Приложение Б

Исходный текст программы

Authorization.xaml

using MySql. Data. MySqlClient;

using System;

using System. Collections. Generic;

using System. Data;

using System. Linq;

using System. Text;

using System. Threading. Tasks;

using System. Windows;

using System. Windows. Controls;

using System. Windows. Data;

using System. Windows. Documents;

using System. Windows. Input;

using System. Windows. Media;

using System. Windows. Media. Imaging;

using System. Windows. Shapes;

namespace jewelryworkshop. Windows

{

public partial class Authorization : Window

{

public Authorization()

{

InitializeComponent();

Classes. ConstantData. authorization = this;

Classes. DB. connection. Open();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

if (tbLogin. Text. Length > 0)

{

if (tbPass. Password. Length > 0)

{

DataTable dt = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter("SELECT full\_name, id\_post, id\_employee\_status, id\_employee FROM employee WHERE login = '" + tbLogin. Text + "' AND password = '" + tbPass. Password + "'", Classes. DB. connection);

adapter. Fill(dt);

if (dt. Rows. Count > 0 && dt. Rows[0][2]. ToString() == "1")

{

if (dt. Rows[0][1]. ToString() == "1")

{

Hide();

Admin\_menu admin\_Menu = new Admin\_menu();

admin\_Menu. lbName. Content = $"{dt. Rows[0][0]. ToString()} (Менеджер)";

Classes. ConstantData. user\_id = dt. Rows[0][3]. ToString();

admin\_Menu. Show();

tbLogin. Text = "";

tbPass. Password = "";

}

else if (dt. Rows[0][1]. ToString() == "2")

Продолжение приложения Б

{

Hide();

Jeweler\_orders jeweler\_Orders = new Jeweler\_orders();

jeweler\_Orders. lbName. Content = $"{dt. Rows[0][0]. ToString()} (Ювелир)";

jeweler\_Orders. Show();

tbLogin. Text = "";

tbPass. Password = "";

}

else if (dt. Rows[0][1]. ToString() == "3")

{

Hide();

Cashier\_products cashier\_Products = new Cashier\_products();

cashier\_Products. lbName. Content = $"{dt. Rows[0][0]. ToString()} (Кассир)";

Classes. ConstantData. user\_fio = dt. Rows[0][0]. ToString();

Classes. ConstantData. user\_id = dt. Rows[0][3]. ToString();

cashier\_Products. Show();

tbLogin. Text = "";

tbPass. Password = "";

}

}

else MessageBox. Show("Такого пользователя не существует! ");

}

else MessageBox. Show("Введите пароль! ");

}

else MessageBox. Show("Введите логин! ");

}

private void Window\_Closed(object sender, EventArgs e)

{

Application. Current. Shutdown();

}

}

}

Cashier\_products.xaml.cs

using jewelryworkshop. UserControls;

using MySql. Data. MySqlClient;

namespace jewelryworkshop. Windows

{

public partial class Cashier\_products : Window

{

string Filter = "", Sort = "";

public Cashier\_products()

{

InitializeComponent();

Classes. ConstantData. cashier\_products = this;

btnBad. Badge = "0";

Upd();

}

public void Upd()

{

wPanel. Children. Clear();

Продолжение приложения Б

MySqlDataAdapter adapter = new MySqlDataAdapter($"select id\_product, product\_title, product\_price, quantity, product\_weight, photo, category. category\_title, material. material\_title, id\_order from product, category, material where product. id\_category = category. id\_category and product. id\_material = material. id\_material and product\_title LIKE '%{tbSearch. Text}%' {Filter} {Sort}", Classes. DB. connection);

DataTable table = new DataTable();

adapter. Fill(table);

for (int i = 0; i < table. Rows. Count; i++)

{

wPanel. Children. Add(new Cashier\_product(table. Rows[i][0]. ToString()));

}

}

private void btnBasket\_Click(object sender, RoutedEventArgs e)

{

Cashier\_busket cashier\_Busket = new Cashier\_busket();

cashier\_Busket. ShowDialog();

}

private void btnChangeUser\_Click(object sender, RoutedEventArgs e)

{

Close();

Classes. ConstantData. quantity\_products. Clear();

Classes. ConstantData. id\_products. Clear();

Classes. ConstantData. id\_busket\_product = "";

Classes. ConstantData. authorization. Show();

}

private void Window\_Closed(object sender, EventArgs e)

{

Classes. ConstantData. quantity\_products. Clear();

Classes. ConstantData. id\_products. Clear();

Classes. ConstantData. id\_busket\_product = "";

}

private void tbSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

Upd();

}

private void cbFilter\_DropDownClosed(object sender, EventArgs e)

{

if (cbFilter. SelectedIndex == 1)

{

Filter = $" and id\_order ! = 'null' ";

Upd();

}

else if (cbFilter. SelectedIndex == 2)

{

Filter = $" and id\_order is null ";

Upd();

}

else

{

Filter = "";

Upd();

}

Продолжение приложения Б

}

private void rbWithout\_Checked(object sender, RoutedEventArgs e)

{

if (rbWithout. IsChecked == true)

{

Sort = "";

Upd();

}

}

private void rbAsc\_Checked(object sender, RoutedEventArgs e)

{

if (rbAsc. IsChecked == true)

{

Sort = " order by product\_price asc";

Upd();

}

}

private void rbDesc\_Checked(object sender, RoutedEventArgs e)

{

if (rbDesc. IsChecked == true)

{

Sort = " order by product\_price desc";

Upd();

}

}

private void btnAddOrder\_Click(object sender, RoutedEventArgs e)

{

Cashier\_addOrders cashier\_AddOrders = new Cashier\_addOrders();

cashier\_AddOrders. ShowDialog();

}

private void btnClients\_Click(object sender, RoutedEventArgs e)

{

Cashier\_clients cashier\_Clients = new Cashier\_clients();

cashier\_Clients. btnDeleteClient. Visibility = Visibility. Hidden;

cashier\_Clients. ShowDialog();

}

private void btnCertificate\_Click(object sender, RoutedEventArgs e)

{

Certificate certificate = new Certificate();

certificate. ShowDialog();

}

private void btnClientsProduct\_Click(object sender, RoutedEventArgs e)

{

Cashier\_clientsOldProduct cashier\_ClientsOldProduct = new Cashier\_clientsOldProduct();

cashier\_ClientsOldProduct. ShowDialog();

}

private void btnClearBusket\_Click(object sender, RoutedEventArgs e)

{

btnBad. Badge = "0";

Classes. ConstantData. quantity\_products. Clear();

Classes. ConstantData. id\_products. Clear();

Продолжение приложения Б

Classes. ConstantData. id\_busket\_product = "";

Upd();

}

}

}

Cashier\_product.xaml.cs

namespace jewelryworkshop. UserControls

{

public partial class Cashier\_product : UserControl

{

string ID;

bool est = false;

int idEst = 0;

int idInBusket = 0;

string order\_id = "";

DataTable table = new DataTable();

public Cashier\_product(string ID)

{

InitializeComponent();

this. ID = ID;

MySqlDataAdapter adapter = new MySqlDataAdapter($"select id\_product, product\_title, product\_price, quantity, product\_weight, photo, category. category\_title, material. material\_title, id\_order from product, category, material where product. id\_category = category. id\_category and product. id\_material = material. id\_material and id\_product = {ID}", Classes. DB. connection);

adapter. Fill(table);

idProduct. Content = $"{table. Rows[0][0]}";

photoPath. Content = $"{table. Rows[0][5]}";

lbTitle. Content = "" + table. Rows[0][1];

lbCost. Content = "" + table. Rows[0][2] + " ₽";

lbQuantity. Content = "В наличии: " + table. Rows[0][3] + " шт. ";

if (Classes. ConstantData. id\_products. Count ! = 0)

{

for (int i = 0; i < Classes. ConstantData. id\_products. Count; i++)

{

for (int j = 0; j < table. Rows. Count; j++)

{

if (Classes. ConstantData. id\_products[i]. ToString() == table. Rows[j][0]. ToString())

{

lbQuantity. Content = "В наличии: " + (int. Parse(table. Rows[j][3]. ToString()) - Classes. ConstantData. quantity\_products[i]) + " шт. ";

break;

}

}

}

}

else lbQuantity. Content = "В наличии: " + table. Rows[0][3] + " шт. ";

lbWeight. Content = $"{table. Rows[0][4]} гр. ";

lbCategory. Content = $"{table. Rows[0][6]}";

lbMaterial. Content = $"{table. Rows[0][7]}";

if (table. Rows[0][8]. ToString() ! = "")

{

Продолжение приложения Б

lbOrder. Content = $"No заказа: {table. Rows[0][8]}";

order\_id = table. Rows[0][8]. ToString();

}

try

{

img. ImageSource = new BitmapImage(new Uri($@"{table. Rows[0][5]}", UriKind. Absolute));

}

catch

{

img. ImageSource = new BitmapImage(new Uri(@"C: \Users\furuh\source\repos\jewelryworkshop\jewelryworkshop\Images\picture. png", UriKind. Relative));

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

string newKolvo = lbQuantity. Content. ToString(). Substring(10);

newKolvo = newKolvo. Substring(0, newKolvo. Length - 4);

if (lbOrder. Content. ToString() ! = " ")

{

MessageBoxResult messageBoxResult = MessageBox. Show("Это изделие изготовлено на заказ, вы уверены, что хотите добавить его в корзину? ", "! ", MessageBoxButton. YesNo);

if (messageBoxResult == MessageBoxResult. Yes)

{

if (tbQuantity. Text == string. Empty)

{

if (int. Parse(newKolvo) >= 1)

{

if (Classes. ConstantData. id\_products. Count == 0)

{

Classes. ConstantData. id\_products. Add(ID);

Classes. ConstantData. quantity\_products. Add(1);

lbQuantity. Content = $"В наличии: {(int. Parse(newKolvo) - 1)} шт. ";

Classes. ConstantData. cashier\_products. btnBad. Badge = int. Parse(Classes. ConstantData. cashier\_products. btnBad. Badge. ToString()) + 1;

Classes. ConstantData. id\_orders. Add(order\_id);

}

else

{

for (int i = 0; i < Classes. ConstantData. id\_products. Count; i++)

{

if (Classes. ConstantData. id\_products[i] == ID)

{

est = true;

idEst = i;

break;

}

}

if (est)

{

Classes. ConstantData. quantity\_products[idEst] += 1;

lbQuantity. Content = $"В наличии: {(int. Parse(newKolvo) - 1)} шт. ";

Продолжение приложения Б

}

else

{

Classes. ConstantData. id\_products. Add(ID);

Classes. ConstantData. quantity\_products. Add(1);

lbQuantity. Content = $"В наличии: {(int. Parse(newKolvo) - 1)} шт. ";

Classes. ConstantData. cashier\_products. btnBad. Badge = int. Parse(Classes. ConstantData. cashier\_products. btnBad. Badge. ToString()) + 1;

Classes. ConstantData. id\_orders. Add(order\_id);

}

}

}

else

{

MessageBox. Show("Данного изделия нет в наличии! ");

}

}

else

{

if (tbQuantity. Text ! = "0")

{

if (int. Parse(newKolvo) >= int. Parse(tbQuantity. Text. ToString()))

{

if (Classes. ConstantData. id\_products. Count == 0)

{

Classes. ConstantData. id\_products. Add(ID);

Classes. ConstantData. quantity\_products. Add(int. Parse(tbQuantity. Text. ToString()));

lbQuantity. Content = $"В наличии: {(int. Parse(newKolvo) - int. Parse(tbQuantity. Text. ToString()))} шт. ";

Classes. ConstantData. cashier\_products. btnBad. Badge = int. Parse(Classes. ConstantData. cashier\_products. btnBad. Badge. ToString()) + 1;

tbQuantity. Text = string. Empty;

Classes. ConstantData. id\_orders. Add(order\_id);

}

else

{

for (int i = 0; i < Classes. ConstantData. id\_products. Count; i++)

{

if (Classes. ConstantData. id\_products[i] == ID)

{

est = true;

idEst = i;

break;

}

}

if (est)

