedure; //тип команды
процедура
            try
            {
                SqlParameter id = new SqlParameter() //создание параметра
                {
                    ParameterName = "@ID_Posetitel", //имя параметра
                    Value = dataGridView1.CurrentRow.Cells[0].Value
                };
                del.Parameters.Add(id);
                del.ExecuteNonQuery();
                _UB.SQLconnect.Close();
                Grid_Load();
                MessageBox.Show("Посетитель удалён");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

private void Button4_Click(object sender, EventArgs e)
{
    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        dataGridView1.Rows[i].Selected = false; //Фокус убирается
        for (int j = 0; j < dataGridView1.ColumnCount; j++)
            if (dataGridView1.Rows[i].Cells[j].Value != null) //если
таблица не равно нулю
            {
                if
                (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text)) //
ищет поле по значению, которое написано в textBox8
                {
                    dataGridView1.Rows[i].Selected = true; //если
найдено тогда выделяется синим
                    break;
                }
            }
    }
}
}
```

10. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ

SPISOK_ZAIVKA.CS

Ниже представлен код программного модуля Spisok_Zaivka.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.Data.SqlClient;
using System.Data.Sql;
using System.Diagnostics;
using Word = Microsoft.Office.Interop.Word;
using Microsoft.Office;
using System.Drawing.Printing;

namespace Prohod
{
    public partial class Spisok_Zaivka : Form
    {
        Procedure _P = new Procedure(); //класс методов
        Word _WD = new Word(); //класс генерации word
        ToolTip _T = new ToolTip(); //подсказка

        public Spisok_Zaivka()
        {
            InitializeComponent();

            //private void Grid_Load()
            //{
            //    _P.Ozaivka_void();
            //    dataGridView1.DataSource = Program.Ozaivka_Select;
            //    dataGridView1.Columns[0].Visible = false;
            //    dataGridView1.Columns[1].HeaderText = "Место выдачи";
            //    dataGridView1.Columns[2].HeaderText = "Дата оформления";
            //    dataGridView1.Columns[3].HeaderText = "Посетитель";
            //    dataGridView1.Columns[4].HeaderText = "Номер и серия
паспорта";
            //    dataGridView1.Columns[5].HeaderText = "Дата прихода";
            //    dataGridView1.Columns[6].HeaderText = "Время прихода";
            //    dataGridView1.Columns[7].HeaderText = "Место прихода";
            //    dataGridView1.Columns[8].HeaderText = "Сотрудник";
            //    dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill;
            //}

            private void Spisok_Zaivka_Load(object sender, EventArgs e)
```

```
    { /*Таблица оформленных заявок*/  
        //Grid_Load();  
        Spisok_Zaivka_load();  
    }  
  
    private void Spisok_Zaivka_FormClosed(object sender,  
FormClosedEventArgs e)  
    {  
        Form1 _F = new Form1(); //главная форма  
        _F.label1.Text = Properties.Settings.Default.L1;  
        _F.label2.Text = Properties.Settings.Default.L2;  
        this.Hide();  
        _F.Show();  
    }  
  
    private void PictureBox1_Click(object sender, EventArgs e)  
    { /*Печать только выделенных записей*/  
        try  
        {  
            Word _WD = new Word();  
            SaveFileDialog sfd = new SaveFileDialog(); //создание  
компонента SaveFileDialog  
  
            sfd.Filter = "Word Documents (*.docx)|*.docx"; //расширение  
файла  
  
            sfd.FileName = "Список.docx"; //имя файла  
  
            if (sfd.ShowDialog() == DialogResult.OK) //Окно сохранения  
            {  
                _WD.MS_Export_Table(dataGridView1, sfd.FileName);  
            }  
        }  
        catch (Exception ex)  
        {  
            MessageBox.Show(ex.Message);  
        }  
    }  
  
    private void PictureBox1_MouseMove(object sender, MouseEventArgs e)  
    { /*Подсказка*/  
        _T.SetToolTip(pictureBox1, "Распечатать выделенные поля");  
    }  
  
    private void Button1_Click_1(object sender, EventArgs e)  
    {  
        Using_Base _UB = new Using_Base();  
        _UB.SQLconnect.Open();  
        /*Команда хранит хранимую процедуру на удаление*/  
        SqlCommand del = new  
SqlCommand("[DBO].[Oformlennai_zaiivka_delete]", _UB.SQLconnect);  
        del.CommandType = CommandType.StoredProcedure; //тип команды  
процедура  
        try
```

```
{
    SqlParameter id = new SqlParameter()//создание параметра
    {
        ParameterName = "@ID_Ozaivka",//имя параметра
        Value = dataGridView1.CurrentRow.Cells[0].Value
    };
    del.Parameters.Add(id);
    del.ExecuteNonQuery();
    _UB.SQLconnect.Close();
    //Grid_Load();
    Spisok_Zaivka_load();
    MessageBox.Show("Запись удалена");
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void Spisok_Zaivka_load()
{
    try
    {
        Using_Base _UB = new Using_Base();
        _UB.SQLconnect.Open();//открытие подключение
        /*Создание виртуальной таблицы */
        SqlCommand _Spisok_Zaivka_Pos = new SqlCommand("SELECT
Oformlennai_zaiivka.ID_Ozaivka, Oformlennai_zaiivka.Mesto_Vidachi,
Oformlennai_zaiivka.Data_oform, Posetitel.FIO_P, " +
            "Oformlennai_zaiivka.Nomer_and_Seria_Pasport,
Oformlennai_zaiivka.Data_prihoda, Oformlennai_zaiivka.Vremi_prihoda,
Oformlennai_zaiivka.Mesto_prihoda, Sotr.FIO_S " +
            "FROM [DBO].[Oformlennai_zaiivka], [DBO].[Posetitel],
[DBO].[Sotr] WHERE Oformlennai_zaiivka.S_ID = Sotr.ID_Sotr and
Oformlennai_zaiivka.Pos_ID = Posetitel.ID_Posetitel", _UB.SQLconnect);
        SqlDataReader SZP = _Spisok_Zaivka_Pos.ExecuteReader();

        DataTable _SZP = new DataTable();
        _SZP.Load(SZP);
        dataGridView1.DataSource = _SZP;
        /*Наименование столбцов*/
        dataGridView1.Columns[0].Visible = false;
        //dataGridView1.Columns[0].HeaderText = "";
        dataGridView1.Columns[1].HeaderText = "Место выдачи";
        dataGridView1.Columns[2].HeaderText = "Дата оформления";
        dataGridView1.Columns[3].HeaderText = "Посетитель";
        dataGridView1.Columns[4].HeaderText = "Номер и серия
паспорта";

        dataGridView1.Columns[5].HeaderText = "Дата прихода";
        dataGridView1.Columns[6].HeaderText = "Время прихода";
        dataGridView1.Columns[7].HeaderText = "Место прихода";
        dataGridView1.Columns[8].HeaderText = "Сотрудник";
        dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill;
```

```
        _UB.SQLconnect.Close();  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.Message);  
    }  
}  
}
```


11. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ PROCEDURE.CS

Ниже представлен код программного модуля Procedure.cs.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using System.Data.Sql;
using System.Diagnostics;
using System.Data;
using System.Windows.Forms;

namespace Prohod
{
    class Procedure
    {
        //-----СОТРУДНИКИ-----
        -----
        public void Sotr_void()//таблица сотрудников
        {
            Using_Base _UB = new Using_Base();/*класс подключения в БД*/
            _UB.SQLconnect.Open();//открытие
            /*команда хранит таблицу*/
            SqlCommand _Sotr = new SqlCommand("SELECT [ID_Sotr], [FIO_S],
[Log_S], [Pass_S], [Role_ID], [Dop_Inf] FROM [DBO].[Sotr]", _UB.SQLconnect);
            SqlDataReader tableReader = _Sotr.ExecuteReader();//чтение
таблицы
            DataTable Table = new DataTable();//создание таблицы
            Table.Load(tableReader);//загрузка таблицы
            Program.Sotr_Select = Table;
            _UB.SQLconnect.Close();//закрытие
        }

        public void Sotr_add(string A, string B, string C, int D, string
F)//ДОБАВЛЕНИЕ СОТРУДНИКОВ
        {
            Using_Base _UB = new Using_Base();
            _UB.SQLconnect.Open();
            SqlCommand Sotr_ADD = new SqlCommand("sotr_insertinto",
            _UB.SQLconnect);
            Sotr_ADD.CommandType = CommandType.StoredProcedure;
            Sotr_ADD.Parameters.AddWithValue("@FIO_S", A);
            Sotr_ADD.Parameters.AddWithValue("@Log_S", B);
            Sotr_ADD.Parameters.AddWithValue("@Pass_S", C);
            Sotr_ADD.Parameters.AddWithValue("@Role_ID", D);
            Sotr_ADD.Parameters.AddWithValue("@Dop_Inf", F);
            Sotr_ADD.ExecuteNonQuery();
            _UB.SQLconnect.Close();
        }
    }
}
```

