Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Российский экономический университет имени Г.В. Плеханова»

Московский приборостроительный техникум

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДАЮ  Преподаватель |  |  |  | УТВЕРЖДАЮ  Преподаватель |
|  |  |  |  |  |
| \_\_\_\_\_\_\_\_\_\_ А.А. Шимбирёв |  |  |  | \_\_\_\_\_\_\_\_\_\_ А.А. Комаров |
|  |  |  |  |  |
| «\_\_\_» \_\_\_\_\_\_\_\_\_\_20\_\_ г. |  |  |  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_20\_\_ г. |

Информационная система аэропорта

Текст программы

ЛИСТ УТВЕРЖДЕНИЯ

МПТ УП 02.01 П50-2-18 5-ЛУ

Листов 21

|  |  |
| --- | --- |
|  | Менеджер проекта, студент группы П50-2-18 |
|  |  |
|  | \_\_\_\_\_\_\_\_\_\_ Г.Д. Дебольский |
|  |  |
|  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_20\_\_ г. |
|  | Ответственный исполнитель, студент группы П50-2-18 |
|  |  |
|  | \_\_\_\_\_\_\_\_\_\_ А.М. Суслин |
|  |  |
|  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_20\_\_ г. |

20\_\_

ПРИЛОЖЕНИЕ И. ТЕКСТ ПРОГРАММЫ

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Российский экономический университет имени Г.В. Плеханова»

Московский приборостроительный техникум

УТВЕРЖДЕН

МПТ УП 02.01 П50-2-18 5-ЛУ

Информационная система аэропорта

Текст программы

МПТ УП 02.01 П50-2-18 5

Листов 21

2021

## АННОТАЦИЯ

В данном программном документе приведен C# скрипт информационной системы аэропорта «IS Air».

В данном программном документе, в разделе «ТЕКСТ ПРОГРАММЫ» указаны: Наименование программы, область применения программы, модули, код программы.

Содержание

[АННОТАЦИЯ 3](#_Toc65210156)

[1. ТЕКСТ ПРОГРАММЫ 5](#_Toc65210157)

[1.1. Наименование программы 5](#_Toc65210158)

[1.2. Область применения программы 5](#_Toc65210159)

[1.3. Модули 5](#_Toc65210160)

[1.4. Код программы 5](#_Toc65210161)

## 1. ТЕКСТ ПРОГРАММЫ

### 1.1. Наименование программы

Наименование – «Информационная система аэропорта «IS Air».

### 1.2. Область применения программы

Программа предназначена для организации работы в аэропорту.

### 1.3. Модули

Таблица 1- Модули

| Модуль | Описание | Количество строк кода | Размер в (Кбайтах) |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| AddEmployee.cs | Данный модуль отвечает за добавление новых сотрудников | 89 | 4 |
| AddFlight.cs | Данный модуль отвечает за добавление новых рейсов | 86 | 4 |
| BuyTiket.cs | Данный модуль отвечает за просмотр рейсов и покупку билетов | 116 | 5 |
| Employee.cs | Данный модуль отвечает за просмотр информации о сотрудниках | 87 | 4 |
| Enter.cs | Данный модуль отвечает за  авторизацию для клиентов | 55 | 3 |
| EnterForSotrudnik.cs | Данный модуль отвечает за авторизацию для сотрудников | 52 | 3 |
| Flights.cs | Данный модуль отвечает за просмотр информации о рейсах | 102 | 4 |
| Main.cs | Данный модуль отвечает за главную навигацию между формами авторизации и регистрации | 54 | 3 |
| Registraciy.cs | Данный модуль отвечает за регистрацию новых клиентов | 83 | 4 |
| Sotrudnik.cs | Данный модуль отвечает за главную форму для сотрудников | 64 | 4 |
| TiketConfirm.cs | Данный модуль отвечает за подтверждение оформления билета | 82 | 4 |
| Tikets.cs | Данный модуль отвечает за просмотр информации о всех билетах | 54 | 3 |

### 1.4. Код программы

1. AddEmployee.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 LP02\_21\_Release

{

public partial class AddEmployee : Form

{

DataClass myClass = new DataClass();

public AddEmployee()

{

InitializeComponent();

}

/// <summary>

/// Метод валидации формы (обязательные поля должны быть заполнены)

/// </summary>

/// <returns></returns>

bool validForm()