{

Classes. ConstantData. quantity\_products[idEst] += int. Parse(tbQuantity. Text. ToString());

lbQuantity. Content = $"В наличии: {(int. Parse(newKolvo) - int. Parse(tbQuantity. Text. ToString()))} шт. ";

tbQuantity. Text = string. Empty;

}

else

Продолжение приложения Б

{

Classes. ConstantData. id\_products. Add(ID);

Classes. ConstantData. quantity\_products. Add(int. Parse(tbQuantity. Text. ToString()));

lbQuantity. Content = $"В наличии: {(int. Parse(newKolvo) - int. Parse(tbQuantity. Text. ToString()))} шт. ";

Classes. ConstantData. cashier\_products. btnBad. Badge = int. Parse(Classes. ConstantData. cashier\_products. btnBad. Badge. ToString()) + 1;

Classes. ConstantData. id\_orders. Add(order\_id);

tbQuantity. Text = string. Empty;

}

}

}

else

{

if (int. Parse(tbQuantity. Text. ToString()) > int. Parse(newKolvo) && (int. Parse(newKolvo) ! = 0))

{

MessageBox. Show("Невозможно добавить! В наличии нету такого количества изделий! ");

tbQuantity. Text = string. Empty;

}

else

{

MessageBox. Show("Данного изделия нет в наличии! ");

tbQuantity. Text = string. Empty;

}

}

}

else MessageBox. Show("Вы ввели некорректное значение в поле 'Количество'");

}

}

}

else

{

if (tbQuantity. Text == string. Empty)

{

if (int. Parse(newKolvo) >= 1)

{

if (Classes. ConstantData. id\_products. Count == 0)

{

Classes. ConstantData. id\_products. Add(ID);

Classes. ConstantData. quantity\_products. Add(1);

lbQuantity. Content = $"В наличии: {(int. Parse(newKolvo) - 1)} шт. ";

Classes. ConstantData. cashier\_products. btnBad. Badge = int. Parse(Classes. ConstantData. cashier\_products. btnBad. Badge. ToString()) + 1;

}

else

{

for (int i = 0; i < Classes. ConstantData. id\_products. Count; i++)

{

if (Classes. ConstantData. id\_products[i] == ID)

{

est = true;

idEst = i;

Продолжение приложения Б

break;

}

}

if (est)

{

Classes. ConstantData. quantity\_products[idEst] += 1;

lbQuantity. Content = $"В наличии: {(int. Parse(newKolvo) - 1)} шт. ";

}

else

{

Classes. ConstantData. id\_products. Add(ID);

Classes. ConstantData. quantity\_products. Add(1);

lbQuantity. Content = $"В наличии: {(int. Parse(newKolvo) - 1)} шт. ";

Classes. ConstantData. cashier\_products. btnBad. Badge = int. Parse(Classes. ConstantData. cashier\_products. btnBad. Badge. ToString()) + 1;

}

}

}

else

{

MessageBox. Show("Данного изделия нет в наличии! ");

}

}

else

{

if (tbQuantity. Text ! = "0")

{

if (int. Parse(newKolvo) >= int. Parse(tbQuantity. Text. ToString()))

{

if (Classes. ConstantData. id\_products. Count == 0)

{

Classes. ConstantData. id\_products. Add(ID);

Classes. ConstantData. quantity\_products. Add(int. Parse(tbQuantity. Text. ToString()));

lbQuantity. Content = $"В наличии: {(int. Parse(newKolvo) - int. Parse(tbQuantity. Text. ToString()))} шт. ";

Classes. ConstantData. cashier\_products. btnBad. Badge = int. Parse(Classes. ConstantData. cashier\_products. btnBad. Badge. ToString()) + 1;

tbQuantity. Text = string. Empty;

}

else

{

for (int i = 0; i < Classes. ConstantData. id\_products. Count; i++)

{

if (Classes. ConstantData. id\_products[i] == ID)

{

est = true;

idEst = i;

break;

}

}

if (est)

{

Продолжение приложения Б

Classes. ConstantData. quantity\_products[idEst] += int. Parse(tbQuantity. Text. ToString());

lbQuantity. Content = $"В наличии: {(int. Parse(newKolvo) - int. Parse(tbQuantity. Text. ToString()))} шт. ";

tbQuantity. Text = string. Empty;

}

else

{

Classes. ConstantData. id\_products. Add(ID);

Classes. ConstantData. quantity\_products. Add(int. Parse(tbQuantity. Text. ToString()));

lbQuantity. Content = $"В наличии: {(int. Parse(newKolvo) - int. Parse(tbQuantity. Text. ToString()))} шт. ";

Classes. ConstantData. cashier\_products. btnBad. Badge = int. Parse(Classes. ConstantData. cashier\_products. btnBad. Badge. ToString()) + 1;

tbQuantity. Text = string. Empty;

}

}

}

else

{

if (int. Parse(tbQuantity. Text. ToString()) > int. Parse(newKolvo) && (int. Parse(newKolvo) ! = 0))

{

MessageBox. Show("Невозможно добавить! В наличии нету такого количества изделий! ");

tbQuantity. Text = string. Empty;

}

else

{

MessageBox. Show("Данного изделия нет в наличии! ");

tbQuantity. Text = string. Empty;

}

}

}

else MessageBox. Show("Вы ввели некорректное значение в поле 'Количество'");

}

}

est = false;

}

private void tbQuantity\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (! Char. IsDigit(e. Text, 0)) e. Handled = true;

}

}

}

Certificate.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Certificate : Window

{

public Certificate()

{

InitializeComponent();

Classes. ConstantData. certificate = this;

private void btnPrint\_Click(object sender, RoutedEventArgs e)

{

Продолжение приложения Б

bool est = false;

if (tbCost. Text ! = string. Empty && int. Parse(tbCost. Text) > 0)

{

if (int. Parse(tbCost. Text) >= 1000)

{

for (int i = 0; i < cbFilter. Items. Count; i++)

{

if (cbFilter. Text == cbFilter. Items[i]. ToString())

{

PrintDialog printDialog = new PrintDialog();

if (printDialog. ShowDialog() == true)

{

printDialog. PrintVisual(gridCertificate, "Печать");

}

est = true;

break;

}

}

if (! est)

{

MessageBox. Show("Такого клиента не существует! ");

}

}

else MessageBox. Show("Сертификат должен быть на сумму 1000 рублей и более! ");

}

else

{

MessageBox. Show("Заполните пустые поля! ");

}

}

}

}

}

}

Cashier\_clients.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Cashier\_clients : Window

{

public Cashier\_clients()

{

InitializeComponent();

DateTime dateTimeNow = DateTime. Now;

DateTime dateTimeMin = dateTimeNow. AddYears(-52);

dpDate. DisplayDateStart = dateTimeMin;

dpDate. DisplayDateEnd = DateTime. Now. AddYears(-18);

dpDate. Text = dpDate. DisplayDateEnd. ToString();

Upd();

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

Продолжение приложения Б

if (Classes. ConstantData. id\_client == "" && btnSave. Content. ToString() == "Добавить")

{

if (tbClientName. Text == "" || tbClientNumber. Text == "" || dpDate. Text == "")

{

MessageBox. Show("Данные не введены", "Ошибка");

}

else

{

MySqlCommand commandabonent = new MySqlCommand($"Insert Into client (full\_name, phone\_number, birth\_date, email) Values ('{tbClientName. Text}', '{tbClientNumber. Text}', '{dpDate. Text}', '{tbClientsEmail. Text}')", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

MessageBox. Show("Данные добавлены! ");

Upd();

}

}

else if (Classes. ConstantData. id\_client ! = "" && btnSave. Content. ToString() == "Изменить")

{

if (tbClientName. Text == "" || tbClientNumber. Text == "" || dpDate. Text == "")

{

MessageBox. Show("Данные не введены", "Ошибка");

}

else

{

MySqlCommand commandabonent = new MySqlCommand($"Update client set full\_name = '{tbClientName. Text}', phone\_number = '{tbClientNumber. Text}', birth\_date = '{dpDate. Text}', email = '{tbClientsEmail. Text}' where id\_client = '{Classes. ConstantData. id\_client}'", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

MessageBox. Show("Данные изменены! ");

Upd();

}

}

Classes. ConstantData. id\_client = "";

btnSave. Content = "Добавить";

tbClientName. Text = "";

tbClientNumber. Text = "";

dpDate. Text = "";

}

private void btnDeleteClient\_Click(object sender, RoutedEventArgs e)

{

try

{

MySqlCommand commandabonent = new MySqlCommand($"delete from client where id\_client = {Classes. ConstantData. id\_client}", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

Upd();

}

catch

{

Продолжение приложения Б

MySqlDataAdapter adapter = new MySqlDataAdapter($"SELECT id\_clients\_discount\_card, discount\_card. discount\_card\_title, discount\_card. discount\_card\_amount FROM clients\_discount\_card, discount\_card where clients\_discount\_card. id\_discount\_card = discount\_card. id\_discount\_card and id\_client = '{Classes. ConstantData. id\_client}'", Classes. DB. connection);

DataTable table = new DataTable();

adapter. Fill(table);

bool delete = false;

for (int i = 0; i < table. Rows. Count; i++)

{

if (table. Rows[i][1]. ToString() == "Скидочная карта")

{

delete = false;

break;

}

else if (table. Rows[i][1]. ToString() == "Сертификат" && table. Rows[i][2]. ToString() ! = "0")

{

delete = false;

break;

}

else delete = true;

}

if (delete)

{

MySqlCommand commandabonent = new MySqlCommand($"delete from clients\_discount\_card where id\_client = {Classes. ConstantData. id\_client}", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

commandabonent = new MySqlCommand($"delete from client where id\_client = {Classes. ConstantData. id\_client}", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

dgCards. ItemsSource = null;

Upd();

}

else

{

MessageBox. Show("Невозможно удалить данного клиента, так как у него остались неиспользованные сертификаты! ");

}

}

}

Cashier\_addOrders.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Cashier\_addOrders : Window

{

public Cashier\_addOrders()

{

InitializeComponent();

DateTime dateTimeNow = DateTime. Now;

DateTime dateTimeMin = dateTimeNow. AddYears(-52);

dpDate. DisplayDateStart = dateTimeMin;

dpDate. DisplayDateEnd = DateTime. Now. AddYears(-18);

Продолжение приложения Б

dpDate. Text = dpDate. DisplayDateEnd. ToString();

cbCategory. Items. Clear();

cbMaterial. Items. Clear();

cbPhoneNumber. Items. Clear();

MySqlDataAdapter adapter1 = new MySqlDataAdapter($"select phone\_number from `client`", Classes. DB. connection);

DataTable table1 = new DataTable();

adapter1. Fill(table1);

for (int i = 0; i < table1. Rows. Count; i++)

{

cbPhoneNumber. Items. Add(table1. Rows[i][0]. ToString());

}

MySqlDataAdapter adapter = new MySqlDataAdapter($"select category\_title from category", Classes. DB. connection);

DataTable table = new DataTable();

adapter. Fill(table);

for (int i = 0; i < table. Rows. Count; i++)

{

cbCategory. Items. Add(table. Rows[i][0]. ToString());

}

adapter = new MySqlDataAdapter($"select material\_title from material", Classes. DB. connection);

table = new DataTable();

adapter. Fill(table);

for (int i = 0; i < table. Rows. Count; i++)

{

cbMaterial. Items. Add(table. Rows[i][0]. ToString());

}

}

void ClientSelect()

{

try

{

MySqlDataAdapter adapter = new MySqlDataAdapter($"select full\_name, birth\_date from `client` where `client`. phone\_number = '{cbPhoneNumber. Text}'", Classes. DB. connection);

DataTable table = new DataTable();

adapter. Fill(table);

tbClientsName. Text = table. Rows[0][0]. ToString();

dpDate. Text = table. Rows[0][1]. ToString();

}

catch { }

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

if (tbComment. Text == "" || tbClientsName. Text == "" || cbPhoneNumber. Text == "" || cbMaterial. Text == "" || cbCategory. Text == "" || dpDate. Text == "")

{

MessageBox. Show("Данные не введены", "Ошибка");

}

else

{

DateTime dateTime = DateTime. Parse(dpDate. Text);

Продолжение приложения Б

string dt = dateTime. ToString("yyyy/MM/dd");

