**Форма «Администратор»**

namespace производПрактика

{

public partial class Admin : Form

{

DataBase dataBase = new DataBase();

public Admin()

{

InitializeComponent();

}

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

{

string query = $"UPDATE zakaz SET Sotrudnik ='{textBox3.Text}', IDZAKAZ = '{textBox17.Text}'";

SqlCommand command = new SqlCommand(query, dataBase.getSqlConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)//проверка на заполненность

{

MessageBox.Show("Выполнено!");

}

else

{

MessageBox.Show("Не выполнено!");

}

}

private void button9\_Click(object sender, EventArgs e)

{

//запрос на изменение данных

string query = $"UPDATE inventar SET Количество = Количество + {textBox8.Text} Where id = '{textBox7.Text}'";

SqlCommand command = new SqlCommand(query, dataBase.getSqlConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)//проверка на заполненность

{

MessageBox.Show("Выполнено!");

}

else

{

MessageBox.Show("Не выполнено!");

}

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Select \* from inventar ORDER BY id";//сортировка по id по возрастанию

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView1.DataSource = table;//вывод данных

}

private void button8\_Click(object sender, EventArgs e)

{

string query = $"INSERT into vidani (id,количество, Заказ) values ('{textBox6.Text}', '{textBox4.Text}', '{textBox16.Text}')";

SqlCommand command = new SqlCommand(query, dataBase.getSqlConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)//проверка на заполненность

{

MessageBox.Show("(\"выдано\");");

}

else

{

MessageBox.Show("Не выполнено!");

}

}

private void button11\_Click(object sender, EventArgs e)

{

string querry = $"insert into inventar(Название,Количество) values ('{textBox9.Text}','{textBox10.Text}')";// запрос на внесение новых данных

SqlCommand command = new SqlCommand(querry, dataBase.getSqlConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)//проверка на заполненность

{

MessageBox.Show("(\"Выполнено\");");

}

else

{

MessageBox.Show("Не выполнено!");

}

}

private void button10\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Select \* from inventar ORDER BY id";//сортировка по id по возрастанию

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView4.DataSource = table;//вывод данных

}

private void button7\_Click(object sender, EventArgs e)

{

//запрос на изменение данных

var count = textBox4.Text;

string query = $"UPDATE inventar SET Количество = Количество - {count} Where id = '{textBox6.Text}'";

SqlCommand command = new SqlCommand(query, dataBase.getSqlConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)//проверка на заполненность

{

MessageBox.Show("Выполнено!");

}

else

{

MessageBox.Show("Не выполнено!");

}

}

private void button6\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Select \* from vidani ORDER BY id";//сортировка по id по возрастанию

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView5.DataSource = table;//вывод данных

}

private void dataGridView5\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

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

{

Application.Exit();

}

private void button12\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Select \* from zdano ORDER BY id";//сортировка по id по возрастанию

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView6.DataSource = table;//вывод данных

}

private void button13\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Delete from vidani WHERE id = '{textBox5.Text}'";//запрос

SqlCommand command = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

dataGridView5.DataSource = table;

}

private void button14\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Select \* from Money ORDER BY id";//сортировка по id по возрастанию

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView2.DataSource = table;//вывод данных

}

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

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"SELECT SUM(Прибыль) from Money";//сортировка по id по возрастанию

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView7.DataSource = table;//вывод данных

}

private void button15\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Select \* from Trati ORDER BY id";//сортировка по id по возрастанию

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView8.DataSource = table;//вывод данных

}

private void button16\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"SELECT SUM(Зарплата)+Sum(Расходы) from Trati";

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView9.DataSource = table;//вывод данных

}

private void button17\_Click(object sender, EventArgs e)

{

double a = Convert.ToDouble(textBox11.Text);

double b = Convert.ToDouble(textBox12.Text);

double c = a - b;

textBox13.Text = c.ToString();

}

private void button18\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Select \* from zakaz ";//сортировка по id по возрастанию

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView1.DataSource = table;//вывод данных

