Санкт-Петербургское государственное бюджетное профессиональное образовательное учреждение

«Колледж информационных технологий»

ОТЧЕТ

ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

по ПМ.05 «Проектирование и разработка информационных систем»

ПМ.06 «Сопровождение информационных систем»

Специальность 09.02.07 «Информационные системы и программирование» (Специалист по информационным системам)

Выполнил

студент гр.: 485 \_\_\_\_\_\_\_\_\_\_\_Ястреб Д.К.

Согласовано

ООО «АСП Консалт» \_\_\_\_\_\_\_\_\_\_\_Алексеев К.В.

Руководитель производственной практики \_\_\_\_\_\_\_\_\_\_\_Бакай О.Н.

Содержание

[**ВВЕДЕНИЕ** 3](#_Toc100610816)

[**ОСНОВНАЯ ЧАСТЬ** 4](#_Toc100610817)

[1. Сбор данных для разработки проектной документации 4](#_Toc100610818)

[2. Разработка проектной документации, подсистемы безопасности 4](#_Toc100610819)

[3. Разработка модулей информационной системы 4](#_Toc100610820)

[4. Тестирование информационной системы 5](#_Toc100610821)

[5. Разработка технической документации 8](#_Toc100610822)

[6. Оценка информационной системы 8](#_Toc100610823)

[7. Разработка технического задания на сопровождение информационной системы 8](#_Toc100610824)

[8. Выполнение отладки программного кода 9](#_Toc100610825)

[9. Разработка обучающей документации 9](#_Toc100610826)

[10. Оценка качества и надежности функционирования информационной системы 9](#_Toc100610827)

[11. Осуществление сопровождения информационной системы 9](#_Toc100610828)

[**ЗАКЛЮЧЕНИЕ** 11](#_Toc100610829)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 12](#_Toc100610830)

[Приложение 1 13](#_Toc100610831)

# **ВВЕДЕНИЕ**

Производственная практика является составной частью основной

профессиональной образовательной программы по специальности.

Производственная практика направлена на углубление первоначального

профессионального опыта, развитие общих и профессиональных компетенций, проверку готовности к самостоятельной трудовой деятельности выпускника, а также на подготовку к выполнению выпускной квалификационной работы в организациях различных организационно-правовых форм.

Прохождение производственной практики повышает качество профессиональной подготовки выпускников, позволяет закрепить приобретаемые теоретические знания, способствует социально-психологической адаптации на местах будущей работы.

ООО "АСП-Консалт" является официальным партнером фирмы 1С и оказывает услуги по автоматизации бухгалтерского учета бюджетных организаций. Главной целью компании является повышение качества жизни предприятий в России за счет автоматизации наиболее важных участков работы в организациях.

Направления работы ООО «АСП Консалт»:

* Услуги 1С по автоматизации бухгалтерского учета бюджетных организаций.
* Обновление программного обеспечения и разработка дополнительных модулей и функций системы.
* Продажа системы Битрикс24 и профессиональное внедрение в процесс работы организации.

# **ОСНОВНАЯ ЧАСТЬ**

## Сбор данных для разработки проектной документации

Сбор данных происходил с помощью интернет ресурсов и включал в себя: анализ и сравнение аналогов, выявление сильных и слабых сторон. По итогу сложилось представление будущей ИС, краткое описание представлено ниже.

Разрабатываемая система должна охватывать два важных бизнес-процесса: хранение и обработка информации, а также обслуживание клиентов с генерацией печатных форм средствами MS Office.

Благодаря автоматизации данных процессов увеличится скорость обслуживания, и соответственно увеличится количество обслуживаемых клиентов. Также уменьшится нагрузка на персонал, что благоприятно отразится на качестве работы.

Система предназначена для различных типов пользователей (требуется авторизация).

## Разработка проектной документации, подсистемы безопасности

После сбора данных совместно с руководителем практики на предприятии была разработана проектная документация, а именно техническое задание и диаграммы IDEF0.