MySqlDataAdapter adapter = new MySqlDataAdapter($"SELECT id\_client FROM jewelry. client where phone\_number = '{cbPhoneNumber. Text}'", Classes. DB. connection);

DataTable table = new DataTable();

adapter. Fill(table);

if (table. Rows. Count ! = 0)

{

MySqlCommand orderCommand = new MySqlCommand($"Insert Into jewelry. order (comment, id\_order\_status, id\_client\_order) Values ('{cbCategory. Text}; {cbMaterial. Text}; {tbComment. Text}', '1', '{table. Rows[0][0]. ToString()}')", Classes. DB. connection);

orderCommand. ExecuteNonQuery();

}

else

{

MySqlCommand clientCommand = new MySqlCommand($"Insert Into jewelry. client (full\_name, phone\_number, birth\_date) Values ('{tbClientsName. Text}', '{cbPhoneNumber. Text}', '{dt}')", Classes. DB. connection);

clientCommand. ExecuteNonQuery();

adapter = new MySqlDataAdapter($"SELECT LAST\_INSERT\_ID()", Classes. DB. connection);

table = new DataTable();

adapter. Fill(table);

MySqlCommand orderCommand = new MySqlCommand($"Insert Into jewelry. order (comment, id\_order\_status, id\_client\_order) Values ('{cbCategory. Text}; {cbMaterial. Text}; {tbComment. Text}', '1', '{table. Rows[0][0]. ToString()}')", Classes. DB. connection);

orderCommand. ExecuteNonQuery();

}

MessageBox. Show("Данные добавлены! ");

}

}

}

}

}

Cashier\_addClient.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Cashier\_addClient : Window

{

public Cashier\_addClient()

{

InitializeComponent();

DateTime dateTimeNow = DateTime. Now;

DateTime dateTimeMin = dateTimeNow. AddYears(-52);

dpDate. DisplayDateStart = dateTimeMin;

dpDate. DisplayDateEnd = DateTime. Now. AddYears(-18);

dpDate. Text = dpDate. DisplayDateEnd. ToString();

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

if (tbClientsName. Text ! = string. Empty && tbClientsNumber. Text ! = string. Empty && dpDate. Text ! = string. Empty)

{

DateTime dateTime = DateTime. Parse(dpDate. Text);

Продолжение приложения Б

string dt = dateTime. ToString("yyyy/MM/dd");

MySqlDataAdapter adapterr = new MySqlDataAdapter($"SELECT phone\_number, full\_name from client where phone\_number = '{tbClientsNumber. Text}'", Classes. DB. connection);

DataTable tablee = new DataTable();

adapterr. Fill(tablee);

if (tablee. Rows. Count == 0)

{

MySqlCommand commandabonent = new MySqlCommand($"Insert Into client (full\_name, phone\_number, birth\_date, email) Values ('{tbClientsName. Text}', '{tbClientsNumber. Text}', '{dt}', '{tbClientsEmail. Text}')", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

if (Classes. ConstantData. addClient\_fromBusket == true)

{

Classes. ConstantData. cashier\_Busket. Upd();

Classes. ConstantData. cashier\_Busket. LoadData();

Classes. ConstantData. cashier\_Busket. ClientNumbers();

MySqlDataAdapter adapter = new MySqlDataAdapter($"SELECT LAST\_INSERT\_ID()", Classes. DB. connection);

DataTable table = new DataTable();

adapter. Fill(table);

adapter = new MySqlDataAdapter($"SELECT full\_name, phone\_number from client where id\_client = '{table. Rows[0][0]. ToString()}'", Classes. DB. connection);

DataTable table1 = new DataTable();

adapter. Fill(table1);

Classes. ConstantData. cashier\_Busket. tbClientsName. Text = table1. Rows[0][0]. ToString();

Classes. ConstantData. cashier\_Busket. cbPhoneNumber. Text = table1. Rows[0][1]. ToString();

}

else if (Classes. ConstantData. addClient\_fromCertificate == true)

{

MySqlDataAdapter adapter = new MySqlDataAdapter($"SELECT LAST\_INSERT\_ID()", Classes. DB. connection);

DataTable table = new DataTable();

adapter. Fill(table);

adapter = new MySqlDataAdapter($"SELECT full\_name from client where id\_client = '{table. Rows[0][0]. ToString()}'", Classes. DB. connection);

DataTable table1 = new DataTable();

adapter. Fill(table1);

Classes. ConstantData. certificate. cbFilter. Text = table1. Rows[0][0]. ToString();

Classes. ConstantData. certificate. Upd();

}

else if (Classes. ConstantData. addClient\_fromOldProduct == true)

{

MySqlDataAdapter adapter = new MySqlDataAdapter($"SELECT LAST\_INSERT\_ID()", Classes. DB. connection);

DataTable table = new DataTable();

adapter. Fill(table);

adapter = new MySqlDataAdapter($"SELECT full\_name, phone\_number from client where id\_client = '{table. Rows[0][0]. ToString()}'", Classes. DB. connection);

DataTable table1 = new DataTable();

adapter. Fill(table1);

Classes. ConstantData. cashier\_ClientsOldProduct. tbClientsName. Text = table1. Rows[0][0]. ToString();

Classes. ConstantData. cashier\_ClientsOldProduct. UpdCB();

Classes. ConstantData. cashier\_ClientsOldProduct. cbPhoneNumber. Text = table1. Rows[0][1]. ToString();

}

Продолжение приложения Б

MessageBox. Show("Данные добавлены! ");

this. Close();

}

else

MessageBox. Show("Клеинт с таким номером телефона уже существует! ");

}

else

{

MessageBox. Show("Заполните все ключчевые поля! ");

}

}

}

}

}

Cashier\_busket.xaml.cs

using Microsoft. Office. Interop. Excel;

using Excel = Microsoft. Office. Interop. Excel;

using DataTable = System. Data. DataTable;

using Window = System. Windows. Window;

using MaterialDesignThemes. Wpf;

namespace jewelryworkshop. Windows

{

public partial class Cashier\_busket : Window

{

string ids\_product = "";

string idCurCard = "";

int sumCost = 0;

int selectedRow = 0;

int discountSum = 0;

int curDiscount = 10;

int curSumCard = 0;

int curAmountCard = 0;

int usedCertSum = 0;

string idClient = "";

string curNum = "";

bool discount = false;

bool est = false;

bool certificate = false;

bool addDiscountCard = false;

bool used\_card = false;

bool haveCard = false;

public static List<string> id\_card = new List<string>();

public static List<string> amount\_card = new List<string>();

public static List<string> before\_amount\_card = new List<string>();

DataTable table = new DataTable();

DataTable tablePhoneCard = new DataTable();

DataTable tableNums = new DataTable();

public Cashier\_busket()

{

InitializeComponent();

Classes. ConstantData. cashier\_Busket = this;

Продолжение приложения Б

Upd();

LoadData();

ClientUpdate();

ClientNumbers();

}

private void dgBusket\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

try

{

DataRowView row = (DataRowView)dgBusket. SelectedItems[0];

selectedRow = dgBusket. SelectedIndex;

Classes. ConstantData. id\_busket\_product = row[0]. ToString();

tbQuantity. Text = row[5]. ToString();

tbCost. Text = row[3]. ToString();

}

catch { }

}

private void btnDelete\_Click(object sender, RoutedEventArgs e)

{

bool deleted = false;

for (int i = 0; i < Classes. ConstantData. id\_products. Count; i++)

{

if (Classes. ConstantData. id\_busket\_product == Classes. ConstantData. id\_products[i])

{

Classes. ConstantData. id\_products. RemoveAt(i);

Classes. ConstantData. quantity\_products. RemoveAt(i);

tbQuantity. Text = "";

tbCost. Text = "";

Upd();

deleted = true;

}

}

if (! deleted)

{

MessageBox. Show("Выберите украшение для удаления! ");

deleted = false;

}

}

private void btnEdit\_Click(object sender, RoutedEventArgs e)

{

if (tbQuantity. Text == "0")

{

btnDelete\_Click(sender, e);

}

else

{

try

{

if (Classes. ConstantData. id\_busket\_product ! = "")

{

int idInList = 0;

Продолжение приложения Б

for (int i = 0; i < Classes. ConstantData. id\_products. Count; i++)

{

if (Classes. ConstantData. id\_busket\_product == Classes. ConstantData. id\_products[i])

{

idInList = i;

}

}

if (int. Parse(tbQuantity. Text) > int. Parse(table. Rows[selectedRow][2]. ToString()))

{

MessageBox. Show("Невозможно изменить! В наличии нету такого количества изделий! ");

}

else

{

Classes. ConstantData. quantity\_products[idInList] = int. Parse(tbQuantity. Text. ToString());

Upd();

if (cbPhoneNumber. Text == string. Empty)

{

tbClientsName. Text = string. Empty;

usedCertSum = 0;

Upd();

discount = false;

}

else

ClientSelect();

}

tbCost. Text = "";

tbQuantity. Text = "";

Classes. ConstantData. id\_busket\_product = "";

}

else

{

MessageBox. Show("Выберите украшение для изменения количества! ");

tbQuantity. Text = string. Empty;

}

}

catch { }

}

}

}

private void btnPrint\_Click(object sender, RoutedEventArgs e)

{

bool clientSelected = false;

if (dgBusket. ItemsSource ! = null)

{

if (int. Parse(tbSumCost. Text) >= 10000)

{

if (! discount)

{

MessageBoxResult messageBoxResult1 = MessageBox. Show($"Добавить бессрочную скидочную карту на 10% за 1000 рублей? ", "! ", MessageBoxButton. YesNo);

if (messageBoxResult1 == MessageBoxResult. Yes)

Продолжение приложения Б

{

if (cbPhoneNumber. Text. Length == 11 && tbClientsName. Text ! = string. Empty)

{

addDiscountCard = true;

tbSumCost. Text = (int. Parse(tbSumCost. Text) + 1000). ToString();

}

else

{

MessageBox. Show("Выберите клиента! ");

clientSelected = true;

}

}

else addDiscountCard = false;

}

}

try

{

if (! clientSelected)

{

string sum\_Cost = tbSumCost. Text;

Excel. Application excel = new Excel. Application();

Excel. Workbook workbook = excel. Workbooks. Open(@"C: \Users\furuh\Documents\Distant\Diplom\1jReceipt. xlsx");

Excel. Worksheet sheet1 = (Excel. Worksheet)workbook. Sheets[1];

DataTable startTable = new DataTable();

DataTable dataTable = new DataTable();

DataTable receiptTable = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter($"SELECT max(id\_receipt) FROM jewelry. receipt\_tp; ", Classes. DB. connection);

adapter. Fill(dataTable);

startTable = ((DataView)dgBusket. ItemsSource). ToTable();

startTable. Columns. Remove("Название");

startTable. Columns. Remove("Цена (руб. )");

receiptTable = ((DataView)dgBusket. ItemsSource). ToTable();

receiptTable. Columns. Remove("Артикул");

receiptTable. Columns. Remove("Количество в наличии (шт. )");

Range range = (Range)sheet1. Cells[9, 1];

string id\_receipt = (int. Parse(dataTable. Rows[0][0]. ToString()) + 1). ToString();

range. Value2 += $"{(int. Parse(dataTable. Rows[0][0]. ToString()) + 1)}";

range = (Range)sheet1. Cells[10, 4];

range. Value2 = DateTime. Now. ToShortDateString();

for (int i = 0; i < receiptTable. Rows. Count; i++)

{

for (int j = 0; j < receiptTable. Columns. Count; j++)

{

range = (Range)sheet1. Cells[i + 12, j + 1];

range. Borders. LineStyle = XlLineStyle. xlContinuous;

range. HorizontalAlignment = XlHAlign. xlHAlignCenter;

range. VerticalAlignment = XlVAlign. xlVAlignCenter;

range. Value2 = receiptTable. Rows[i][j]. ToString();

}

Продолжение приложения Б

}

int addDiscount = 0;

if (addDiscountCard)

{

for (int i = 1; i < 5; i++)

{

addDiscount = 1;