}

private void button1\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Delete from zakaz WHERE IDZAKAZ = '{textBox18.Text}'";//запрос

SqlCommand command = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

dataGridView5.DataSource = table;

}

private void button19\_Click(object sender, EventArgs e)

{

AdminForm adminForm = new AdminForm();

adminForm.ShowDialog();

}

private void button20\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Select \* from Sotrudniki ";//сортировка по id по возрастанию

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView3.DataSource = table;//вывод данных

}

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

{

string query = $"INSERT into Sotrudniki (Name,Familia,Otchestvo) values ('{textBox2.Text}', '{textBox1.Text}', '{textBox14.Text}' , '{textBox20.Text}')";

SqlCommand command = new SqlCommand(query, dataBase.getSqlConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)//проверка на заполненность

{

MessageBox.Show("(\"Выполнено\");");

}

else

{

MessageBox.Show("Не выполнено!");

}

}

private void button21\_Click(object sender, EventArgs e)

{

var count = textBox19.Text;

string query = $"UPDATE Sotrudniki SET Zakazov = Zakazov + {count} Where id = '{textBox15.Text}'";

SqlCommand command = new SqlCommand(query, dataBase.getSqlConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)//проверка на заполненность

{

MessageBox.Show("Выполнено!");

}

else

{

MessageBox.Show("Не выполнено!");

}

}

private void button22\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Delete from Sotrudniki WHERE id = '{textBox22.Text}'";//запрос

SqlCommand command = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

dataGridView3.DataSource = table;

}

private void button23\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Delete from zdano";//запрос

SqlCommand command = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

dataGridView6.DataSource = table;

}

private void button24\_Click(object sender, EventArgs e)

{

var count = textBox23.Text;

string query = $"UPDATE Trati SET Зарплата = {count} Where id = '{textBox26.Text}'";

SqlCommand command = new SqlCommand(query, dataBase.getSqlConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)//проверка на заполненность

{

MessageBox.Show("Выполнено!");

}

else

{

MessageBox.Show("Не выполнено!");

}

}

private void button26\_Click(object sender, EventArgs e)

{

var count = textBox28.Text;

string query = $"UPDATE Trati SET Расходы = Расходы + {count} Where id = '{textBox21.Text}'";

SqlCommand command = new SqlCommand(query, dataBase.getSqlConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)//проверка на заполненность

{

MessageBox.Show("(\"Выполнено\");");

}

else

{

MessageBox.Show("Не выполнено!");

}

}

private void button27\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Select \* from Trati";//запрос

SqlCommand command = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

dataGridView10.DataSource = table;

}

private void button25\_Click(object sender, EventArgs e)

{

var count1 = textBox25.Text;

var count2 = textBox24.Text;

string query = $"UPDATE Money SET Количество = Количество + {count1}, Прибыль = Прибыль + {count2} Where id = '{textBox27.Text}'";

SqlCommand command = new SqlCommand(query, dataBase.getSqlConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)//проверка на заполненность

{

MessageBox.Show("(\"Выполнено\");");

}

else

{

MessageBox.Show("Не выполнено!");

}

}

private void button28\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Select \* from Money";//запрос

SqlCommand command = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

dataGridView11.DataSource = table;

}

}

}

**Мобильное приложение**

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

<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="practica.MainPage">

<ContentPage Title="Главная">

<StackLayout>

<Label Text="Услуги"

HorizontalOptions="Center"

VerticalOptions="Center"

FontSize="Large" />

<Image Source="https://cdn-icons-png.flaticon.com/128/4221/4221771.png"/>

<Button Text="Генеральная уборка"

Clicked="Button1\_Clicked"

HorizontalOptions="Center"

VerticalOptions="Center" />

<Image Source="https://cdn-icons-png.flaticon.com/128/2790/2790507.png"/>

<Button Text="Дезинфекция квартир"

Clicked="Button2\_Clicked"

HorizontalOptions="Center"

VerticalOptions="Center" />

<Image Source="https://cdn-icons-png.flaticon.com/128/1617/1617302.png"/>

<Button Text="Уборка после ремонта"