## Разработка модулей информационной системы

Были разработаны модули авторизации, управления пользователями, и модуль оператора. (рис.1-3)

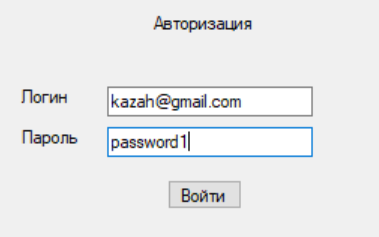


Рисунок 1 - Модуль авторизации

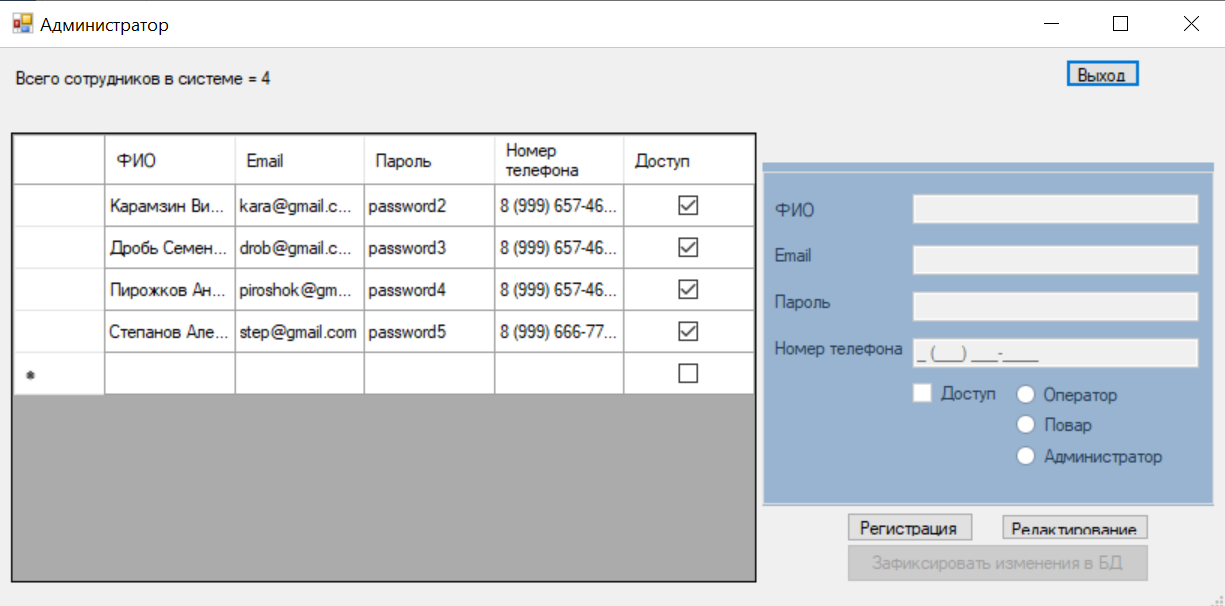


