**Содержание**

1. Постановка задачи 7

2. Структура таблиц в БД 9

3. Схема базы данных 10

4. Описание программных модулей 11

5. Листинг программы 24

1. **Постановка задачи**

Разработать информационную систему по ведению базы данных «**Автомастерская**» содержащую информацию об автомобилях и ремонтных работах.

ЗАДАНИЕ

1. База данных должна содержать следующую информацию:
   * таблица «**Автомеханик**»: **номер**, **фамилия, имя, отчество, стаж, разряд**
   * таблица «**Автомобиль**»: **номер**, **марка,** **название, тип кузова, год выпуска**.
   * таблица «**Ремонт**»: **дата, время, стоимость**. Данная таблица должна быть связана с автомехаником и автомобилем.
2. Определить ключи (если необходимо, добавьте поля). Автоматизировать поля ввода данных с помощью выпадающих списков или маски ввода. Предусмотреть контроль ввода данных, применив блокировку ввода некорректных данных.
3. Ввести записи в таблицы согласно приложению 1 и дополнить своими реальными данными в количестве 10 записей, в каждую таблицу соответственно.
4. Создайте форму для добавления информации, и форму для удаления и редактирования данных в существующих таблицах.
5. Создайте отдельную форму «**Ремонты**», которая должна содержать следующие поля:  
   **номер, ФИО автомеханика, название и марка автомобиля, дата и время ремонта**.
6. Реализовать создание отчета на форме «**Ремонты**» выделенной записи в формат MS Word 2003/2007 со всей информацией.

Приложение 1

Данные об автомеханиках

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **номер** | **фамилия** | **имя** | **отчество** | **стаж** | **разряд** |
| 10 | Савельев | [Вадим](https://www.kinopoisk.ru/name/570253/) | Валерьевич | 4 года | 4 разряд |
| 12 | Янышев | Олег | Валентинович | 2 года | 3 разряд |
| 17 | Лагин | Леонид | Юрьевич | 10 лет | 5 разряд |

Данные об автомобилях

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **номер** | **марка** | **название** | **тип кузова** | **год выпуска** |
| 3 | Ford | Focus | седан | 2011 |
| 4 | УАЗ | Patriot | универсал | 2014 |
| 5 | Opel | Astra | хетчбэк | 2015 |

Обязательно продумать реализацию хранения стажа у автомеханика в базе данных (например, 2 года, 7 месяцев и 15 дней).

**Общее задание**

1. АИС должна состоять из приложения (технология WinForm или WPF), написанного на языке программирования С# и базы данных в формате файла MS Access 2003/2007.
2. Приложение должно обеспечивать доступ к базе данных и состоять из следующих форм: главная форма, форма для редактирования и удаления данных, форма для добавления информации в БД.
3. Для манипулирования данными необходимо использовать только язык SQL (добавление/удаление/обновление информации). В каждой таблице необходимо определить первичный ключ.
4. Имена столбцов должны быть написаны на иностранном языке без использования транслита.
5. Информация в БД может быть записана с использованием русского или английского языка.
6. Для отображения базы данных на формах в приложении использовать только компонент платформы .Net Framework - DataGridView.
7. **Структура таблиц в БД**

Таблица 2.1 - Описание таблицы «car»

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название поля** | **Тип данных** | **Описание поля** |
| 1 | car\_id | integer | Идентификатор поля. Первичный ключ |
| 2 | car\_number | integer | Номер автомобиля |
| 3 | car\_mark | string(65) | Марка автомобиля |
| 4 | car\_name | string(65) | Модель автомобиля |
| 5 | car\_type | string(65) | Тип кузова |
| 6 | car\_year | integer | Год выпуска автомобиля |

Таблица 2.2 - Описание таблицы «mechanic»