{

if((tbLastName.Text != "") && (tbFirstName.Text != "") && (tbEmail.Text != "") && (tbPassword.Text != "") && (cbPost.SelectedItem != null))

{

return true;

}

else

{

MessageBox.Show("Некорректный ввод", "Ошибка");

return false;

}

}

/// <summary>

/// Подтверждение добавления сотрудника

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

if (!validForm()) { return; }

try

{

myClass.executeQuery($"insert into LichnieDannie(Family, Imy, Otchestvo) values " +

$"('{tbLastName.Text}', '{tbFirstName.Text}', '{tbPatronymic.Text}'); " +

$"insert into Sotrudnik(Email, Password, IDLichnieDannie, IDDoljnost) values " +

$"('{tbEmail.Text}', '{tbPassword.Text}', " +

$"(select top (1) ID from LichnieDannie order by ID desc), " +

$"(select ID from Doljnost where Naimenovanie='{cbPost.SelectedItem.ToString()}'))");

MessageBox.Show("Сотрудник успешно добавлен", "Успех");

this.Close();

}

catch(Exception excep)

{

MessageBox.Show(excep.Message, "Ошибка");

}

}

/// <summary>

/// При закрытии окна - открывается предыдущее

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void AddEmployee\_FormClosed(object sender, FormClosedEventArgs e)

{

Employee form = (Employee)Application.OpenForms["Employee"];

form.showDataEmployee();

form.Show();

}

/// <summary>

/// При загрузке формы - заполняется комбо-бокс с должностями

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

cbPost.DataSource = myClass.GetListItems("select Naimenovanie from Doljnost");

}

}

}

2. AddFlight.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 LP02\_21\_Release

{

public partial class AddFlight : Form

{

DataClass myClass = new DataClass();

public AddFlight()

{

InitializeComponent();

}

/// <summary>

/// При закрытии окна - открывается предыдущее

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void AddFlight\_FormClosed(object sender, FormClosedEventArgs e)

{

Flights form = (Flights)Application.OpenForms["Flights"];

form.showDataFlights();

form.Show();

}

/// <summary>

/// Метод валидации формы (обязательные поля должны быть заполнены)

/// </summary>

/// <returns></returns>

bool validForm()

{

if((tbFrom.Text != "") && (tbTo.Text != "") && (tbCost.Text != "") && (cbAircrafts.SelectedItem != null))

{

return true;

}

else

{

MessageBox.Show("Некорректный ввод", "Ошибка");

return false;

}

}

/// <summary>

/// Подтверждение добавления нового рейса

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

if (!validForm()) { return; }

try

{

myClass.executeQuery($"insert into Reis(MestoVilita, MestoPribitya, DateVileta, Czena, IDSamolet) values " +

$"('{tbFrom.Text}', '{tbTo.Text}', '{dateTimePicker.Value}', {tbCost.Text}, " +

$"(select ID from Samolet where Naimenovanie='{cbAircrafts.SelectedItem}'))");

MessageBox.Show("Рейс успешно добавлен", "Успех");

this.Close();

}

catch(Exception excep)

{

MessageBox.Show(excep.Message, "Ошибка");

}

}

/// <summary>

/// При открытии формы - загружается список самолётов в комбо-бокс

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

cbAircrafts.DataSource = myClass.GetListItems("select Naimenovanie from Samolet");

}

}

}

3. Tikets.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 LP02\_21\_Release

{

public partial class Tikets : Form

{

DataClass myClass = new DataClass();

public Tikets()

{

InitializeComponent();

}

/// <summary>

/// При открытии окна - загружается информация о всех билетах

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

dataGridTikets.DataSource = myClass.getData($"select " +

$"Reis.MestoVilita as [Откуда], " +

$"Reis.MestoPribitya as [Куда], " +

$"Reis.DateVileta as [Дата вылета], " +

$"LichnieDannie.Family as [Фамилия], " +

$"LichnieDannie.Imy as [Имя], " +

$"Bilet.Summa as [Стоимость], " +

$"Bilet.DateVormirovaniya as [Дата оформления] " +

$"from Bilet " +

$"left join Klient on Bilet.IDKlient = Klient.ID " +

$"left join LichnieDannie on Klient.IDLichnieDannie = LichnieDannie.ID " +

$"left join Reis on Bilet.IDReis = Reis.ID ").Tables[0];

}

/// <summary>

/// При закрытии окна - открывается предыдущее

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void Tikets\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.OpenForms["Sotrudnik"].Show();