Range myRange1 = (Range)sheet1. Cells[receiptTable. Rows. Count + 12, i];

myRange1. Borders. LineStyle = XlLineStyle. xlContinuous;

myRange1. HorizontalAlignment = XlHAlign. xlHAlignCenter;

myRange1. VerticalAlignment = XlVAlign. xlVAlignCenter;

if (i == 1)

myRange1. Value2 = $"Скидочная карта (10%)";

else if (i == 2 || i == 3)

myRange1. Value2 = $"1000";

else myRange1. Value2 = "1";

myRange1. Font. Name = "Roboto Light";

}

}

int certCount = 0;

if (certificate)

{

for (int i = 0; i < id\_card. Count; i++)

{

Range myRange1 = sheet1. get\_Range((Range)sheet1. Cells[receiptTable. Rows. Count + 12 + i + addDiscount, 1], (Range)sheet1. Cells[receiptTable. Rows. Count + 12 + i + addDiscount, 4]);

myRange1. Merge();

myRange1. Borders. LineStyle = XlLineStyle. xlContinuous;

myRange1. HorizontalAlignment = XlHAlign. xlHAlignCenter;

myRange1. VerticalAlignment = XlVAlign. xlVAlignCenter;

if (int. Parse(before\_amount\_card[i]) - int. Parse(amount\_card[i]) ! = 0)

{

myRange1. Value2 = $"Использован сертификат No{id\_card[i]} на сумму {int. Parse(before\_amount\_card[i]) - int. Parse(amount\_card[i])} рублей";

}

else myRange1. Value2 = $"Использован сертификат No{id\_card[i]} на сумму {int. Parse(amount\_card[i])} рублей";

myRange1. Font. Name = "Roboto Light";

certCount++;

}

}

Range range1 = sheet1. get\_Range((Range)sheet1. Cells[receiptTable. Rows. Count + 13 + certCount + addDiscount, 1], (Range)sheet1. Cells[receiptTable. Rows. Count + 13 + certCount + addDiscount, 4]);

range1. Merge();

range1. Font. Name = "Roboto Light";

range1. Value2 = $"Кассир: {Classes. ConstantData. user\_fio}";

range1 = (Range)sheet1. Cells[receiptTable. Rows. Count + 12 + certCount + addDiscount, 3];

if (discount)

{

range1 = sheet1. get\_Range((Range)sheet1. Cells[receiptTable. Rows. Count + 12 + certCount + addDiscount, 2], (Range)sheet1. Cells[receiptTable. Rows. Count + 12 + certCount + addDiscount, 3]);

Продолжение приложения Б

range1. Merge();

range1. Value2 = $"Итого (со скидкой {curDiscount}%): {sum\_Cost} руб. ";

}

else range1. Value2 = $"Итого: {sum\_Cost} руб. ";

range1. Font. Name = "Roboto Light";

range1. HorizontalAlignment = XlHAlign. xlHAlignCenter;

range1. VerticalAlignment = XlVAlign. xlVAlignCenter;

Range myRange = sheet1. get\_Range((Range)sheet1. Cells[receiptTable. Rows. Count + 14 + certCount + addDiscount, 1], (Range)sheet1. Cells[receiptTable. Rows. Count + 14 + certCount + addDiscount, 4]);

myRange. Merge();

myRange. Font. Name = "Roboto Light";

myRange. HorizontalAlignment = XlHAlign. xlHAlignCenter;

myRange. Font. Italic = true;

myRange. Value2 = $"Спасибо за покупку! ";

Range myRange6 = sheet1. UsedRange;

myRange6. Columns. AutoFit();

if (id\_card. Count ! = 0)

{

for (int i = 0; i < id\_card. Count; i++)

{

MySqlCommand SqlCommand;

SqlCommand = new MySqlCommand($"update discount\_card set discount\_card\_amount = '{amount\_card[i]}' where id\_discount\_card = ({id\_card[i]})", Classes. DB. connection);

SqlCommand. ExecuteNonQuery();

}

}

string ids\_orders = "";

for (int i = 0; i < Classes. ConstantData. id\_orders. Count; i++)

{

if (i == 0)

{

ids\_orders = Classes. ConstantData. id\_orders[i];

}

else

{

ids\_orders += $", {Classes. ConstantData. id\_orders[i]}";

}

}

MySqlCommand mySqlCommand;

if (Classes. ConstantData. id\_orders. Count ! = 0)

{

mySqlCommand = new MySqlCommand($"update jewelry. order set id\_order\_status = '3' where id\_order in ({ids\_orders})", Classes. DB. connection);

mySqlCommand. ExecuteNonQuery();

}

for (int i = 0; i < Classes. ConstantData. id\_products. Count; i++)

{

mySqlCommand = new MySqlCommand($"insert into receipt\_tp (id\_receipt, id\_product, id\_employee, receipt\_date, quantity, sum\_cost) values ('{id\_receipt}', '{startTable. Rows[i][0]. ToString()}', '{Classes. ConstantData. user\_id}', '{DateTime. Now. ToString("yyyy-MM-dd")}', '{startTable. Rows[i][3]. ToString()}', '{startTable. Rows[i][2]. ToString()}')", Classes. DB. connection);

Продолжение приложения Б

mySqlCommand. ExecuteNonQuery();

}

for (int i = 0; i < Classes. ConstantData. id\_products. Count; i++)

{

mySqlCommand = new MySqlCommand($"update product set quantity = '{int. Parse(startTable. Rows[i][1]. ToString()) - int. Parse(startTable. Rows[i][3]. ToString())}' where id\_product = '{startTable. Rows[i][0]. ToString()}'", Classes. DB. connection);

mySqlCommand. ExecuteNonQuery();

mySqlCommand = new MySqlCommand($"update product set id\_order = null where id\_product in ({ids\_product})", Classes. DB. connection);

mySqlCommand. ExecuteNonQuery();

}

if (addDiscountCard)

{

mySqlCommand = new MySqlCommand($"Insert Into discount\_card (discount\_card\_title, discount\_card\_percent) Values ('Скидочная карта', '10')", Classes. DB. connection);

mySqlCommand. ExecuteNonQuery();

MySqlDataAdapter adapterCardId = new MySqlDataAdapter($"SELECT LAST\_INSERT\_ID()", Classes. DB. connection);

DataTable tableCardId = new DataTable();

adapterCardId. Fill(tableCardId);

mySqlCommand = new MySqlCommand($"Insert Into clients\_discount\_card (id\_client, id\_discount\_card, create\_date, id\_employee\_cards) Values ('{idClient}', '{tableCardId. Rows[0][0]. ToString()}', '{DateTime. Now. ToString("yyyy-MM-dd")}', '{Classes. ConstantData. user\_id}')", Classes. DB. connection);

mySqlCommand. ExecuteNonQuery();

}

excel. Visible = true;

usedCertSum = 0;

addDiscountCard = false;

this. Close();

}

}

catch

{

MessageBox. Show("Данные клиента введены некорректно! ");

}

}

else MessageBox. Show("Добавьте товары в корзину для составления кассового чека");

}

private void btnUseCert\_Click(object sender, RoutedEventArgs e)

{

if (btnUseCert. Content. ToString() == "Использовать сертификат")

{

btnUseCert. Content = "Готово";

tbCertNum. Visibility = Visibility. Visible;

}

else

{

string usedCertID = tbCertNum. Text. ToString();

try

{

Продолжение приложения Б

MySqlDataAdapter adapter = new MySqlDataAdapter($"select id\_discount\_card, discount\_card\_amount FROM jewelry. discount\_card where discount\_card\_title = 'Сертификат' and id\_discount\_card = '{usedCertID}'", Classes. DB. connection);

DataTable table = new DataTable();

adapter. Fill(table);

for (int i = 0; i < id\_card. Count; i++)

{

if (id\_card. Count ! = 0)

{

if (id\_card[i] == table. Rows[0][0]. ToString())

{

MessageBox. Show("Данный сертификат уже использован! ");

used\_card = true;

}

}

}

if (used\_card == false)

{

if (int. Parse(table. Rows[0][1]. ToString()) ! = 0)

{

MessageBoxResult messageBoxResult = MessageBox. Show($"Использовать сертификат на сумму {table. Rows[0][1]. ToString()} рублей? ", "! ", MessageBoxButton. YesNo);

if (messageBoxResult == MessageBoxResult. Yes)

{

if (int. Parse(tbSumCost. Text) >= int. Parse(table. Rows[0][1]. ToString()))

{

id\_card. Add(table. Rows[0][0]. ToString());

before\_amount\_card. Add(table. Rows[0][1]. ToString());

amount\_card. Add("0");

tbSumCost. Text = ((int. Parse(tbSumCost. Text) - int. Parse(table. Rows[0][1]. ToString())). ToString());

usedCertSum += int. Parse(table. Rows[0][1]. ToString());

}

else

{

id\_card. Add(table. Rows[0][0]. ToString());

before\_amount\_card. Add(table. Rows[0][1]. ToString());

usedCertSum += (int. Parse(table. Rows[0][1]. ToString()) - int. Parse(tbSumCost. Text));

amount\_card. Add((int. Parse(table. Rows[0][1]. ToString()) - int. Parse(tbSumCost. Text)). ToString());

tbSumCost. Text = "0";

}

certificate = true;

}

}

else MessageBox. Show("На данном сертификате не осталось средств! ");

}

}

catch

{

MessageBox. Show("Такого сертификата не существует! ");

}

used\_card = false;

Продолжение приложения Б

tbCertNum. Text = string. Empty;

btnUseCert. Content = "Использовать сертификат";

tbCertNum. Visibility = Visibility. Collapsed;

}

}

private void ComboBox\_TextChanged(object sender, TextChangedEventArgs e)

{

if (cbPhoneNumber. Text == string. Empty)

{

tbClientsName. Text = string. Empty;

usedCertSum = 0;

Upd();

discount = false;

}

else

ClientSelect();

}

public void cbPhoneNumber\_DropDownClosed(object sender, EventArgs e)

{

if (curNum == cbPhoneNumber. Text || curNum == "")

{

curNum = cbPhoneNumber. Text;

ClientSelect();

}

else

{

ClientSelect();

id\_card. Clear();

usedCertSum = 0;

}

curNum = cbPhoneNumber. Text;

}

private void cbPhoneNumber\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

if (! Char. IsDigit(e. Text, 0)) e. Handled = true;

}

}

}

Cashier\_clientsOldProducts.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Cashier\_clientsOldProduct : Window

{

DataTable busket = new DataTable();

int sumCost = 0;

string curNum = "";

string id\_client = string. Empty;

string title = string. Empty;

string weight = string. Empty;

string description = string. Empty;

string cost = string. Empty;

Продолжение приложения Б

public Cashier\_clientsOldProduct()

{

InitializeComponent();

busket. Columns. Add("title");

busket. Columns. Add("weight");

busket. Columns. Add("description");

busket. Columns. Add("cost");

Classes. ConstantData. cashier\_ClientsOldProduct = this;

UpdCB();

}

private void ComboBox\_TextChanged(object sender, TextChangedEventArgs e)

{

if (cbPhoneNumber. Text == string. Empty)

{

dgClientProducts. ItemsSource = null;

dgOldProducts. ItemsSource = null;

tbClientsName. Text = string. Empty;

}

else

LoadDataOldProducts();

}

private void tbProductWeight\_TextChanged(object sender, TextChangedEventArgs e)

{

try

{

tbCost. Text = (int. Parse(tbProductWeight. Text) \* 2000). ToString();

}

catch

{

tbCost. Text = string. Empty;

}

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

if (tbProductTitle. Text ! = string. Empty && tbProductWeight. Text ! = string. Empty && tbClientsName. Text ! = string. Empty && cbPhoneNumber. Text ! = string. Empty)

{

busket. Rows. Add(new object[] { tbProductTitle. Text, tbProductWeight. Text, tbProductDescription. Text, tbCost. Text });

dgOldProducts. ItemsSource = busket. DefaultView;

Sum();

}

else MessageBox. Show("Заполните все поля! ");

}

private void dgOldProducts\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

try

{

DataRowView row = (DataRowView)dgOldProducts. SelectedItems[0];

title = row[0]. ToString();

weight = row[1]. ToString();

Продолжение приложения Б

description = row[2]. ToString();

cost = row[3]. ToString();

}

catch { }

}

private void btnDelete\_Click(object sender, RoutedEventArgs e)