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название поля** | **Тип данных** | **Описание поля** |
| 1 | mechanic\_id | integer | Идентификатор поля. Первичный ключ |
| 2 | mechanic\_number | integer | Номер механика |
| 3 | mechanic\_surname | string(65) | Фамилия механика |
| 4 | mechanic\_name | string(65) | Имя механика |
| 5 | mechanic\_patronymic | string(65) | Отчество механика |
| 6 | mechanic\_exp | string(65) | Стаж механика |
| 7 | mechanic\_rank | integer | Разряд механика |

Таблица 2.3 - Описание таблицы «repair»

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название поля** | **Тип данных** | **Описание поля** |
| 1 | repair\_id | integer | Идентификатор поля. Первичный ключ |
| 2 | mechanic\_id | integer | Идентификационный номер механика |
| 3 | car\_id | integer | Идентификационный номер автомобиля |
| 4 | repair\_date | string(65) | Дата ремонта |
| 5 | repair\_time | string(65) | Время ремонта |
| 6 | repair\_cost | string(65) | Стоимость ремонта |

1. **Схема базы данных**

На рисунке ниже (рисунок 3.1) представлена схема базы данных и используемые связи для соединения таблиц.

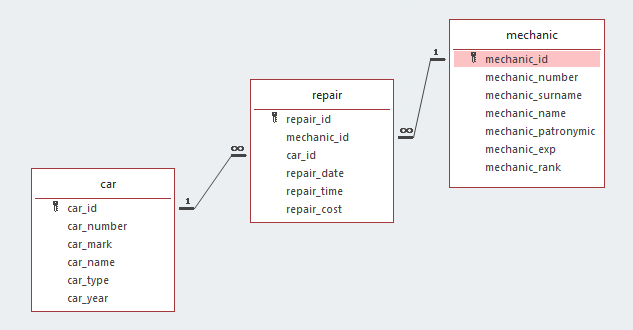


Рисунок 3.1 – Схема базы данных

1. **Описание программных модулей**

Для запуска программы необходимо запустить исполняющий файл «Garage.exe», находящийся в папке «bin».

После запуска программы пользователь попадает на главную форму, на которой есть возможность перехода на другие формы. Главная форма программы представлена на рисунке 4.1.

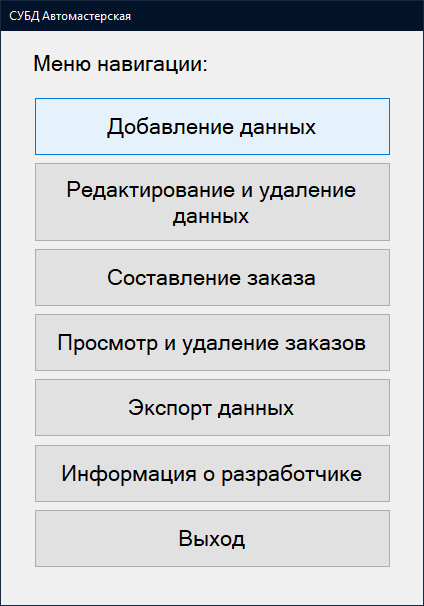


Рисунок 4.1 – Главная форма программы

При нажатии на кнопку «Добавление данных» пользователь переходит на форму добавления данных в соответствии с рисунком 4.2.

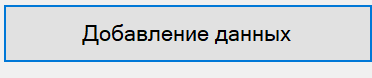


Рисунок 4.2 – Кнопка добавления данных

После перехода пользователю доступны формы добавления данных о машинах и механиках, в соответствии с рисунками 4.3 и 4.4.