Рисунок 2 - Модуль управления пользователями

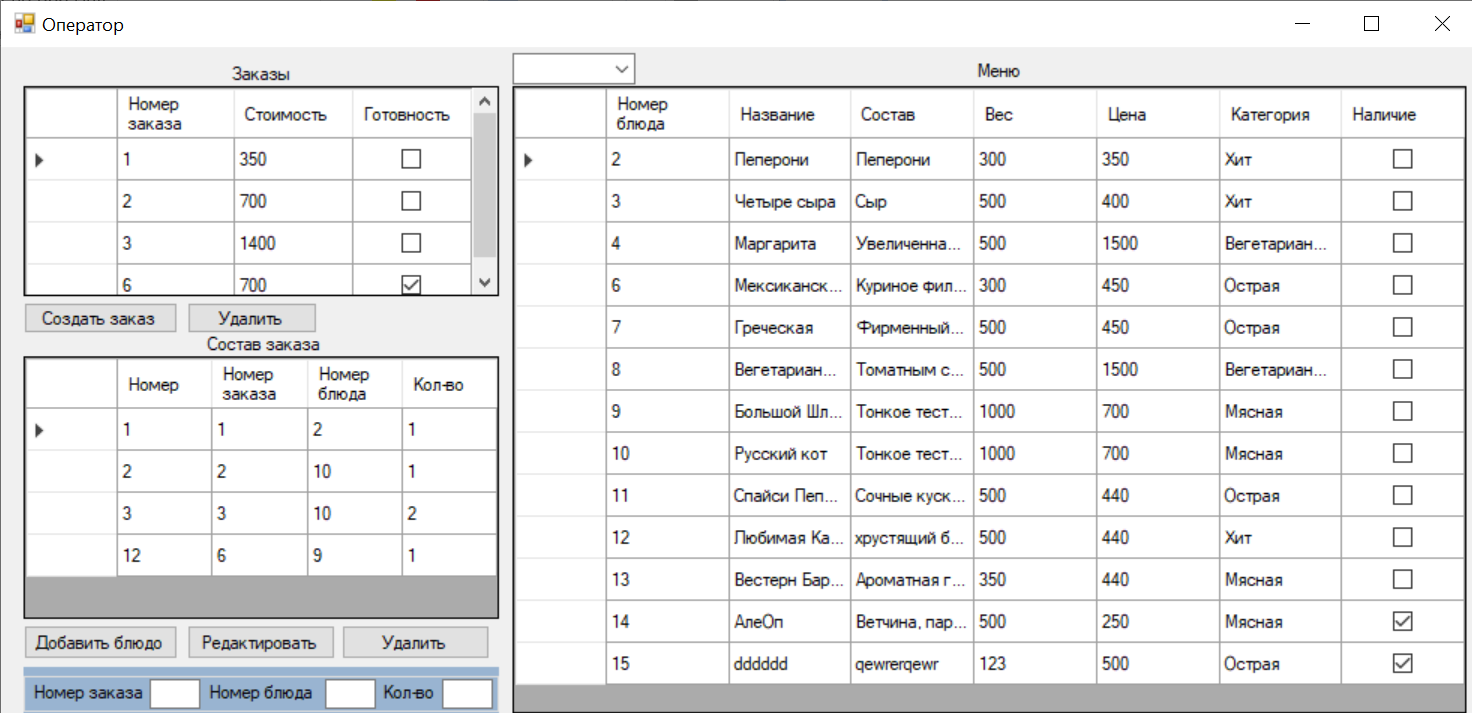


Рисунок 3 - Модуль оператора

## Тестирование информационной системы

В процессе тестирования было проведено модульное тестирование и написаны авторизированные тесты. (рис.4-9)