{

for (int i = 0; i < busket. Rows. Count; i++)

{

if (busket. Rows[i][0]. ToString() == title &&

busket. Rows[i][1]. ToString() == weight &&

busket. Rows[i][2]. ToString() == description &&

busket. Rows[i][3]. ToString() == cost)

{

busket. Rows[i]. Delete();

Sum();

break;

}

}

}

private void btnPrint\_Click(object sender, RoutedEventArgs e)

{

if (dgOldProducts. ItemsSource ! = null)

{

Excel. Application excel = new Excel. Application();

Excel. Workbook workbook = excel. Workbooks. Open(@"C: \Users\furuh\Documents\Distant\Diplom\oldProductReceipt. xlsx");

Excel. Worksheet sheet1 = (Excel. Worksheet)workbook. Sheets[1];

Range range = (Range)sheet1. Cells[11, 3];

range. Value2 = DateTime. Now. ToShortDateString();

for (int i = 0; i < busket. Rows. Count; i++)

{

for (int j = 0; j < busket. Columns. Count; j++)

{

if (j == 2)

{

continue;

}

else if (j == 3)

{

range = (Range)sheet1. Cells[i + 13, j];

}

else

{

range = (Range)sheet1. Cells[i + 13, j + 1];

}

range. Borders. LineStyle = XlLineStyle. xlContinuous;

range. HorizontalAlignment = XlHAlign. xlHAlignCenter;

range. VerticalAlignment = XlVAlign. xlVAlignCenter;

range. Font. Name = "Roboto Light";

range. Value2 = busket. Rows[i][j]. ToString();

Продолжение приложения Б

}

}

Range range1 = (Range)sheet1. Cells[busket. Rows. Count + 13, 3];

range1. Value2 = $"Итого: {sumCost} руб. ";

range1. HorizontalAlignment = XlHAlign. xlHAlignCenter;

range1. VerticalAlignment = XlVAlign. xlVAlignCenter;

range1. Font. Name = "Roboto Light";

range1 = sheet1. get\_Range((Range)sheet1. Cells[busket. Rows. Count + 14, 1], (Range)sheet1. Cells[busket. Rows. Count + 14, 3]);

range1. Merge();

range1. Font. Name = "Roboto Light";

range1. Value2 = $"Кассир: {Classes. ConstantData. user\_fio}";

Range myRange = sheet1. get\_Range((Range)sheet1. Cells[busket. Rows. Count + 15, 1], (Range)sheet1. Cells[busket. Rows. Count + 15, 3]);

myRange. Merge();

myRange. Font. Name = "Roboto Light";

myRange. HorizontalAlignment = XlHAlign. xlHAlignCenter;

myRange. Font. Italic = true;

myRange. Value2 = $"Спасибо за покупку! ";

Range myRange6 = sheet1. UsedRange;

myRange6. Columns. AutoFit();

MySqlCommand mySqlCommand;

for (int i = 0; i < busket. Rows. Count; i++)

{

mySqlCommand = new MySqlCommand($"insert into old\_product (old\_product\_title, old\_product\_weight, old\_product\_description, old\_product\_status) values ('{busket. Rows[i][0]. ToString()}', '{busket. Rows[i][1]. ToString()}', '{busket. Rows[i][2]. ToString()}', 'На складе')", Classes. DB. connection);

mySqlCommand. ExecuteNonQuery();

MySqlDataAdapter adapterOldId = new MySqlDataAdapter($"SELECT LAST\_INSERT\_ID()", Classes. DB. connection);

DataTable tableOldId = new DataTable();

adapterOldId. Fill(tableOldId);

mySqlCommand = new MySqlCommand($"insert into clients\_old\_product (id\_old\_product, id\_client\_product, id\_employee\_product, cost, sale\_date) values ('{tableOldId. Rows[0][0]. ToString()}', '{id\_client}', '{Classes. ConstantData. user\_id}', '{busket. Rows[i][3]. ToString()}', '{DateTime. Now. ToString("yyyy-MM-dd")}')", Classes. DB. connection);

mySqlCommand. ExecuteNonQuery();

}

excel. Visible = true;

}

else MessageBox. Show("Добавьте товары для составления кассового чека");

sumCost = 0;

cbPhoneNumber. Text = string. Empty;

tbClientsName. Text = string. Empty;

tbSumCost. Text = string. Empty;

busket. Rows. Clear();

dgOldProducts. ItemsSource = null;

btnClear\_Click(sender, e);

}

}

}

Продолжение приложения Б

Jeweler\_orders.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Jeweler\_orders : Window

{

string Filter = "";

public Jeweler\_orders()

{

InitializeComponent();

Classes. ConstantData. jeweler\_orders = this;

Upd();

dgOrder. SelectedIndex = 0;

}

public void Upd()

{

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter($"SELECT id\_order as 'Код', comment as 'Описание украшения', client. full\_name as 'ФИО клиента', client. phone\_number as 'Номер телефона клиента', order\_status\_title as 'Статус заказа' FROM jewelry. order, order\_status, client where order. id\_order\_status = order\_status. id\_order\_status and order. id\_client\_order = client. id\_client {Filter}", Classes. DB. connection);

adapter. Fill(table);

dgOrder. ItemsSource = table. DefaultView;

}

private void dgOrder\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

try

{

DataRowView row = (DataRowView)dgOrder. SelectedItems[0];

Classes. ConstantData. id\_order = row[0]. ToString();

tbOrderDescription. Text = row[1]. ToString();

tbClientName. Text = row[2]. ToString();

tbClientsNumber. Text = row[3]. ToString();

tbStatus. Text = row[4]. ToString();

}

catch { }

}

private void btnAdd\_Click(object sender, RoutedEventArgs e)

{

if (Classes. ConstantData. id\_order == "")

{

MessageBox. Show("Выберите заказ! ");

}

else if (tbStatus. Text == "Изготовлен" || tbStatus. Text == "Продан")

{

MessageBox. Show("Данное украшение уже было изготовлено");

}

else

{

Jeweler\_addProduct jeweler\_AddProduct = new Jeweler\_addProduct(Classes. ConstantData. id\_order);

jeweler\_AddProduct. ShowDialog();

}

Продолжение приложения Б

}

private void btnChangeUser\_Click(object sender, RoutedEventArgs e)

{

Close();

Classes. ConstantData. authorization. tbLogin. Text = "";

Classes. ConstantData. authorization. tbPass. Password = "";

Classes. ConstantData. id\_order = "";

tbOrderDescription. Text = "";

tbClientName. Text = "";

tbClientsNumber. Text = "";

tbStatus. Text = "";

Classes. ConstantData. authorization. Show();

}

private void btnDelete\_Click(object sender, RoutedEventArgs e)

{

if (tbStatus. Text == "Продан")

{

MySqlCommand commandabonent = new MySqlCommand($"delete from jewelry. order where id\_order = '{Classes. ConstantData. id\_order}'", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

MessageBox. Show("Заказ удален! ");

Upd();

}

else if (tbStatus. Text == "Изготавливается")

MessageBox. Show("Данный заказ нельзя удалить, так как изделие еще не изготовлено! ");

else if (tbStatus. Text == "Изготовлен")

MessageBox. Show("Данный заказ нельзя удалить, так как изделие еще не продано! ");

}

private void btnOldProducts\_Click(object sender, RoutedEventArgs e)

{

Jeweler\_oldProducts jeweler\_OldProducts = new Jeweler\_oldProducts();

jeweler\_OldProducts. ShowDialog();

}

}

}

Jeweler\_addProduct.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Jeweler\_addProduct : Window

{

string ID = "";

public Jeweler\_addProduct(string ID = "")

{

this. ID = ID;

InitializeComponent();

Classes. ConstantData. jeweler\_AddProduct = this;

cbCategory. Items. Clear();

cbMaterial. Items. Clear();

MySqlDataAdapter adapter = new MySqlDataAdapter($"select category\_title from category", Classes. DB. connection);

DataTable table = new DataTable();

Продолжение приложения Б

adapter. Fill(table);

for (int i = 0; i < table. Rows. Count; i++)

{

cbCategory. Items. Add(table. Rows[i][0]. ToString());

}

adapter = new MySqlDataAdapter($"select material\_title from material", Classes. DB. connection);

table = new DataTable();

adapter. Fill(table);

for (int i = 0; i < table. Rows. Count; i++)

{

cbMaterial. Items. Add(table. Rows[i][0]. ToString());

}

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

string categoryId = "", materialId = "";

try

{

MySqlDataAdapter categoryAdapter = new MySqlDataAdapter($"select id\_category from category where category\_title = '{cbCategory. SelectedItem. ToString()}'", Classes. DB. connection);

DataTable categoryTable = new DataTable();

categoryAdapter. Fill(categoryTable);

categoryId = categoryTable. Rows[0][0]. ToString();

MySqlDataAdapter materialAdapter = new MySqlDataAdapter($"select id\_material from material where material\_title = '{cbMaterial. SelectedItem. ToString()}'", Classes. DB. connection);

DataTable materialTable = new DataTable();

materialAdapter. Fill(materialTable);

materialId = materialTable. Rows[0][0]. ToString();

}

catch { }

if (tbTitle. Text == "" || tbCost. Text == "" || tbWeight. Text == "" || cbMaterial. Text == "" || cbCategory. Text == "" || tbLostMaterialWeight. Text == "")

{

MessageBox. Show("Данные не введены", "Ошибка");

}

else

{

string id\_material = (cbMaterial. SelectedIndex + 1). ToString();

MySqlDataAdapter adapter = new MySqlDataAdapter($"select material\_weight from material where id\_material = {id\_material}", Classes. DB. connection);

DataTable table = new DataTable();

adapter. Fill(table);

if ((int. Parse(table. Rows[0][0]. ToString()) - int. Parse(tbLostMaterialWeight. Text)) < 0)

{

MessageBox. Show("Вы ввели некорректное значение в поле потраченного материала! ");

}

else

{

int materialOstatok = (int. Parse(table. Rows[0][0]. ToString()) - int. Parse(tbLostMaterialWeight. Text));

Продолжение приложения Б

MySqlCommand command = new MySqlCommand($"Insert Into product (product\_title, product\_price, quantity, product\_weight, id\_category, id\_order, id\_material) Values ('{tbTitle. Text}', '{tbCost. Text}', '1', '{tbWeight. Text}', '{categoryId}', '{Classes. ConstantData. id\_order}', '{materialId}')", Classes. DB. connection);

command. ExecuteNonQuery();

command = new MySqlCommand($"update jewelry. order set id\_order\_status = '2' where id\_order = {Classes. ConstantData. id\_order}", Classes. DB. connection);

command. ExecuteNonQuery();

command = new MySqlCommand($"update material set material\_weight = '{materialOstatok}' where id\_material = {materialId}", Classes. DB. connection);

command. ExecuteNonQuery();

MessageBox. Show("Данные добавлены! ");

Classes. ConstantData. jeweler\_orders. Upd();

Classes. ConstantData. jeweler\_orders. tbStatus. Text = "Изготовлен";

}

}

}

}

}

Jeweler\_oldProduct.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Jeweler\_oldProducts : Window

{

string selected\_id = string. Empty;

string filter = string. Empty;

public Jeweler\_oldProducts()

{

InitializeComponent();

LoadDataOldProducts();

}

private void cbFilter\_DropDownClosed(object sender, EventArgs e)

{

if (cbFilter. SelectedIndex == 0)

{

filter = string. Empty;

LoadDataOldProducts();

}

else if (cbFilter. SelectedIndex == 1)

{

filter = "where old\_product\_status = 'На складе'";

LoadDataOldProducts();

}

else if (cbFilter. SelectedIndex == 2)

{

filter = "where old\_product\_status = 'Переплавлено'";

LoadDataOldProducts();

}

}

private void dgOrder\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

try

Продолжение приложения Б

{

DataRowView row = (DataRowView)dgOldProducts. SelectedItems[0];

selected\_id = row. Row[0]. ToString();

}

catch { }

}

private void btnFi\_Click(object sender, RoutedEventArgs e)

{

try

{

MySqlCommand mySqlCommand = new MySqlCommand($"update old\_product set old\_product\_status = 'Переплавлено' where id\_old\_product = '{selected\_id}'", Classes. DB. connection);

mySqlCommand. ExecuteNonQuery();