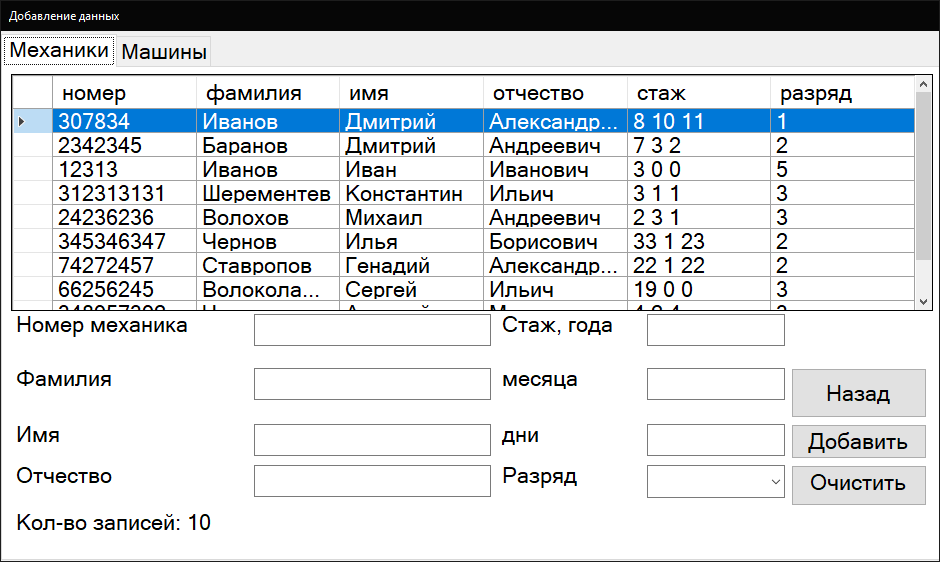


Рисунок 4.3 – Добавление данных о механиках

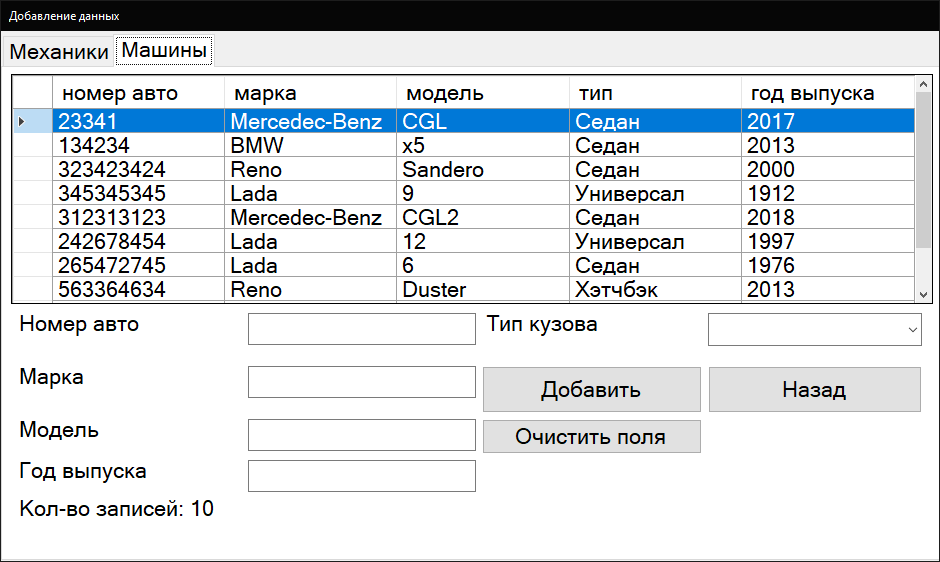


Рисунок 4.4 – Добавление данных о машинах

При нажатии на кнопку «Назад» пользователь переходит на главную форму, в соответствии с рисунком 4.5.



Рисунок 4.5 – Кнопка назад

При нажатии на кнопку «Очистить» очищаются текстовые поля формы, в соответствии с рисунком 4.6.



Рисунок 4.6 – Кнопка очистки

При нажатии на кнопку «Добавить» программа добавляет новые записи в базу данных, в соответствии с рисунком 4.7.



Рисунок 4.7 – Кнопка добавления

Если не все поля будут заполнены программа укажет на эту ошибку диалоговым окном, в соответствии с рисунком 4.8, так же если все заполнено успешно, появляется диалоговое окно, в соответствии с рисунком 4.9.