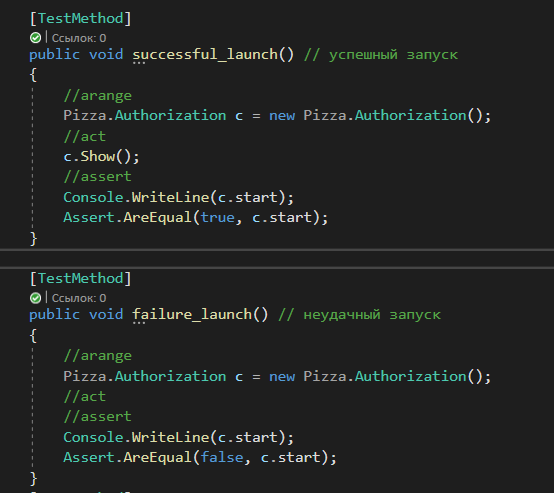


Рисунок 4 - UNIT-тест №1,2

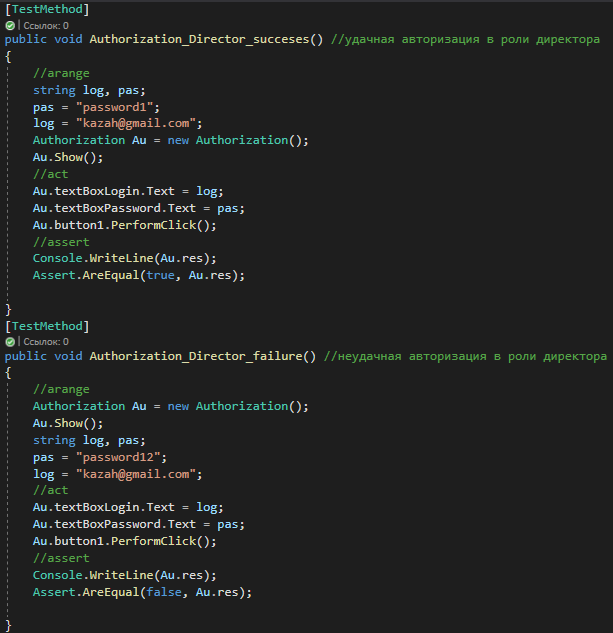


Рисунок 5 - UNIT-тест №3,4

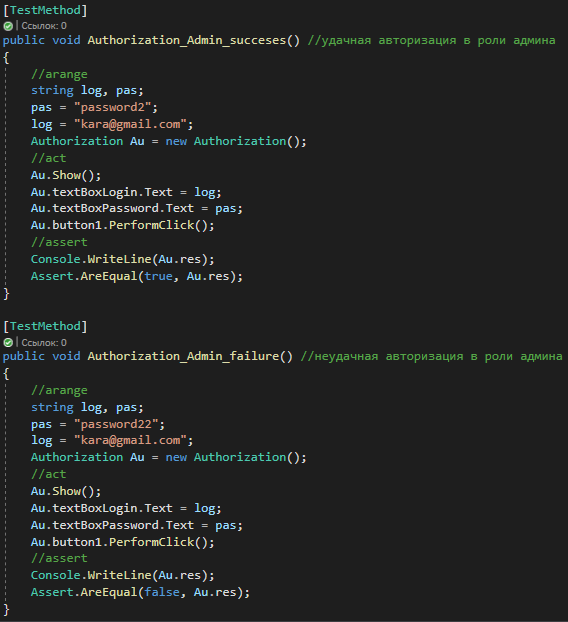


Рисунок 6 - UNIT-тест №5,6

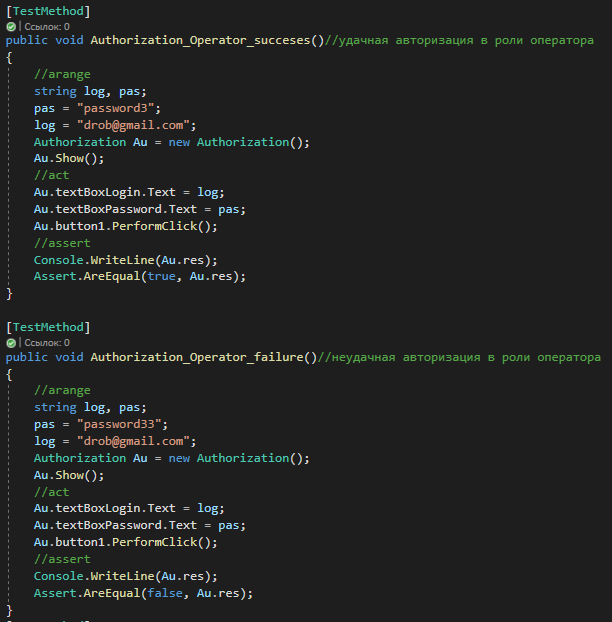


Рисунок 7 - UNIT-тест №7,8

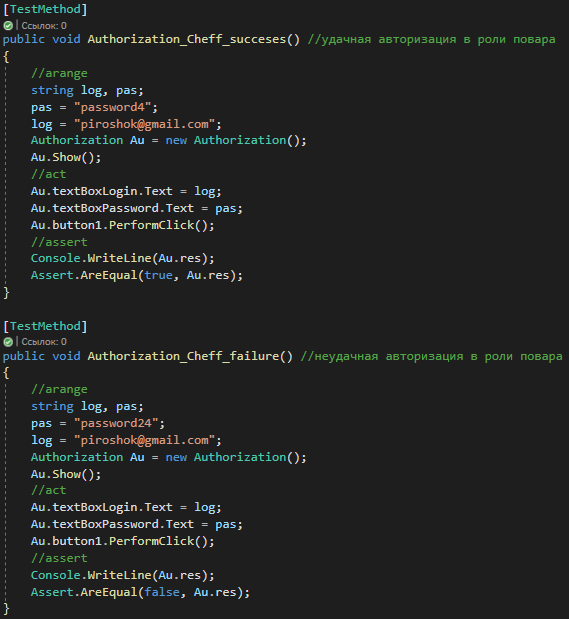


Рисунок 8 - UNIT-тест №9,10

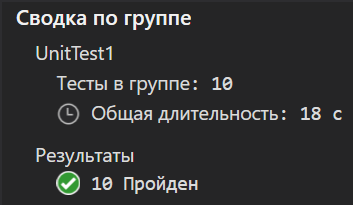


Рисунок - Результат UNIT тестирования

## Разработка технической документации

После разработки и тестирования было разработано техническое руководство на использование и FAQ.

## Оценка информационной системы

Оценка продукта производилась посредством общего анализа и сравнения с аналогами.

Таблица - Оценка информационной системы

|  |  |
| --- | --- |
| Критерий | Оценка по пятибалльной шкале |
| Практичность | 3 |
| Целостность | 3 |
| Корректность | 4 |
| Удобство обслуживания | 4 |
| Гибкость | 5 |
| Адаптируемость | 5 |
| Мобильность | 2 |
| Удобство эксплуатации | 4 |

Исходя из информации, указанной в таблице, можно понять, что информационная система далека до идеала, но обладает хорошей гибкостью и адаптируемостью, что позволяет легко ее модернизировать.

## Разработка технического задания на сопровождение информационной системы

После оценки ИС, можно было приступать к сопровождению и модернизации, но для этого требовалось установить и зафиксировать направления и области модернизации, для этого совместно с руководителем практики было разработано техническое задание на сопровождение ИС.

Основными направлениями стали: исправление багов в коде и увеличение функционала.

## Выполнение отладки программного кода

Баги в коде были в самом сложном модуле – модуле оператора. Суть заключалась в том, что при определенных условиях неправильно происходил подсчет позиций. В ходе анализа стало понятно, что функцию отлаживать нерационально, необходимо было кардинально менять логику и алгоритм вычисления, в следствии, полностью переписывать функцию.

## Разработка обучающей документации

В процессе разработки обучающей документации было разработано руководство пользователя.

## Оценка качества и надежности функционирования информационной системы

После отладки кода был проведен повторный анализ ИС, и результаты были в разы лучше предыдущих - программа работала абсолютно правильно, без ошибок, задержек и исключений. Были проведены повторные тесты по результатам которых можно с уверенностью сказать, что программа надежна.

## Осуществление сопровождения информационной системы

В процессе сопровождения ИС был разработан дополнительный модуль для обработки конечной информации. (рис.10)