MessageBox. Show("Украшение переплавлено! ");

LoadDataOldProducts();

}

catch

{

MessageBox. Show("Выберите украшения для переплавки! ");

}

}

private void btnBack\_Click(object sender, RoutedEventArgs e)

{

this. Close();

}

}

}

Admin\_menu.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Admin\_menu : Window

{

public Admin\_menu()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

Admin\_products admin\_Products = new Admin\_products();

admin\_Products. ShowDialog();

}

}

}

Admin\_categories.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Admin\_categories : Window

{

public Admin\_categories()

{

InitializeComponent();

Продолжение приложения Б

Upd();

}

public void Upd()

{

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter($"select id\_category as 'Код категории', category\_title as 'Название категории' from category WHERE category\_title Like '%{tbSearchCategory. Text}%'", Classes. DB. connection);

adapter. Fill(table);

dgCategory. ItemsSource = table. DefaultView;

}

private void EditCategory\_Click(object sender, RoutedEventArgs e)

{

btnSave. Content = "Сохранить";

DataRowView row = (DataRowView)dgCategory. SelectedItems[0];

Classes. ConstantData. id\_category = row[0]. ToString();

tbCategoryTitle. Text = row[1]. ToString();

}

private void DeleteCategory\_Click(object sender, RoutedEventArgs e)

{

DataRowView row = (DataRowView)dgCategory. SelectedItems[0];

MySqlCommand commandabonent = new MySqlCommand($"delete from category where id\_category = {row[0]. ToString()}", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

Upd();

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

if (Classes. ConstantData. id\_category == "" && btnSave. Content. ToString() == "Добавить")

{

if (tbCategoryTitle. Text == "")

{

MessageBox. Show("Данные не введены", "Ошибка");

}

else

{

MySqlCommand commandabonent = new MySqlCommand($"Insert Into category (category\_title) Values ('{tbCategoryTitle. Text}')", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

MessageBox. Show("Данные добавлены! ");

Upd();

}

}

else if (Classes. ConstantData. id\_category ! = "" && btnSave. Content. ToString() == "Сохранить")

{

if (tbCategoryTitle. Text == "")

{

MessageBox. Show("Данные не введены", "Ошибка");

}

else

{

Продолжение приложения Б

MySqlCommand commandabonent = new MySqlCommand($"Update category set category\_title = '{tbCategoryTitle. Text}' where id\_category = '{Classes. ConstantData. id\_category}'", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

MessageBox. Show("Данные изменены! ");

Upd();

}

}

Classes. ConstantData. id\_category = "";

tbCategoryTitle. Text = "";

btnSave. Content = "Добавить";

}

}

}

Admin\_employee.xaml.cs

using jewelryworkshop. UserControls;

using MySql. Data. MySqlClient;

using System. Windows. Media;

using System. Windows. Media. Imaging;

using System. Windows. Media. Media3D;

using System. Windows. Shapes;

namespace jewelryworkshop. Windows

{

public partial class Admin\_employee : Window

{

bool Manager = false;

public Admin\_employee()

{

InitializeComponent();

DateTime dateTimeNow = DateTime. Now;

DateTime dateTimeMin = dateTimeNow. AddYears(-52);

dpDate1. DisplayDateStart = dateTimeMin;

dpDate1. DisplayDateEnd = DateTime. Now. AddYears(-18);

dpDate1. Text = dpDate1. DisplayDateEnd. ToString();

Upd();

}

private void Window\_Closed(object sender, EventArgs e)

{

Classes. ConstantData. id\_employee = "";

}

private void btnClearSupplier\_Click(object sender, RoutedEventArgs e)

{

Classes. ConstantData. id\_employee = "";

tbEmployeeName. Text = "";

tbEmployeeNumber. Text = "";

tbEmployeePassport. Text = "";

tbEmployeeLogin. Text = "";

tbEmployeePassword. Text = "";

tbEmployeeDateA. Text = "";

tbEmployeeDateB. Text = "";

cbPost. Text = "";

tbStatus. Text = "";

Продолжение приложения Б

dpDate1. Text = "";

btnSave. Content = "Добавить";

}

private void DeleteEmployee\_Click(object sender, RoutedEventArgs e)

{

DataRowView row = (DataRowView)dgEmployee. SelectedItems[0];

if (row[0]. ToString() == Classes. ConstantData. user\_id)

{

MessageBox. Show("Вы не можете уволить сами себя! ");

}

else

{

MySqlCommand commandcity = new MySqlCommand("Update employee set id\_employee\_status = '2', dismissal\_date = '" + DateTime. Now. ToString("yyyy/MM/dd") + "' where id\_employee = " + row[0]. ToString(), Classes. DB. connection);

commandcity. ExecuteNonQuery();

MessageBox. Show("Сотрудник уволен! ");

Upd();

}

}

private void btnReset\_Click(object sender, RoutedEventArgs e)

{

if (dgEmployee. SelectedItems. Count == 0)

{

MessageBox. Show("Выберите сотрудника для восстановления");

}

else

{

DataRowView row = (DataRowView)dgEmployee. SelectedItems[0];

if (row["Статус"]. ToString() == "Устроен")

{

MessageBox. Show("Сотрудник не был уволен! ");

}

else

{

MySqlCommand commandcity = new MySqlCommand("Update employee set id\_employee\_status = '1', dismissal\_date = NULL, employment\_date = '" + DateTime. Now. ToString("yyyy/MM/dd") + "' where id\_employee = " + row[0]. ToString(), Classes. DB. connection);

commandcity. ExecuteNonQuery();

MessageBox. Show("Сотрудник восстановлен! ");

Upd();

}

}

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

if (Classes. ConstantData. id\_employee == "" && btnSave. Content. ToString() == "Добавить")

{

if (tbEmployeeName. Text == "" || tbEmployeePassport. Text == "" || tbEmployeeNumber. Text == "" || tbEmployeeLogin. Text == "" || tbEmployeePassword. Text == "" || cbPost. Text == "" || dpDate1. Text == "")

{

Продолжение приложения Б

MessageBox. Show("Данные не введены", "Ошибка");

}

else

{

DateTime dateTime = DateTime. Parse(dpDate1. Text);

string dt = dateTime. ToString("yyyy/MM/dd");

MySqlCommand commandabonent = new MySqlCommand($"Insert Into employee (full\_name, passport, phone\_number, birth\_date, login, password, id\_post, employment\_date, id\_employee\_status) Values ('{tbEmployeeName. Text}', '{tbEmployeePassport. Text}', '{tbEmployeeNumber. Text}', '{dt}', '{tbEmployeeLogin. Text}', '{tbEmployeePassword. Text}', '{(cbPost. SelectedIndex + 2)}', '{DateTime. Now. ToString("yyyy/MM/dd")}', '1')", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

MessageBox. Show("Данные добавлены! ");

Upd();

}

}

else if (Classes. ConstantData. id\_employee ! = "" && btnSave. Content. ToString() == "Сохранить")

{

if (tbEmployeeName. Text == "" || tbEmployeePassport. Text == "" || tbEmployeeNumber. Text == "" || tbEmployeeLogin. Text == "" || tbEmployeePassword. Text == "" || cbPost. Text == "" || dpDate1. Text == "")

{

MessageBox. Show("Данные не введены", "Ошибка");

}

else

{

MySqlCommand commandabonent;

if (Manager == false)

{

commandabonent = new MySqlCommand($"Update product set full\_name = '{tbEmployeeName. Text}', passport = '{tbEmployeePassport. Text}', phone\_number = '{tbEmployeeNumber. Text}', birth\_date = '{dpDate1. Text}', login = '{tbEmployeeLogin. Text}', password = '{tbEmployeePassword. Text}', id\_post = '{(cbPost. SelectedIndex + 2)}, where id\_employee = '{Classes. ConstantData. id\_employee}'", Classes. DB. connection);

}

else commandabonent = new MySqlCommand($"Update product set full\_name = '{tbEmployeeName. Text}', passport = '{tbEmployeePassport. Text}', phone\_number = '{tbEmployeeNumber. Text}', birth\_date = '{dpDate1. Text}', login = '{tbEmployeeLogin. Text}', password = '{tbEmployeePassword. Text}' where id\_employee = '{Classes. ConstantData. id\_employee}'", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

MessageBox. Show("Данные изменены! ");

Upd();

Manager = false;

}

}

btnSave. Content = "Добавить";

}

private void btnEditEmployee\_Click(object sender, RoutedEventArgs e)

{

btnSave. Content = "Сохранить";

DataRowView row = (DataRowView)dgEmployee. SelectedItems[0];

Classes. ConstantData. id\_employee = row[0]. ToString();

tbEmployeeName. Text = row[1]. ToString();

Продолжение приложения Б

tbEmployeePassport. Text = row[2]. ToString();

tbEmployeeNumber. Text = row[3]. ToString();

dpDate1. Text = row[4]. ToString();

tbEmployeeLogin. Text = row[5]. ToString();

tbEmployeePassword. Text = row[6]. ToString();

cbPost. Text = row[7]. ToString();

tbEmployeeDateA. Text = row[8]. ToString(). Replace(" 0:00:00", "");

tbEmployeeDateB. Text = row[9]. ToString(). Replace(" 0:00:00", "");

tbStatus. Text = row[10]. ToString();

if (row[7]. ToString() == "Менеджер")

{

Manager = true;

cbPost. IsEnabled = false;

}

else cbPost. IsEnabled = true;

}

}

}

Admin\_materials.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Admin\_materials : Window

{

public Admin\_materials()

{

InitializeComponent();

Upd();

}

public void Upd()

{

cbSupplier. Items. Clear();

MySqlDataAdapter adapter = new MySqlDataAdapter($"select supplier\_title from supplier", Classes. DB. connection);

DataTable table = new DataTable();

adapter. Fill(table);

for (int i = 0; i < table. Rows. Count; i++)

{

cbSupplier. Items. Add(table. Rows[i][0]. ToString());

}

table = new DataTable();

adapter = new MySqlDataAdapter($"SELECT id\_material as 'Код материала', material\_title as 'Название материала', material\_weight as 'Общий вес (гр. )', supplier. supplier\_title as 'Поставщик' FROM material, supplier where material. id\_supplier = supplier. id\_supplier and material\_title Like '%{tbSearchMaterial. Text}%'", Classes. DB. connection);

adapter. Fill(table);

dgMaterial. ItemsSource = table. DefaultView;

}

private void EditMaterial\_Click(object sender, RoutedEventArgs e)

{

btnSave. Content = "Сохранить";

DataRowView row = (DataRowView)dgMaterial. SelectedItems[0];

Продолжение приложения Б

Classes. ConstantData. id\_supplier = row[0]. ToString();

tbMaterialTitle. Text = row[1]. ToString();

tbMaterialWeight. Text = row[2]. ToString();

cbSupplier. Text = row[3]. ToString();

}

private void DeleteMaterial\_Click(object sender, RoutedEventArgs e)

{

DataRowView row = (DataRowView)dgMaterial. SelectedItems[0];

MySqlCommand commandabonent = new MySqlCommand($"delete from material where id\_material = {row[0]. ToString()}", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

Upd();

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

string supplierId = "";

try

{

MySqlDataAdapter categoryAdapter = new MySqlDataAdapter($"select id\_supplier from supplier where supplier\_title = '{cbSupplier. SelectedItem. ToString()}'", Classes. DB. connection);

DataTable categoryTable = new DataTable();

categoryAdapter. Fill(categoryTable);

supplierId = categoryTable. Rows[0][0]. ToString();

}

catch { }

if (Classes. ConstantData. id\_material == "" && btnSave. Content. ToString() == "Добавить")

{

if (tbMaterialTitle. Text == "" || tbMaterialWeight. Text == "" || cbSupplier. Text == "")

{

MessageBox. Show("Данные не введены", "Ошибка");

}

else

{

MySqlCommand commandabonent = new MySqlCommand($"Insert Into material (material\_title, material\_weight, id\_supplier) Values ('{tbMaterialTitle. Text}', '{tbMaterialWeight. Text}', '{supplierId}')", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

MessageBox. Show("Данные добавлены! ");