Clicked="Button3\_Clicked"

HorizontalOptions="Center"

VerticalOptions="Center" />

<Image Source="https://cdn-icons-png.flaticon.com/128/994/994928.png"/>

<Button Text="Комплексная уборка"

Clicked="Button4\_Clicked"

HorizontalOptions="Center"

VerticalOptions="Center" />

<Image Source="https://zhazhda.biz/wp-content/uploads/2016/08/Biznes-ideya-kliningovaya-kompaniya-960x960.png"/>

</StackLayout>

</ContentPage>

<ContentPage Title="Заказ">

<StackLayout>

<Label Text="Заказать"

HorizontalOptions="Center"

VerticalOptions="Center"

FontSize="Large" />

<Entry x:Name="NameEntry"

Placeholder="Имя" />

<Entry x:Name="PhoneEntry"

Placeholder="Номер телефона"

Keyboard="Telephone" />

<Entry x:Name="EmailEntry"

Placeholder="Email"

Keyboard="Email" />

<Entry x:Name="ServicePicker"

Placeholder="Услуга" />

<Editor x:Name="MessageEditor"

Placeholder="Сообщение" />

<DatePicker x:Name="DateDatePicker"

Format="d MMM yyyy" />

<Button Text="Заказать" Clicked="OnButtonClicked"/>

<Image Source="https://cdn-icons-png.flaticon.com/128/5177/5177789.png"/>

</StackLayout>

</ContentPage>

<ContentPage Title="Корзина">

<StackLayout>

<Label Text="Корзина"

HorizontalOptions="Center"

VerticalOptions="Center"

FontSize="Large" />

<Label x:Name="infolabel" />

</StackLayout>

</ContentPage>

</TabbedPage>

namespace practica

{

public partial class MainPage : TabbedPage

{

public MainPage()

{

InitializeComponent();

}

private async void Button1\_Clicked(object sender, EventArgs e)

{

await DisplayAlert("Описание", "Перечень работ оказываемых в рамках услуги \"Генеральная уборка\":\r\n-отмывание любых видов полов и плинтуса;\r\n-удаление трудновыводимых пятен;\r\n-удаление пыли с осветительных приборов и кондиционеров;\r\n-комплексная уборка санузлов (пол, кафельные стены, раковины, \r\nписсуары и унитазы, перегородки, зеркала, двери, мусорные корзины, \r\nдиспенсеры для гигиенической продукции и т.д.);\r\n-чистка и мытье стен, удаление локальных пятен \r\n(для влагостойких поверхностей);\r\n-чистка откосов, подоконников и лоджий;\r\n-протирка радиаторов отопления;\r\n-обеспыливание мебели, оргтехники, предметов интерьера;\r\n-удаление пыли и пятен с розеток, удлинителей и т.д.;\r\n-полировка мебели из натурального дерева;\r\n-уход за предметами интерьера (картины, вазоны, панно и т.д.);\r\n-чистка и полировка предметов из нержавеющей стали, латуни, бронзы;\r\n-вынос мусора, замена пакетов, чистка пепельниц;\r\n-уборка дверных блоков (протир панелей дверей, наличников, полировка \r\nстеклянных элементов, полировка металлических элементов — ручки, \r\n-фурнитура, латунные таблички;\r\n-комплексная уборка входных групп и грязезащитных \r\nзон при входе в здание и т.д. \nСтоимость = 70р за м2.", "OK");

}

private async void Button2\_Clicked(object sender, EventArgs e)

{

await DisplayAlert("Описание", "Дезинфекция квартир\r\nДезинфекция – набор мероприятий механического, \r\nфизического и химического свойства, \r\nкоторые направлены на уничтожение болезнетворных микроорганизмов \r\nдо того, как вирус, \r\nгрибок или бактерия вступят в контакт с человеком \r\n(попадут на кожу, проникнут внутрь организма через дыхательные пути).\nСтоимость = 5000р.", "OK");