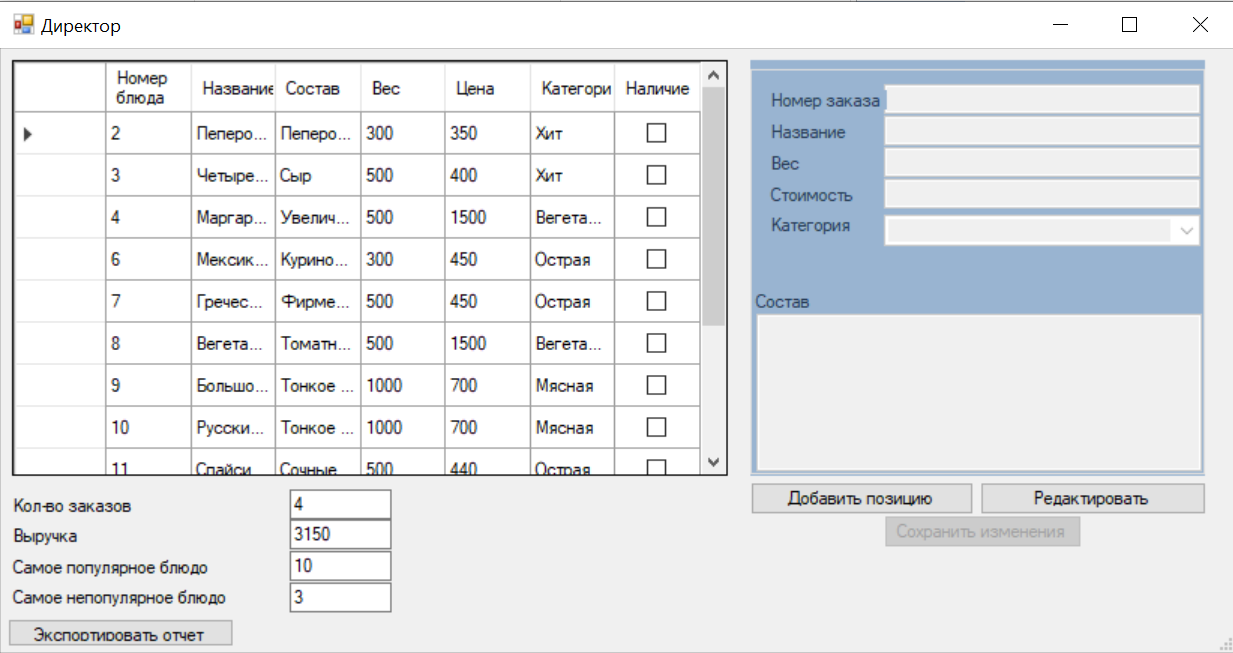


Рисунок 10 - Модуль обработки информации

# **ЗАКЛЮЧЕНИЕ**

В процессе производственной практики были развиты общие и профессиональные компетенции, проверена готовность к самостоятельной трудовой деятельности, а также подготовка к выполнению выпускной квалификационной работы в организациях различных организационно-правовых форм. Были закреплены приобретаемые теоретические знания, получен социально-психологический опыт адаптации на местах будущей работы.

В результате прохождения производственной практики на ООО «АСП Консалт» значительно расширены профессиональные навыки.   
Также были изучены основные принципы деятельности организации ООО «АСП Консалт». В ходе практики был поставлен ряд задач, которые были выполнены в полном объеме.

Результатом производственной практики является разработанная программа, отчет по производственной практике.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. ГОСТ Р 2.105-2019 «ОБЩИЕ ТРЕБОВАНИЯ К ТЕКСТОВЫМ ДОКУМЕНТАМ»
2. ГОСТ 7.32-2017 «ОТЧЕТ О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ»
3. Книга. Тестирование программного обеспечения. Базовый курс [Электронный ресурс] https://skillbox.ru/media/code/ (Дата обращения 05.04.2022)
4. Учебник. Приступая к работе с компонентом Database Engine [Электронный ресурс] https://docs.microsoft.com/ru-ru/sql/relational-databases/tutorial-getting-started-with-the-database-engine?view=sql-server-ver15 (Дата обращения 05.04.2022)
5. Учебник. Создание простого консольного приложения [Электронный ресурс] https://docs.microsoft.com/ru-ru/visualstudio/get-started/visual-basic/tutorial-console?view=vs-2022 (Дата обращения 05.04.2022)
6. Учебник. Создание средства просмотра изображений приложения Windows Forms в Visual Studio [Электронный ресурс] https://docs.microsoft.com/ru-ru/visualstudio/get-started/csharp/tutorial-windows-forms-picture-viewer-layout?view=vs-2022 (Дата обращения 05.04.2022)
7. Полное руководство по языку программирования С# [Электронный ресурс] https://metanit.com/sharp/tutorial/ (Дата обращения 05.04.2022)