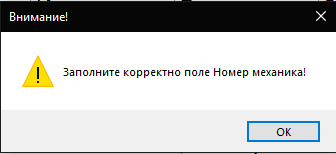


Рисунок 4.8 – Сообщение пользователю об ошибке

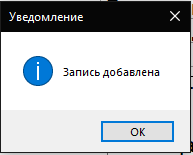


Рисунок 4.9 – Сообщение пользователю об успешном добавлении

При нажатии на копку «Редактирование и удаление данных», в соответствии с рисунком 4.10, пользователь переходит на форму, которая позволяет удалять или обновлять информацию в базе данных, в соответствии с рисунками 4.11 и 4.12.

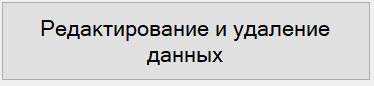


Рисунок 4.10 – Кнопка перехода на форму редактирования записей

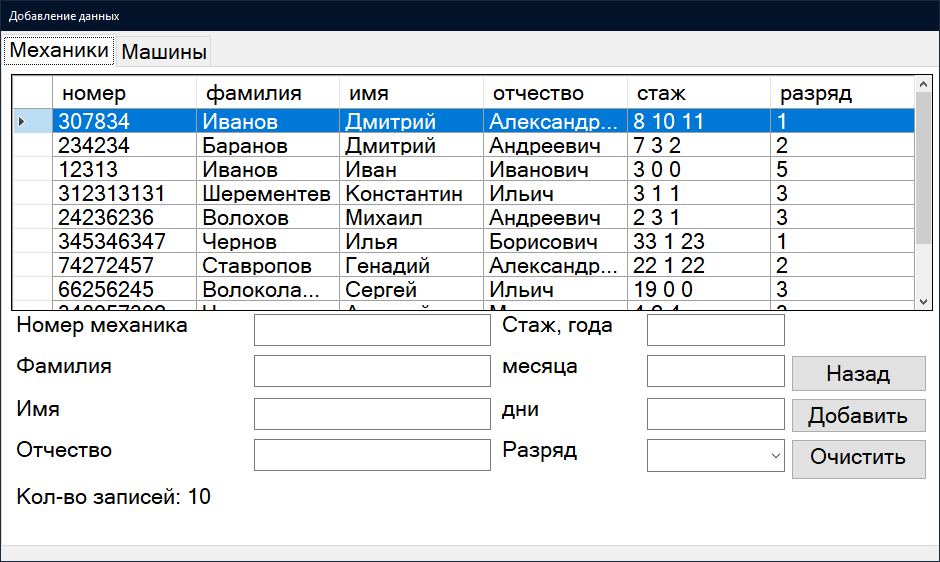


Рисунок 4.11 – Редактирование информации о механиках

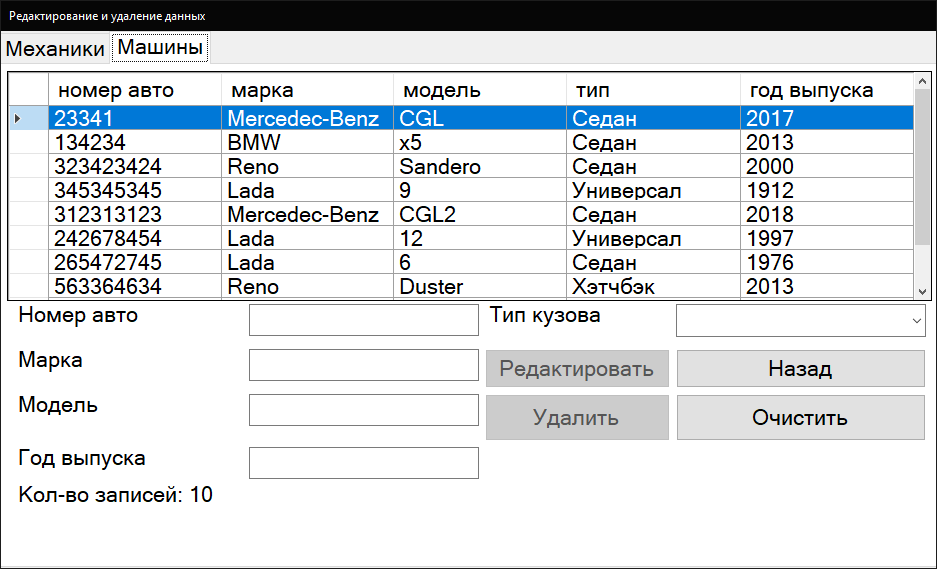


Рисунок 4.12 – Редактирование информации о машинах

При нажатии на кнопку «Назад» пользователь переходит на главную форму, в соответствии с рисунком 4.13.



Рисунок 4.13 – Кнопка назад