```
//-----Роль-----
public void Role_void()//таблица Роль
{
    Using_Base _UB = new Using_Base();
    _UB.SQLconnect.Open();
    SqlCommand _Role = new SqlCommand("SELECT [ID_role], [Rolle] FROM
[DBO].[Rolli]", _UB.SQLconnect);
    SqlDataReader tableReader = _Role.ExecuteReader();
    DataTable Table = new DataTable();
    Table.Load(tableReader);
    Program.Role_Select = Table;
    _UB.SQLconnect.Close();
}

public void role_add(string A)//ДОБАВЛЕНИЕ Роли
{
    Using_Base _UB = new Using_Base();
    _UB.SQLconnect.Open();
    SqlCommand role_ADD = new SqlCommand("role_insertinto",
_UB.SQLconnect);
    role_ADD.CommandType = CommandType.StoredProcedure;
    role_ADD.Parameters.AddWithValue("@Rolle", A);
    role_ADD.ExecuteNonQuery();
    _UB.SQLconnect.Close();
}

//-----Посетитель-----
---
public void Posetitel_void()//таблица сотрудников
{
    Using_Base _UB = new Using_Base();
    _UB.SQLconnect.Open();
    SqlCommand _Posetitel = new SqlCommand("SELECT [ID_Posetitel],
[FIO_P], [Nomer_and_Seria_Pasport] FROM [DBO].[Posetitel]", _UB.SQLconnect);
    SqlDataReader tableReader = _Posetitel.ExecuteReader();
    DataTable Table = new DataTable();
    Table.Load(tableReader);
    Program.Posetitel_Select = Table;
    _UB.SQLconnect.Close();
}

public void Posetitel_add(string A, string B)//ДОБАВЛЕНИЕ Посетителей
{
    Using_Base _UB = new Using_Base();
    _UB.SQLconnect.Open();
    SqlCommand Posetitel_ADD = new
SqlCommand("Posetitel_insertinto", _UB.SQLconnect);
    Posetitel_ADD.CommandType = CommandType.StoredProcedure;
    Posetitel_ADD.Parameters.AddWithValue("@FIO_P", A);
    Posetitel_ADD.Parameters.AddWithValue("@Nomer_and_Seria_Pasport", B);
    Posetitel_ADD.ExecuteNonQuery();
    _UB.SQLconnect.Close();
}
```

```
    }

    //-----Заявка-----
    -----
    public void Zaivka_void()//таблица заявки
    {
        Using_Base _UB = new Using_Base();
        _UB.SQLconnect.Open();
        SqlCommand _Zaivka = new SqlCommand("SELECT [ID_zaiivka],
[Sotr_ID], [Data_oform], [Posetitel_ID], [Nomer_and_Seria_Pasport],
[Data_prihoda], [Vremi_prihoda], [Mesto_prihoda] FROM [DBO].[Zaivka]",
_UB.SQLconnect);
        SqlDataReader tableReader = _Zaivka.ExecuteReader();
        DataTable Table = new DataTable();
        Table.Load(tableReader);
        Program.Zaivka_Select = Table;
        _UB.SQLconnect.Close();
    }

    public void Zaivka_add(int A, string B, int C, string D, string I,
string F, string J)//ДОБАВЛЕНИЕ Заявки
    {
        Using_Base _UB = new Using_Base();
        _UB.SQLconnect.Open();
        SqlCommand Zaivka_ADD = new SqlCommand("Zaivka_insertinto",
_UB.SQLconnect);
        Zaivka_ADD.CommandType = CommandType.StoredProcedure;
        Zaivka_ADD.Parameters.AddWithValue("@Sotr_ID", A);
        Zaivka_ADD.Parameters.AddWithValue("@Data_oform", B);
        Zaivka_ADD.Parameters.AddWithValue("@Posetitel_ID", C);
        Zaivka_ADD.Parameters.AddWithValue("@Nomer_and_Seria_Pasport",
D);
        Zaivka_ADD.Parameters.AddWithValue("@Data_prihoda", I);
        Zaivka_ADD.Parameters.AddWithValue("@Vremi_prihoda", F);
        Zaivka_ADD.Parameters.AddWithValue("@Mesto_prihoda", J);
        Zaivka_ADD.ExecuteNonQuery();
        _UB.SQLconnect.Close();
    }