# Приложение 1

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Pizza

{

public partial class Authorization : Form

{

public Authorization()

{

InitializeComponent();

}

PizzaDataSet.UsersDataTable dataUsers;

//Событие запуска приложения

private void Form1\_Load(object sender, EventArgs e)

{

//Строка подключения из файла конфигурации

string connectionString = @"Data Source=DESKTOP-U0O3POV\SQLEXPRESS;Initial Catalog=Pizza;Integrated Security=True";

//Создание и настройка объекта подключения к БД

SqlConnection connection = new SqlConnection();

connection.ConnectionString = connectionString;

//Контроль за удачным подключением

try

{

connection.Open();

}

catch (SqlException ex) //Обработка сбоя при подключении

{

switch (ex.Number) //Номер ошибки

{

case 17: MessageBox.Show("Неверное имя сервера"); break;

case 4060: MessageBox.Show("Неверное имя БД"); break;

case 18456: MessageBox.Show("Неверное имя пользователя или пароль"); break;

}

MessageBox.Show(ex.Message + " Уровень ошибки " + ex.Class);

Application.Exit();

}

catch (Exception ex) //Общий сбой при подключении

{

MessageBox.Show("Ошибка подключения " + ex.Message);

Application.Exit();

}

MessageBox.Show("Связь с сервером установлена");

//Дальнейшие действия с подключенной БД

//Получить все записи таблицы Users через адаптер usersTableAdapter

dataUsers = this.usersTableAdapter1.GetData();

//Отобразить полученные записи в компоненте

this.dataGridViewUsers.DataSource = dataUsers;

//Настроить компонент по ширине

dataGridViewUsers.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

}

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

{

string log, pas;

pas = this.textBoxPassword.Text;

log = this.textBoxLogin.Text; //Исходные данные

var filter = dataUsers.Where(p => p.Email == log && p.Password == pas);

if (filter.Count() == 0) //Логин и пароль не совпали

MessageBox.Show("Таких данных нет.");

else //Данные в бд есть

{

//Получение данных о пользователе и запись их в общий

//Первая и единственная запись с 0ым индексом

Total.ID\_User = filter.ElementAt(0).ID;

Total.ID\_Role = filter.ElementAt(0).ID\_Role;

switch (Total.ID\_Role)

{

case 2:

MessageBox.Show("Вы успешно авторизовались как администратор");

FormAdmin fa = new FormAdmin();

this.Hide();

fa.ShowDialog();

this.Show();

break;

case 1:

MessageBox.Show("Вы успешно авторизовались как директор");

FormDirector fd = new FormDirector();

this.Hide();

fd.ShowDialog();

this.Show();

break;

case 3:

MessageBox.Show("Вы успешно авторизовались как оператор");

FormOperator fo = new FormOperator();

this.Hide();

fo.ShowDialog();

this.Show();

break;

case 4:

MessageBox.Show("Вы успешно авторизовались как повар");

FormCheff fc = new FormCheff();

this.Hide();

fc.ShowDialog();

this.Show();

break;

}

}

}

private void textBoxLogin\_TextChanged(object sender, EventArgs e)

{

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Excel = Microsoft.Office.Interop.Excel;

using Word = Microsoft.Office.Interop.Word;

using System.Threading;

namespace Pizza

{

public partial class FormDirector : Form

{

enum ActionsAdmin { Insert, UpDate, Block }; //Возможные действия админа

ActionsAdmin action;