}

}

}

4. TiketConfirm.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 LP02\_21\_Release

{

public partial class TiketConfirm : Form

{

public int reisID = 0;

public int klientID = 0;

DataClass myClass = new DataClass();

public TiketConfirm()

{

InitializeComponent();

}

/// <summary>

/// При загрузке окна - заполняем форму информацией о клиенте и рейсе

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

tbLastName.Text = dataGridFIO.Rows[0].Cells["Family"].Value.ToString();

tbFirstName.Text = dataGridFIO.Rows[0].Cells["Imy"].Value.ToString();

tbPatronymic.Text = dataGridFIO.Rows[0].Cells["Otchestvo"].Value.ToString();

tbFrom.Text = dataGridReis.Rows[0].Cells["MestoVilita"].Value.ToString();

tbTo.Text = dataGridReis.Rows[0].Cells["MestoPribitya"].Value.ToString();

tbDate.Text = dataGridReis.Rows[0].Cells["DateVileta"].Value.ToString();

tbSumma.Text = dataGridReis.Rows[0].Cells["Czena"].Value.ToString();

int quantityPlace = (int)myClass.getData($"select KolVoMest from Samolet where ID=(select IDSamolet from Reis where ID={reisID})").Tables[0].Rows[0].ItemArray[0];

int quantityTickets = myClass.getData($"select \* from Bilet where IDReis={reisID}").Tables[0].Rows.Count;

tbQuantityFree.Text = (quantityPlace - quantityTickets).ToString();

}

/// <summary>

/// При закрытии окно - открываем предыдущее

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void TiketConfirm\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.OpenForms["BuyTiket"].Show();

}

/// <summary>

/// Если есть свободные места на самолёте - оформляет билет и заносит его в БД

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

if(tbQuantityFree.Text != "0")

{

try

{

myClass.executeQuery($"insert into Bilet(DateVormirovaniya, Summa, IDKlient, IDReis) values " +

$"('{DateTime.Now}', {tbSumma.Text}, {klientID}, {reisID})");

MessageBox.Show("Билет куплен", "Успех");

this.Close();

}

catch(Exception excep)

{

MessageBox.Show(excep.Message, "Ошибка");

}

}

else

{

MessageBox.Show("Свободных мест нету.", "Ошибка");

}

}

}

}

5. BuyTiket.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 LP02\_21\_Release

{

public partial class BuyTiket : Form

{

DataClass myClass = new DataClass();

string where = "";

public int klientID = 0;

public BuyTiket()

{

InitializeComponent();

}

/// <summary>

/// При закрытии формы - переходим на предыдущее окно

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void BuyTiket\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.OpenForms["Enter"].Show();

}

/// <summary>

/// Метод отображения информации о рейсах в таблицу

/// </summary>

void showDataFlights()

{

dataGridFlights.DataSource = myClass.getData($"select " +

$"Reis.MestoVilita as [Откуда], " +

$"Reis.MestoPribitya as [Куда], " +

$"Reis.DateVileta as [Когда], " +

$"Reis.Czena as [Стоимость], " +

$"Samolet.Naimenovanie as [Самолёт], " +

$"Reis.ID as [ID] " +

$"from Reis " +

$"left join Samolet on Reis.IDSamolet = Samolet.ID " +

$"where 666=666 {where}").Tables[0];

dataGridFlights.Columns[5].Visible = false;

}

/// <summary>

/// При загрузке окна - заполняем таблицу

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

showDataFlights();

}

/// <summary>

/// Метод фильтрации рейсов по городу отправки и городу прибытия

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

if(tbFrom.Text != "")

{

where += $"and MestoVilita='{tbFrom.Text} '";

}

if (tbTo.Text != "")

{

where += $"and MestoPribitya='{tbTo.Text} '";

}

showDataFlights();

where = "";

}

/// <summary>

/// Вызывает форму подтверждения оформления билета

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

if (dataGridFlights.SelectedRows.Count == 0) { MessageBox.Show("Ничего не выбрано", "Ошибка"); return; }

this.Hide();