    //-----Оформленная заявка-----
    -----
    public void Ozaivka_void()//таблица оформленной заявки
    {
        Using_Base _UB = new Using_Base();
        _UB.SQLconnect.Open();
        SqlCommand _Ozaivka = new SqlCommand("SELECT [ID_Ozaivka],
[Mesto_Vidachi], [Data_oform], [Pos_ID], [Nomer_and_Seria_Pasport],
[Data_prihoda], [Vremi_prihoda], [Mesto_prihoda], [S_ID] FROM
[DBO].[Oformlennai_zaiivka]", _UB.SQLconnect);
        SqlDataReader tableReader = _Ozaivka.ExecuteReader();
        DataTable Table = new DataTable();
        Table.Load(tableReader);
        Program.Ozaivka_Select = Table;
    }
}
```

```
public void Ozaivka_add(string A, string B, int C, string D, string
E, string F, string G, int H)//ДОБАВЛЕНИЕ в учёт
{
    Using_Base _UB = new Using_Base();
    _UB.SQLconnect.Open();
    SqlCommand Ozaivka_ADD = new
SqlCommand("Oformlennai_zaiivka_insertinto", _UB.SQLconnect);
    Ozaivka_ADD.CommandType = CommandType.StoredProcedure;
    Ozaivka_ADD.Parameters.AddWithValue("@Mesto_Vidachi", A);
    Ozaivka_ADD.Parameters.AddWithValue("@Data_oform", B);
    Ozaivka_ADD.Parameters.AddWithValue("@Pos_ID", C);
    Ozaivka_ADD.Parameters.AddWithValue("@Nomer_and_Seria_Pasport",
D);

    Ozaivka_ADD.Parameters.AddWithValue("@Data_prihoda", E);
    Ozaivka_ADD.Parameters.AddWithValue("@Vremi_prihoda", F);
    Ozaivka_ADD.Parameters.AddWithValue("@Mesto_prihoda", G);
    Ozaivka_ADD.Parameters.AddWithValue("@S_ID", H);
    Ozaivka_ADD.ExecuteNonQuery();
    _UB.SQLconnect.Close();
}

//-----Учёт входа пользователей-----
-----
public void Time_vhod_void()//таблица учёта входа пользователей
{
    Using_Base _UB = new Using_Base();
    _UB.SQLconnect.Open();
    SqlCommand _Time_vhod = new SqlCommand("SELECT [ID_vhod],
[Sotr_ID], [FIO_S], [Vrema] FROM [DBO].[Time_vhod]", _UB.SQLconnect);
    SqlDataReader tableReader = _Time_vhod.ExecuteReader();
    DataTable Table = new DataTable();
    Table.Load(tableReader);
    Program.Time_vhod_Select = Table;
}

public void Time_vhod_add(int A, string B, string C)//ДОБАВЛЕНИЕ в
учёт
{
    Using_Base _UB = new Using_Base();
    _UB.SQLconnect.Open();
    SqlCommand Time_vhod_ADD = new
SqlCommand("Time_vhod_insertinto", _UB.SQLconnect);
    Time_vhod_ADD.CommandType = CommandType.StoredProcedure;
    Time_vhod_ADD.Parameters.AddWithValue("@Sotr_ID", A);
    Time_vhod_ADD.Parameters.AddWithValue("@FIO_S", B);
    Time_vhod_ADD.Parameters.AddWithValue("@Vrema", C);
    Time_vhod_ADD.ExecuteNonQuery();
    _UB.SQLconnect.Close();
}