}

private async void Button3\_Clicked(object sender, EventArgs e)

{

await DisplayAlert("Описание", "Перечень работ оказываемых в рамках услуги:\r\nудаление пыли и грязи с полов, стен, потолка;\r\nмытье отопительных труб, батарей;\r\nудаление пыли с дверных блоков, внешних поверхностей мебели и \r\nпредметов интерьера;\r\nвлажная и сухая протирка осветительных приборов \r\nмалых размеров, оргтехники;\r\nсухая чистка ковров и мягкой мебели пылесосом;\r\nмытье ванных комнат, душевых кабин, туалетов;\r\nвлажная протирка внешних поверхностей холодильника, \r\nкухонной плиты, и другого кухонного оборудования;\r\nвлажная уборка полов с твердым покрытием \r\n(паркет, ламинат, линолеум, плитка);\r\nвынос мусора (до 10 кг).\nСтоимость = 75р за м2.", "OK");

}

private async void Button4\_Clicked(object sender, EventArgs e)

{

await DisplayAlert("Описание", "Перечень работ оказываемых в рамках услуги:\r\nмытье полов\r\nмебель и подоконники\r\nкухня: влажная протирка мебели (доступные поверхности)\r\nкухня: Внешние поверхности холодильника, плиты и микроволновки\r\nванная, туалет: мытье раковины (металл, искусственный камень)\r\nванная, туалет: раковины, унитазы, биде, поддон и все смесители\r\nочистка кранов и смесителей от налета и ржавчины\r\nосветительные приборы\r\nплинтуса\r\nбатареи и трубы (доступные)\r\nсухая чистка мебели\r\nвлажная уборка всех стен\r\nмытье дверей\r\nвлажная чистка мебели\r\nвлажная и сухая протирка орг техники \nСтоимость = 30-40м2 = 5000р;\n40-60м2 = 6000р;\n60-80м2 = 8000р;\n80-100м2 = 10000р.", "OK");

}

private void OnButtonClicked(object sender, System.EventArgs e)

{

string name = NameEntry.Text;

string phone = PhoneEntry.Text;

string email = EmailEntry.Text;

string service = ServicePicker.Text;

string message = MessageEditor.Text;

DateTime date = DateDatePicker.Date;

infolabel.Text = $"Имя:{name} \nНомер телефона:{phone} \nEmail: {email} \nУслуга: {service} \nСообщение: {message} \nЖелаемая дата: {date}";

}

}

}

**Форма «Сотрудники»**

namespace производПрактика

{

public partial class AdminForm : Form

{

DataBase dataBase = new DataBase();

public AdminForm()

{

InitializeComponent();

}

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

{

}

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

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Select \* from vidani WHERE Заказ = {textBox4.Text}";//сортировка по id по возрастанию

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView2.DataSource = table;//вывод данных

}

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

{

Application.Exit();

}

private void button1\_Click(object sender, EventArgs e)

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Select \* from zakaz ";//сортировка по id по возрастанию

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView1.DataSource = table;//вывод данных

}

private void button6\_Click(object sender, EventArgs e)

{

string querry = $"insert into zdano(id,Количество) values ('{textBox5.Text}','{textBox6.Text}')";// запрос на внесение новых данных

SqlCommand command = new SqlCommand(querry, dataBase.getSqlConnection());

dataBase.openConnection();

if (command.ExecuteNonQuery() == 1)//проверка на заполненность

{

MessageBox.Show("(\"Выполнено\");");

}

else

{

MessageBox.Show("Не выполнено!");

}

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView4.DataSource = table;//вывод данных

}

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

{

SqlDataAdapter adapter = new SqlDataAdapter();

DataTable table = new DataTable();

string querry = $"Select \* from inventar ORDER BY id";//сортировка по id по возрастанию

SqlCommand Chengecommand = new SqlCommand(querry, dataBase.getSqlConnection());

adapter.SelectCommand = Chengecommand;

adapter.Fill(table);

dataGridView4.DataSource = table;//вывод данных

}

}

}