При нажатии на кнопку «Очистить» очищаются текстовые поля формы, в соответствии с рисунком 4.14.



Рисунок 4.14 – Кнопки очистки

Для того чтобы удалить запись из базы данных пользователь должен выделить необходимую ему строку и нажать кнопку «Удалить», в соответствии с рисунком 4.15.

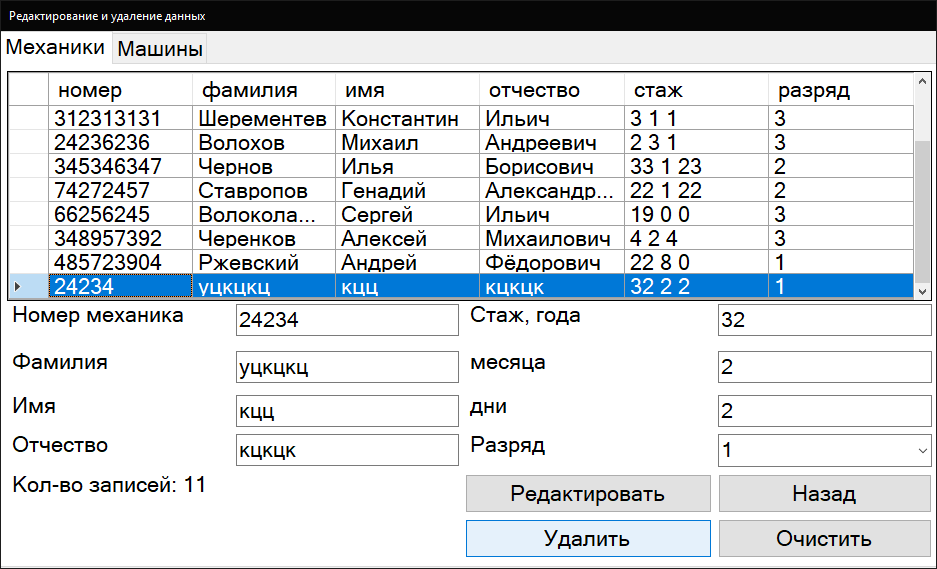


Рисунок 4.15 – Удаление записи

Если пользователь не заполнил какие-то поля то программа предупредит об этом в соответствии с рисунком 4.16, так же если удаление выполнено успешно, появится диалоговое окно, в соответствии с рисунком 4.17.

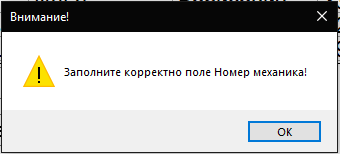


Рисунок 4.16 – Сообщение об ошибке

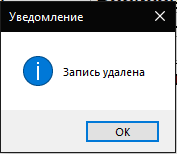


Рисунок 4.17 – Сообщение об успешном удалении

При нажатии на кнопку «Составление заказа» в соответствии с рисунком 4.18 пользователь переходит на форму записи тренировок в соответствии с рисунком 4.19.