//-----combobox роли-----
-----
public class Role
```

```
    /*создание переменных*/
    public int id { get; set; }
    public string ROLLE { get; set; }
    public override string ToString()
    {
        return string.Format("Rolle: {1}, ID_role: {0}", ROLLE, id);
    }
}

public void Rol_Load(ComboBox CB)
{
    try
    {
        Using_Base _UB = new Using_Base();
        _UB.SQLconnect.Open();
        /*команда создания таблицы*/
        SqlCommand _rolli = new SqlCommand("SELECT ID_role, Rolle
FROM [DBO].[Rolli]", _UB.SQLconnect); // вывод таблицы с ролями
        /*создание листа в которой каждая переменная будет хранить
поле из таблицы*/
        List<Role> rol = new List<Role>();
        {
            SqlDataReader ReadRol = _rolli.ExecuteReader();
            while (ReadRol.Read())
            {
                Role coun = new Role
                {
                    id = int.Parse(ReadRol["ID_role"].ToString()),
                    ROLLE = (ReadRol["Rolle"].ToString())
                };
                rol.Add(coun);
            }
        }
        CB.DataSource = rol;
        CB.DisplayMember = "ROLLE"; //показывает это
        CB.ValueMember = Convert.ToString("id"); //записывает это
        _UB.SQLconnect.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

//-----combobox сотрудник-----
-----

public class Sotr
{
    public int ID_sotr { get; set; }
    public string FIO_S { get; set; }
    public override string ToString()
    {
        return string.Format("FIO_S: {1}, ID_sotr: {0}", FIO_S,
ID_sotr);
    }
}
```

```
    }
}

public void Sotr_Load(ComboBox CB)
{
    Using_Base _UB = new Using_Base();
    _UB.SQLconnect.Open();
    SqlCommand _rolli = new SqlCommand("SELECT ID_sotr, FIO_S FROM
[DBO].[Sotr]", _UB.SQLconnect); // вывод таблицы

    List<Sotr> rol = new List<Sotr>(); // создание листа
    {
        SqlDataReader ReadRol = _rolli.ExecuteReader();
        while (ReadRol.Read())
        {
            Sotr coun = new Sotr
            {
                ID_sotr = int.Parse(ReadRol["ID_sotr"].ToString()),
                FIO_S = (ReadRol["FIO_S"].ToString())
            };
            rol.Add(coun);
        }
    }
    CB.DataSource = rol;
    CB.DisplayMember = "FIO_S";
    CB.ValueMember = Convert.ToString("ID_sotr");
    _UB.SQLconnect.Close();
}

//-----combobox посетитель--
-----
public class Pos
{
    public int ID_Posetitel { get; set; }
    public string FIO_P { get; set; }
    public override string ToString()
    {
        return string.Format("FIO_P: {1}, ID_Posetitel: {0}", FIO_P,
ID_Posetitel);
    }
}

public void Pos_Load(ComboBox CB)
{
    Using_Base _UB = new Using_Base();
    _UB.SQLconnect.Open();
    SqlCommand _rolli = new SqlCommand("SELECT ID_Posetitel, FIO_P
FROM [DBO].[Posetitel]", _UB.SQLconnect); // вывод таблицы

    List<Pos> rol = new List<Pos>(); // создание листа
    {
        SqlDataReader ReadRol = _rolli.ExecuteReader();
        while (ReadRol.Read())
        {
```

```
Pos coun = new Pos
{
    ID_Posetitel
int.Parse(ReadRol["ID_Posetitel"].ToString()),
    FIO_P = (ReadRol["FIO_P"].ToString())
};
rol.Add(coun);
}
}
CB.DataSource = rol;
CB.DisplayMember = "FIO_P";
CB.ValueMember = Convert.ToString("ID_Posetitel");
_UB.SQLconnect.Close();
}
}
```

12. ТЕКСТ ПРОГРАММНОГО МОДУЛЯ USING_BASE.CS

Ниже представлен код программного модуля Using_Base.cs.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace Prohod
{
    class Using_Base
    {
        public string bd, bdpass, bdhost, bdlog; //создание переменны
        public static string BD, BDPass, BDHost, BDLog; //создание
статических переменных
        /*подключение к базе данных*/
        public SqlConnection SQLconnect = new SqlConnection("Data Source="
+ BDHost + ";Initial Catalog=" + BD + ";User ID=" + BDLog + ";Password=" +
BDPass + "");

        public void SetConn()
        { /*передача статических данных в динамические переменные*/
            try
            {
                BD = bd;
                BDPass = bdpass;
                BDHost = bdhost;
                BDLog = bdlog;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
    }
}
```