TiketConfirm form = new TiketConfirm();

form.dataGridFIO.DataSource = myClass.getData($"select " +

$"LichnieDannie.Family as [Family], " +

$"LichnieDannie.Imy as [Imy], " +

$"LichnieDannie.Otchestvo as [Otchestvo] " +

$"from Klient " +

$"left join LichnieDannie on Klient.IDLichnieDannie = LichnieDannie.ID " +

$"where Klient.ID={klientID}").Tables[0];

form.dataGridReis.DataSource = myClass.getData($"select " +

$"MestoVilita as [MestoVilita], " +

$"MestoPribitya as [MestoPribitya], " +

$"DateVileta as [DateVileta], " +

$"Czena as [Czena] " +

$"from Reis " +

$"where ID={dataGridFlights.SelectedRows[0].Cells["ID"].Value}").Tables[0];

form.reisID = (int)dataGridFlights.SelectedRows[0].Cells["ID"].Value;

form.klientID = klientID;

form.Show();

}

}

}

6. Sotrudnik.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 LP02\_21\_Release

{

public partial class Sotrudnik : Form

{

public Sotrudnik()

{

InitializeComponent();

}

/// <summary>

/// При закрытии окна - открывается предыдущее

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void Sotrudnik\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.OpenForms["EnterForSotrudnik"].Show();

}

/// <summary>

/// Вызов формы просмотра списка рейсов

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

this.Hide();

new Flights().Show();

}

/// <summary>

/// Вызов формы просмотра списка сотрудников

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

this.Hide();

new Employee().Show();

}

/// <summary>

/// Вызов формы просмотра списка билетов

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

this.Hide();

new Tikets().Show();

}

}

}

7. Registraciy.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 LP02\_21\_Release

{

public partial class Registraciy : Form

{

DataClass myClass = new DataClass();

public Registraciy()

{

InitializeComponent();

}

/// <summary>

/// Метод регистрации нового клиента

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

if(!validForm()) { return; }

try

{

myClass.executeQuery($"insert into LichnieDannie(Family, Imy, Otchestvo) values " +

$"('{tbLastName.Text}', '{tbFirstName.Text}', '{tbPatronymic.Text}'); " +

$"insert into Klient(Email, Password, Telefon, NomerCarti, IDLichnieDannie) values " +

$"('{tbEmail.Text}', '{tbPassword.Text}', '{tbPhone.Text}', '{tbCardNumber.Text}', " +

$"(select top (1) ID from LichnieDannie order by ID desc))");

MessageBox.Show("Вы успешно зарегестрировались", "Успех");

this.Close();

}

catch(Exception excep)

{

MessageBox.Show(excep.Message, "Ошибка");

}

}

/// <summary>

/// Метод валидации формы (обязательные поля должны быть заполнены)

/// </summary>

/// <returns></returns>

bool validForm()

{

if ((tbLastName.Text != "") && (tbFirstName.Text != "") && (tbEmail.Text != "") && (tbPassword.Text != "") && (tbPasswordConfirm.Text != ""))

{

if (tbPassword.Text == tbPasswordConfirm.Text)

{

return true;

}

else

{

MessageBox.Show("Пароли не совпадают", "Ошибка");

return false;

}

}

else

{

MessageBox.Show("Некорректный ввод", "Ошибка");

return false;

}

}

/// <summary>

/// При закрытии окна - открывается предыдущее

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void Registraciy\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.OpenForms["Main"].Show();

}

}

}

8. Main.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 LP02\_21\_Release

{

public partial class Main : Form

{

public Main()

{

InitializeComponent();

}

/// <summary>

/// Вызов формы авторизации для клиентов

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

this.Hide();

new Enter().Show();

}

/// <summary>

/// Вызов формы регистрации для клиентов

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

this.Hide();

new Registraciy().Show();

}

/// <summary>

/// Вызов формы авторизации для сотрудников

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

this.Hide();

new EnterForSotrudnik().Show();

}

}

}

9. Flights.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 LP02\_21\_Release

{

public partial class Flights : Form

{

DataClass myClass = new DataClass();

string where = "";

public Flights()

{

InitializeComponent();

}

/// <summary>

/// Метод отображения информации о рейсах в таблице

/// </summary>

public void showDataFlights()

{

dataGridFlights.DataSource = myClass.getData($"select " +

$"Reis.MestoVilita as [Откуда], " +

$"Reis.MestoPribitya as [Куда], " +

$"Reis.DateVileta as [Когда], " +

$"Reis.Czena as [Стоимость], " +

$"Samolet.Naimenovanie as [Самолёт], " +

$"Reis.ID " +

$"from Reis " +

$"left join Samolet on Reis.IDSamolet = Samolet.ID " +

$"where 666=666 {where}").Tables[0];

dataGridFlights.Columns[5].Visible = false;