PizzaDataSet.MenuDataTable dataMenu;

PizzaDataSet.OrdersDataTable dataOrders;

PizzaDataSet.OrdersRow rowOrders;

PizzaDataSet.ReceiptDataTable dataReceipt;

PizzaDataSet.ReceiptRow rowReceipt;

Excel.Application excelApp; //Сервер Excel

Excel.Workbook excelBook; //Отдельная книга

Excel.Worksheet excelSheet; //Один лист

Excel.Range excelCells; //Ячейки

Word.Application wordApp; //ceрвер

Word.Document wordDoc; //документ

Word.Bookmark wordBookmarks; //закладки

public FormDirector()

{

InitializeComponent();

}

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

{

dataMenu = this.menuTableAdapter1.GetData();

dataOrders = this.ordersTableAdapter1.GetData();

dataReceipt = this.receiptTableAdapter1.GetData();

this.dataGridView1.DataSource = dataMenu;

dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

this.dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

this.dataGridView1.Columns["Photo"].Visible = false;

this.dataGridView1.Columns["ID"].HeaderText = "Номер блюда";

this.dataGridView1.Columns["Title"].HeaderText = "Название";

this.dataGridView1.Columns["Composition"].HeaderText = "Состав";

this.dataGridView1.Columns["Weight"].HeaderText = "Вес";

this.dataGridView1.Columns["Cost"].HeaderText = "Цена";

this.dataGridView1.Columns["Caterogy"].HeaderText = "Категория";

this.dataGridView1.Columns["Active"].HeaderText = "Наличие";

//this.dataGridView1.Columns["Photo"].Visible = false;

//this.dataGridView1.Columns["Id"].Visible = false;

//this.dataGridView1.Columns["Composition"].Visible = false;

this.groupBoxMenu.Enabled = false; //заблокировать поля ввода

buttonSave.Enabled = false; //раззрешить кнопку внесения изменений

textBoxIDMenu.Enabled = false;

textBoxNumOrders.Text = dataOrders.Count.ToString(); // кол-во заказов

int money = 0; // выручка

for (int i = 0; i <= 1021; i++)

{

rowOrders = dataOrders.FindByID(i);

if(rowOrders != null)

{

money += rowOrders.Price;

}

}

textBoxMoney.Text = money.ToString();

int Bludo1 = 0;

int Bludo2 = 0;

int Bludo3 = 1;

for (int j = 2; j <=15; j++)

{

Bludo1 = 0;

for (int i = 0; i <= 1031; i++)

{

rowReceipt = dataReceipt.FindByID(i);

if (rowReceipt != null)

{

if (rowReceipt.ID\_Dish == j)

{

Bludo1++;

}

}

}

if (Bludo1 > Bludo2)

{

textBoxMostPopular.Text = j.ToString();

Bludo2 = Bludo1;

}

if (Bludo1 < Bludo3)

{

textBoxAnPopular.Text = j.ToString();

Bludo3 = Bludo1;

}

}

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

int numRow = e.RowIndex; //Получить номер выбранной строки

this.textBoxIDMenu.Text = dataGridView1.Rows[numRow].Cells[0].Value.ToString();

this.textBoxTitle.Text = dataGridView1.Rows[numRow].Cells[1].Value.ToString();

this.textBoxComposition.Text = dataGridView1.Rows[numRow].Cells[2].Value.ToString();

this.textBoxWeight.Text = dataGridView1.Rows[numRow].Cells[3].Value.ToString();

this.textBoxCost.Text = dataGridView1.Rows[numRow].Cells[4].Value.ToString();

this.comboBoxCategory.Text = dataGridView1.Rows[numRow].Cells[5].Value.ToString();

}

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

{

this.groupBoxMenu.Enabled = true; //Активность контейнера

//Очистка всех элементов в контейнере

foreach (var item in this.groupBoxMenu.Controls)

{

if (item is TextBox) (item as TextBox).Text = "";