Upd();

}

}

else if (Classes. ConstantData. id\_material ! = "" && btnSave. Content. ToString() == "Сохранить")

{

if (tbMaterialTitle. Text == "" || tbMaterialWeight. Text == "" || cbSupplier. Text == "")

{

MessageBox. Show("Данные не введены", "Ошибка");

}

else

{

MySqlCommand commandabonent = new MySqlCommand($"Update supplier set material\_title = '{tbMaterialTitle. Text}', material\_weight = '{tbMaterialWeight. Text}', id\_supplier = '{supplierId}' where id\_material = '{Classes. ConstantData. id\_material}'", Classes. DB. connection);

Продолжение приложения Б

commandabonent. ExecuteNonQuery();

MessageBox. Show("Данные изменены! ");

Upd();

}

}

Classes. ConstantData. id\_material = "";

btnSave. Content = "Добавить";

tbMaterialTitle. Text = "";

tbMaterialWeight. Text = "";

cbSupplier. Text = "";

}

}

}

Admin\_products.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Admin\_products : Window

{

string Filter = "";

string Sort = "";

public Admin\_products()

{

InitializeComponent();

Classes. ConstantData. admin\_Products = this;

Upd();

imagePhoto. Source = new BitmapImage(new Uri("C: \\Users\\furuh\\source\\repos\\jewelryworkshop\\jewelryworkshop\\Images\\picture. png", UriKind. Absolute));

}

public void Upd()

{

cbCategory. Items. Clear();

cbMaterial. Items. Clear();

MySqlDataAdapter adapter = new MySqlDataAdapter($"select category\_title from category", Classes. DB. connection);

DataTable table = new DataTable();

adapter. Fill(table);

for (int i = 0; i < table. Rows. Count; i++)

{

cbCategory. Items. Add(table. Rows[i][0]. ToString());

}

adapter = new MySqlDataAdapter($"select material\_title from material", Classes. DB. connection);

table = new DataTable();

adapter. Fill(table);

for (int i = 0; i < table. Rows. Count; i++)

{

cbMaterial. Items. Add(table. Rows[i][0]. ToString());

}

wPanel. Children. Clear();

Продолжение приложения Б

adapter = new MySqlDataAdapter($"select id\_product, product\_title, product\_price, quantity, product\_weight, photo, category. category\_title, material. material\_title, id\_order from product, category, material where product. id\_category = category. id\_category and product. id\_material = material. id\_material and product\_title LIKE '%{tbSearch. Text}%' {Filter} {Sort}", Classes. DB. connection);

table = new DataTable();

adapter. Fill(table);

for (int i = 0; i < table. Rows. Count; i++)

{

wPanel. Children. Add(new Admin\_product(table. Rows[i][0]. ToString()));

}

}

private void btnSelectPhoto\_Click(object sender, RoutedEventArgs e)

{

try

{

var dialog = new Microsoft. Win32. OpenFileDialog();

dialog. Filter = "PNG Files (\*. png)|\*. png|JPG Files (\*. jpg)|\*. jpg|All Files (\*. \*)|\*. \*";

bool? result = dialog. ShowDialog();

if (result == true)

{

Classes. ConstantData. photopath = dialog. FileName;

string filename = dialog. FileName;

imagePhoto. Source = new BitmapImage(new Uri(filename, UriKind. Absolute));

tbPhoto. Text = filename;

}

}

catch (Exception)

{

}

}

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

string categoryId = "", materialId = "";

try

{

MySqlDataAdapter categoryAdapter = new MySqlDataAdapter($"select id\_category from category where category\_title = '{cbCategory. SelectedItem. ToString()}'", Classes. DB. connection);

DataTable categoryTable = new DataTable();

categoryAdapter. Fill(categoryTable);

categoryId = categoryTable. Rows[0][0]. ToString();

MySqlDataAdapter materialAdapter = new MySqlDataAdapter($"select id\_material from material where material\_title = '{cbMaterial. SelectedItem. ToString()}'", Classes. DB. connection);

DataTable materialTable = new DataTable();

materialAdapter. Fill(materialTable);

materialId = materialTable. Rows[0][0]. ToString();

}

catch { }

DataTable prodTable = null;

try

{

Продолжение приложения Б

MySqlDataAdapter prodAdapter = new MySqlDataAdapter($"SELECT \* FROM jewelry. product where product\_title = '{tbTitle. Text}' and product\_price = '{tbCost. Text}' and product\_weight = {tbWeight. Text} and id\_category = '{categoryId}' and id\_material = '{materialId}'", Classes. DB. connection);

prodTable = new DataTable();

prodAdapter. Fill(prodTable);

}

catch { }

if (Classes. ConstantData. id\_product == "" && btnSave. Content. ToString() == "Добавить")

{

if (tbTitle. Text == "" || tbCost. Text == "" || tbQuantity. Text == "" || tbWeight. Text == "" || cbMaterial. Text == "" || cbCategory. Text == "")

{

MessageBox. Show("Данные не введены", "Ошибка");

}

else

{

if (prodTable. Rows. Count ! = 0)

{

MySqlCommand commandabonent = new MySqlCommand($"Update product set quantity = '{int. Parse(prodTable. Rows[0][3]. ToString()) + int. Parse(tbQuantity. Text)}' where id\_product = '{prodTable. Rows[0][0]. ToString()}'", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

MessageBox. Show("Данные изменены! ");

Upd();

}

else

{

if (tbPhoto. Text ! = "")

{

MySqlCommand commandabonent = new MySqlCommand($"Insert Into product (product\_title, product\_price, quantity, product\_weight, photo, id\_category, id\_material) Values ('{tbTitle. Text}', '{tbCost. Text}', '{tbQuantity. Text}', '{tbWeight. Text}', '{tbPhoto. Text. Replace("\\", "\\\\")}', '{categoryId}', '{materialId}')", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

MessageBox. Show("Данные добавлены! ");

Upd();

}

else

{

MySqlCommand commandabonent = new MySqlCommand($"Insert Into product (product\_title, product\_price, quantity, product\_weight, id\_category, id\_material) Values ('{tbTitle. Text}', '{tbCost. Text}', '{tbQuantity. Text}', '{tbWeight. Text}', '{categoryId}', '{materialId}')", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

MessageBox. Show("Данные добавлены! ");

Upd();

}

}

}

}

else if (Classes. ConstantData. id\_product ! = "" && btnSave. Content. ToString() == "Сохранить")

{

Продолжение приложения Б

if (tbTitle. Text == "" || tbCost. Text == "" || tbQuantity. Text == "" || tbWeight. Text == "" || cbMaterial. Text == "" || cbCategory. Text == "")

{

MessageBox. Show("Данные не введены", "Ошибка");

}

else

{

if (tbPhoto. Text ! = "")

{

MySqlCommand commandabonent = new MySqlCommand($"Update product set product\_title = '{tbTitle. Text}', product\_price = '{tbCost. Text}', quantity = '{tbQuantity. Text}', product\_weight = '{tbWeight. Text}', photo = '{tbPhoto. Text. Replace("\\", "\\\\")}', id\_category = '{categoryId}', id\_material = '{materialId}' where id\_product = '{Classes. ConstantData. id\_product}'", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

MessageBox. Show("Данные изменены! ");

Upd();

}

else

{

MySqlCommand commandabonent = new MySqlCommand($"Update product set product\_title = '{tbTitle. Text}', product\_price = '{tbCost. Text}', quantity = '{tbQuantity. Text}', product\_weight = '{tbWeight. Text}', id\_category = '{categoryId}', id\_material = '{materialId}', photo = null where id\_product = '{Classes. ConstantData. id\_product}'", Classes. DB. connection);

commandabonent. ExecuteNonQuery();

MessageBox. Show("Данные изменены! ");

Upd();

}

}

}

Classes. ConstantData. id\_product = "";

tbTitle. Text = "";

tbCost. Text = "";

tbQuantity. Text = "";

tbWeight. Text = "";

tbPhoto. Text = "";

cbCategory. Text = "";

cbMaterial. Text = "";

imagePhoto. Source = new BitmapImage(new Uri("C: \\Users\\furuh\\source\\repos\\jewelryworkshop\\jewelryworkshop\\Images\\picture. png", UriKind. Absolute));

btnSave. Content = "Добавить";

}

private void btnClearPhoto\_Click(object sender, RoutedEventArgs e)

{

tbPhoto. Text = "";

imagePhoto. Source = new BitmapImage(new Uri("C: \\Users\\furuh\\source\\repos\\jewelryworkshop\\jewelryworkshop\\Images\\picture. png", UriKind. Absolute));

}

private void btnCancel\_Click(object sender, RoutedEventArgs e)

{

Classes. ConstantData. id\_product = "";

btnSave. Content = "Добавить";

Продолжение приложения Б

}

}

}

}

Admin\_product.xaml.cs

using jewelryworkshop. Classes;

using jewelryworkshop. Windows;

using System;

using System. Collections. Generic;

using System. Data;

using static System. Net. Mime. MediaTypeNames;

namespace jewelryworkshop. UserControls

{

public partial class Admin\_product : UserControl

{

string ID;

string Image\_path;

DataTable table = new DataTable();

public Admin\_product(string ID)

{

InitializeComponent();

MySqlDataAdapter adapter = new MySqlDataAdapter($"select id\_product, product\_title, product\_price, quantity, product\_weight, photo, category. category\_title, material. material\_title, id\_order from product, category, material where product. id\_category = category. id\_category and product. id\_material = material. id\_material and id\_product = {ID}", Classes. DB. connection);

adapter. Fill(table);

idProduct. Content = $"{table. Rows[0][0]}";

photoPath. Content = $"{table. Rows[0][5]}";

lbTitle. Content = "" + table. Rows[0][1];

lbCost. Content = "" + table. Rows[0][2] + " ₽";

lbQuantity. Content = "В наличии: " + table. Rows[0][3] + " шт. ";

lbWeight. Content = $"{table. Rows[0][4]} гр. ";

lbCategory. Content = $"{table. Rows[0][6]}";

lbMaterial. Content = $"{table. Rows[0][7]}";

if (table. Rows[0][8]. ToString() ! = "")

{

lbOrder. Content = $"No заказа: {table. Rows[0][8]}";

}

Image\_path = table. Rows[0][5]. ToString();

try

{ img. ImageSource = new BitmapImage(new Uri($@"{table. Rows[0][5]}", UriKind. Absolute)); }

catch

{ img. ImageSource = new BitmapImage(new Uri(@"C: \Users\furuh\source\repos\jewelryworkshop\jewelryworkshop\Images\picture. png", UriKind. Relative)); }

this. ID = ID;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

Classes. ConstantData. id\_product = idProduct. Content. ToString();

Classes. ConstantData. admin\_Products. btnSave. Content = "Сохранить";

Продолжение приложения Б

Classes. ConstantData. admin\_Products. imagePhoto. Source = new BitmapImage(new Uri("C: \\Users\\furuh\\source\\repos\\jewelryworkshop\\jewelryworkshop\\Images\\picture. png", UriKind. Absolute));

Classes. ConstantData. admin\_Products. tbTitle. Text = lbTitle. Content. ToString();

Classes. ConstantData. admin\_Products. tbCost. Text = lbCost. Content. ToString(). Substring(0, lbCost. Content. ToString(). Length - 2);

string quantity = lbQuantity. Content. ToString(). Replace("В наличии: ", "");

quantity = quantity. Replace(" шт. ", "");

Classes. ConstantData. admin\_Products. tbQuantity. Text = quantity;

Classes. ConstantData. admin\_Products. tbWeight. Text = lbWeight. Content. ToString(). Replace(" гр. ", "");

Classes. ConstantData. admin\_Products. cbMaterial. Text = lbMaterial. Content. ToString();

Classes. ConstantData. admin\_Products. cbCategory. Text = lbCategory. Content. ToString();

Classes. ConstantData. admin\_Products. tbPhoto. Text = photoPath. Content. ToString();

if (photoPath. Content. ToString() ! = "")

{

Classes. ConstantData. admin\_Products. imagePhoto. Source = new BitmapImage(new Uri(photoPath. Content. ToString(), UriKind. Absolute));

}

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

if (lbOrder. Content. ToString() ! = " ")