}

/// <summary>

/// При открытии окна - загружается информация о рейсах в таблицу

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

showDataFlights();

}

/// <summary>

/// При закрытии формы - открывается предыдущая

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void Flights\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.OpenForms["Sotrudnik"].Show();

}

/// <summary>

/// Вызывает форму добавления нового рейса

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

this.Hide();

new AddFlight().Show();

}

/// <summary>

/// Метод фильтрации рейсов по городу отправления и городу прибытия

/// </summary>

void filtredData()

{

if (tbFrom.Text != "")

{

where += $"and MestoVilita='{tbFrom.Text} '";

}

if (tbTo.Text != "")

{

where += $"and MestoPribitya='{tbTo.Text} '";

}

showDataFlights();

where = "";

}

/// <summary>

/// Применения фильтрации таблицы

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

filtredData();

}

}

}

10. EnterForSotrudnik.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 LP02\_21\_Release

{

public partial class EnterForSotrudnik : Form

{

DataClass myClass = new DataClass();

public EnterForSotrudnik()

{

InitializeComponent();

}

/// <summary>

/// Авторизации сотрудников

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

DataSet ds = myClass.getData($"select \* from Sotrudnik where Email='{tbEmail.Text}' and Password='{tbPassword.Text}'");

if (ds.Tables[0].Rows.Count == 0)

{

MessageBox.Show("Неправильный логин и/или пароль", "Error");

}

else

{

this.Hide();

new Sotrudnik().Show();

}

}

/// <summary>

/// При закрытии формы - открывается предыдущая

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void EnterForSotrudnik\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.OpenForms["Main"].Show();

}

}

}

11. Enter.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;

namespace LP02\_21\_Release

{

public partial class Enter : Form

{

DataClass myClass = new DataClass();

public Enter()

{

InitializeComponent();

}

/// <summary>

/// Авторизация клиента

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

DataSet ds = myClass.getData($"select \* from Klient where Email='{tbEmail.Text}' and Password='{tbPassword.Text}'");

if(ds.Tables[0].Rows.Count == 0)

{

MessageBox.Show("Неправильный логин и/или пароль", "Error");

}

else

{

this.Hide();

BuyTiket form = new BuyTiket();

form.klientID = (int)ds.Tables[0].Rows[0].ItemArray[0];

form.Show();

}

}

/// <summary>

/// При закрытии окна - открывается предыдущее

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void Enter\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.OpenForms["Main"].Show();

}

}

}

12. Employee.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 LP02\_21\_Release

{

public partial class Employee : Form

{

DataClass myClass = new DataClass();

public Employee()

{

InitializeComponent();

}

/// <summary>

/// При закрытии формы - открывается предыдущая

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void Employee\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.OpenForms["Sotrudnik"].Show();

}

/// <summary>

/// Вызов формы добавления нового сотрудника

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

this.Hide();

new AddEmployee().Show();

}

/// <summary>

/// Метод отображения информации о сотрудниках в таблицу

/// </summary>

public void showDataEmployee()

{

dataGridEmployees.DataSource = myClass.getData($"select " +

$"LichnieDannie.Family as [Фамилия], " +

$"LichnieDannie.Imy as [Имя], " +

$"LichnieDannie.Otchestvo as [Отчество], " +

$"Sotrudnik.Email as [Почта], " +

$"Sotrudnik.Password as [Пароль], " +

$"Doljnost.Naimenovanie as [Должность], " +

$"Sotrudnik.ID as [id] " +

$"from Sotrudnik " +

$"left join LichnieDannie on Sotrudnik.IDLichnieDannie = LichnieDannie.ID " +

$"left join Doljnost on Sotrudnik.IDDoljnost = Doljnost.ID ").Tables[0];

dataGridEmployees.Columns[6].Visible = false;

}

/// <summary>

/// При открытии окна - загружается таблица

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

showDataEmployee();

}

/// <summary>

/// Метод удаления выбранного сотрудника

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

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

{

if(dataGridEmployees.SelectedRows.Count == 0) { MessageBox.Show("Ничего не выбрано", "Ошибка"); return; }

myClass.executeQuery($"delete from Sotrudnik where ID={dataGridEmployees.SelectedRows[0].Cells["ID"].Value} ");

showDataEmployee();

MessageBox.Show("Сотрудник успешно удалён", "Успех");

}

}

}