{

MessageBoxResult dialogResult = MessageBox. Show("Это украшение изготовлено на заказ, вы уверены, что хотите удалить его? ", "Удаление", MessageBoxButton. YesNo);

switch (dialogResult)

{

case MessageBoxResult. Yes:

MySqlCommand commandcity = new MySqlCommand($"DELETE FROM product WHERE id\_product = {idProduct. Content}", Classes. DB. connection);

commandcity. ExecuteNonQuery();

Classes. ConstantData. admin\_Products. Upd();

break;

case MessageBoxResult. No:

break;

}

}

else

{

MySqlCommand commandcity = new MySqlCommand($"DELETE FROM product WHERE id\_product = {idProduct. Content}", Classes. DB. connection);

commandcity. ExecuteNonQuery();

Classes. ConstantData. admin\_Products. Upd();

}

}

}

}

Admin\_supplier.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Admin\_suppliers : Window

{

Продолжение приложения Б

public Admin\_suppliers()

{

InitializeComponent();

Upd();

}

public void Upd()

{

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter($"select id\_supplier as 'Код поставщика', supplier\_title as 'Название компании поставщика', supplier\_number as 'Номер телефона', address as 'Адрес' from supplier WHERE supplier\_title Like '%{tbSearchSupplier. Text}%'", Classes. DB. connection);

adapter. Fill(table);

dgSupplier. ItemsSource = table. DefaultView;

}

}

}

Admin\_report.xaml.cs

using Microsoft. Office. Interop. Excel;

using MySql. Data. MySqlClient;

using Excel = Microsoft. Office. Interop. Excel;

using DataTable = System. Data. DataTable;

using Window = System. Windows. Window;

using System. Xml. Linq;

namespace jewelryworkshop. Windows

{

public partial class Admin\_report : Window

{

string data = "";

string employee = "";

DataTable table;

public Admin\_report()

{

InitializeComponent();

try

{

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter($"select MIN(receipt\_date) from receipt\_tp", Classes. DB. connection);

adapter. Fill(table);

dpDate1. DisplayDateStart = DateTime. Parse(table. Rows[0][0]. ToString());

dpDate1. SelectedDate = DateTime. Parse(table. Rows[0][0]. ToString());

dpDate2. DisplayDateStart = DateTime. Parse(table. Rows[0][0]. ToString());

table = new DataTable();

adapter = new MySqlDataAdapter($"select MAX(receipt\_date) from receipt\_tp", Classes. DB. connection);

adapter. Fill(table);

dpDate2. DisplayDateEnd = DateTime. Parse(table. Rows[0][0]. ToString());

dpDate2. SelectedDate = DateTime. Parse(table. Rows[0][0]. ToString());

dpDate1. DisplayDateEnd = DateTime. Parse(table. Rows[0][0]. ToString());

}

catch { }

Upd();

Продолжение приложения Б

cbEmployee. SelectedIndex = 0;

}

public void Upd()

{

table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter($"select id\_receipt as 'Номер чека', product\_title as 'Название украшения', receipt\_tp. quantity as 'Количество', product\_price as 'Цена', sum\_cost as 'Общая стоимость', receipt\_date as 'Дата продажи', employee. full\_name as 'ФИО кассира', id\_receipt\_tp from receipt\_tp, product, employee where receipt\_tp. id\_product = product. id\_product and receipt\_tp. id\_employee = employee. id\_employee {data} {employee}", Classes. DB. connection);

adapter. Fill(table);

dgReport. ItemsSource = table. DefaultView;

DataTable dt = new DataTable();

adapter = new MySqlDataAdapter($"select full\_name from employee where id\_post = '3'", Classes. DB. connection);

adapter. Fill(dt);

cbEmployee. Items. Clear();

cbEmployee. Items. Add("Все сотрудники");

for (int i = 0; i < dt. Rows. Count; i++)

{

cbEmployee. Items. Add(dt. Rows[i][0]. ToString());

}

}

private void dpDate1\_SelectedDateChanged(object sender, SelectionChangedEventArgs e)

{

dpDate2. DisplayDateStart = dpDate1. SelectedDate;

if (dpDate1. Text ! = "" && dpDate2. Text ! = "")

{

data = $" And receipt\_date >= STR\_TO\_DATE('{dpDate1. Text}', '%d. %m. %Y') And receipt\_date <= STR\_TO\_DATE('{dpDate2. Text}', '%d. %m. %Y')";

Upd();

}

}

private void dpDate2\_SelectedDateChanged(object sender, SelectionChangedEventArgs e)

{

dpDate1. DisplayDateEnd = dpDate2. SelectedDate;

if (dpDate1. Text ! = "" && dpDate2. Text ! = "")

{

data = $" And receipt\_date >= STR\_TO\_DATE('{dpDate1. Text}', '%d. %m. %Y') And receipt\_date <= STR\_TO\_DATE('{dpDate2. Text}', '%d. %m. %Y')";

Upd();

}

}

private void btnReport\_Click(object sender, RoutedEventArgs e)

{

if (dgReport. ItemsSource ! = null)

{

if (table. Rows. Count ! = 0)

{

Excel. Application excel = new Excel. Application();

Excel. Workbook workbook = excel. Workbooks. Open(@"C: \Users\furuh\Documents\Distant\Diplom\Report. xlsx");

Продолжение приложения Б

Excel. Worksheet sheet1 = (Excel. Worksheet)workbook. Sheets[1];

MySqlCommand mySqlCommand;

DataTable dataTable = new DataTable();

dataTable = ((DataView)dgReport. ItemsSource). ToTable();

dataTable. Columns. Remove("id\_receipt\_tp");

DataTable dt = new DataTable();

dt = ((DataView)dgReport. ItemsSource). ToTable();

int reportNumber = 0;

DataTable reportNum = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter($"SELECT max(id\_report) from report\_tp", Classes. DB. connection);

adapter. Fill(reportNum);

if (reportNum. Rows[0][0]. ToString() == string. Empty)

{

reportNumber = 1;

}

else

{

reportNumber = int. Parse(reportNum. Rows[0][0]. ToString()) + 1;

}

Range range = (Range)sheet1. Cells[10, 1];

range. Value2 = $"Отчет по проданным изделиям No {reportNumber}";

range = (Range)sheet1. Cells[11, 4];

range. Value2 = $"Период с {dpDate1. Text} по {dpDate2. Text}";

range = (Range)sheet1. Cells[12, 6];

range. Font. Name = "Roboto Light";

range. Value2 = DateTime. Now. ToShortDateString();

string curDate = DateTime. Now. ToShortDateString();

int sumcost = 0;

DateTime sentDate;

for (int i = 0; i < dataTable. Rows. Count; i++)

{

for (int j = 0; j < dataTable. Columns. Count; j++)

{

if (j == 4)

{

sumcost += int. Parse(dataTable. Rows[i][j]. ToString());

}

range = (Range)sheet1. Cells[i + 14, j + 1];

range. Font. Name = "Roboto Light";

range. Borders. LineStyle = XlLineStyle. xlContinuous;

range. HorizontalAlignment = XlHAlign. xlHAlignCenter;

range. VerticalAlignment = XlVAlign. xlVAlignCenter;

if (j == 5)

{

sentDate = DateTime. Parse(dataTable. Rows[i][j]. ToString());

range. Value2 = sentDate. ToString("dd. MM. yyyy");

}

else

{

range. Value2 = dataTable. Rows[i][j]. ToString();

Продолжение приложения Б

}

}

}

range = (Range)sheet1. Cells[dataTable. Rows. Count + 15, 5];

range. Font. Name = "Roboto Light";

range. HorizontalAlignment = XlHAlign. xlHAlignCenter;

range. Value2 = $"Итого: {sumcost} руб. ";

range = (Range)sheet1. Cells[dataTable. Rows. Count + 16, 1];

range. Font. Name = "Roboto Light";

range. HorizontalAlignment = XlHAlign. xlHAlignCenter;

range. Value2 = $"Подпись менеджера: ";

Range myRange6 = sheet1. UsedRange;

myRange6. Columns. AutoFit();

for (int i = 0; i < dataTable. Rows. Count; i++)

{

mySqlCommand = new MySqlCommand($"insert into report\_tp (id\_report, id\_receipt\_tp, date\_report) values ('{reportNumber}', '{dt. Rows[i][7]. ToString()}', '{DateTime. Now. ToString("yyyy-MM-dd")}')", Classes. DB. connection);

mySqlCommand. ExecuteNonQuery();

}

excel. Visible = true;

}

else MessageBox. Show("Выберите период, в котором будут записи! ");

}

else MessageBox. Show("Выберите период, в котором будут записи! ");

}

}

}

Admin\_reportCertificate.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Admin\_reportCertificate : Window

{

string data = "";

DataTable table;

public Admin\_reportCertificate()

{

InitializeComponent();

Upd();

}

}

}

Admin\_reportOldProducts.xaml.cs

namespace jewelryworkshop. Windows

{

public partial class Admin\_reportOldProducts : Window

{

string data = "";

string employee = "";

DataTable table;

public Admin\_reportOldProducts()

Продолжение приложения Б

{

InitializeComponent();

Upd();

cbEmployee. SelectedIndex = 0;

}

void Upd()

{

table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter($"select id\_clients\_old\_product, old\_product\_title, old\_product\_weight, cost, `client`. full\_name as 'клиентФ', `client`. phone\_number as 'клиентН', sale\_date, employee. full\_name as 'сотрФ' from clients\_old\_product, old\_product, `client`, employee where clients\_old\_product. id\_old\_product = old\_product. id\_old\_product and clients\_old\_product. id\_client\_product = `client`. id\_client and clients\_old\_product. id\_employee\_product = employee. id\_employee {data} {employee}", Classes. DB. connection);

adapter. Fill(table);

dgReport. ItemsSource = table. DefaultView;

DataTable dt = new DataTable();

adapter = new MySqlDataAdapter($"select full\_name from employee where id\_post = '3'", Classes. DB. connection);

adapter. Fill(dt);

cbEmployee. Items. Clear();

cbEmployee. Items. Add("Все сотрудники");

for (int i = 0; i < dt. Rows. Count; i++)

{

cbEmployee. Items. Add(dt. Rows[i][0]. ToString());

}

}

}

ConstantData.cs

using jewelryworkshop. UserControls;

using jewelryworkshop. Windows;

using System;

using System. Collections. Generic;

using System. Linq;

using System. Text;

using System. Threading. Tasks;

namespace jewelryworkshop. Classes

{

internal class ConstantData

{

public static string photopath = "";

public static string id\_product = "";

public static string id\_supplier = "";

public static string id\_category = "";

public static string id\_employee = "";

public static string id\_material = "";

public static string id\_order = "";

public static string id\_client = "";

public static string id\_busket\_product = "";

public static string user\_fio = "";

public static string user\_id = "";

public static Authorization authorization;

Продолжение приложения Б

public static Admin\_products admin\_Products;

public static Jeweler\_orders jeweler\_orders;

public static Jeweler\_addProduct jeweler\_AddProduct;

public static Cashier\_products cashier\_products;

public static Certificate certificate;

public static Cashier\_busket cashier\_Busket;

public static Cashier\_clientsOldProduct cashier\_ClientsOldProduct;

public static bool addClient\_fromBusket = false;

public static bool addClient\_fromCertificate = false;

public static bool addClient\_fromOldProduct = false;

public static List<string> id\_products = new List<string>();

public static List<string> id\_orders = new List<string>();

public static List<int> quantity\_products = new List<int>();

}

}

DB.cs

using MySql. Data. MySqlClient;

using System;

using System. Collections. Generic;

using System. Data. SqlClient;

using System. Linq;

using System. Text;

using System. Threading. Tasks;

namespace jewelryworkshop. Classes

{

internal class DB

{

public static MySqlConnection connection = new MySqlConnection(@"Server = localhost; Database= jewelry; port= 3306; User Id= root; password=251436");

public void openConnection()

{

if (connection. State == System. Data. ConnectionState. Open)

{

connection. Open();

}

}

public void closeConnection()

{

if (connection. State == System. Data. ConnectionState. Closed)

{

connection. Close();

}

}

public MySqlConnection GetMySqlConnection()

{

return connection;

}

}

}