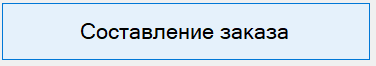


Рисунок 4.18 – Кнопка «Составления заказа»

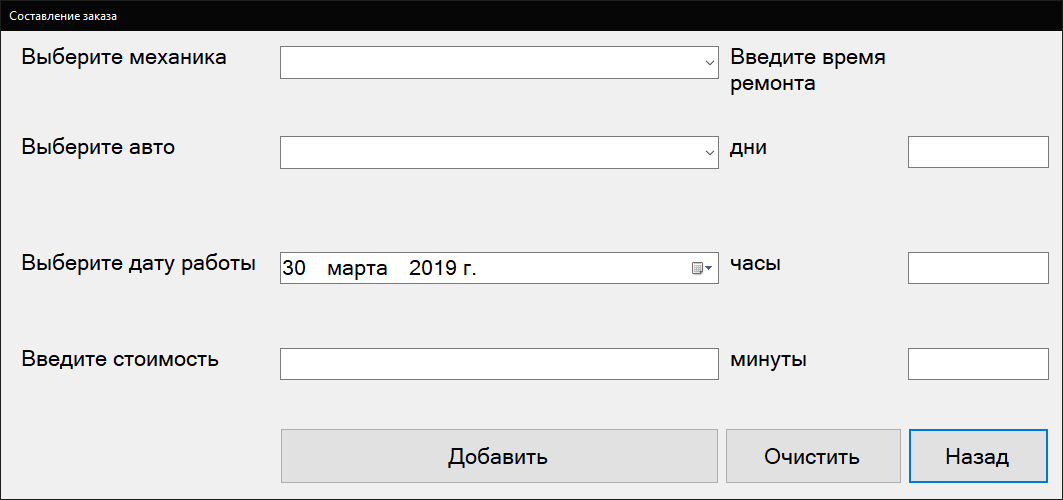


Рисунок 4.19 – Форма записи на тренировку

Для того чтобы составить заказ пользователь должен заполнить корректно все поля, если же при заполнении будет допущена ошибка, то программа сообщит об этом в диалоговом окне, в соответствии с рисунком 4.20, если же будет допущена ошибка в заполнении времени ремонта, программа так же сообщит об этом в диалоговом окне, в соответствии с рисунком 4.21. Когда все поля будут корректно заполнены при нажатии на кнопку добавить, в соответствии с рисунком 4.22, появится диалоговое окно, сообщающее об успешном добавлении записи, в соответствии с рисунком 4.23.

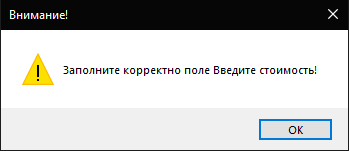


Рисунок 4.20 – Предупреждение о некорректном заполнении поля

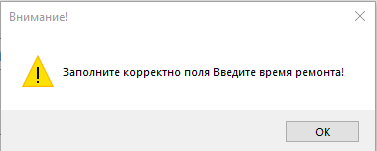


Рисунок 4.21 – Предупреждение о неправильном времени



Рисунок 4.22 – Кнопка добавления записи

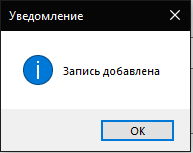


Рисунок 4.23 – Сообщение об удачной записи

Для очистки полей формы необходимо нажать кнопку «Очистить», в соответствии с рисунком 4.24, после успешной очистки появится уведомляющее об этом диалоговое окно, в соответствии с рисунком 4.25.



Рисунок 4.24 – Кнопка «Очистить»

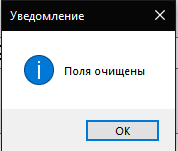


Рисунок 4.25 – сообщение об успешной очистке полей

При нажатии на кнопку «Просмотр и удаление заказов» в главном меню, в соответствии с рисунком 4.26, пользователь перейдёт на форму, на которой сможет посмотреть все записи на тренировку, в соответствии с рисунком 4.27.

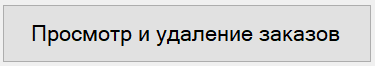


Рисунок 4.26 – Кнопка «Просмотр и удаление заказов»

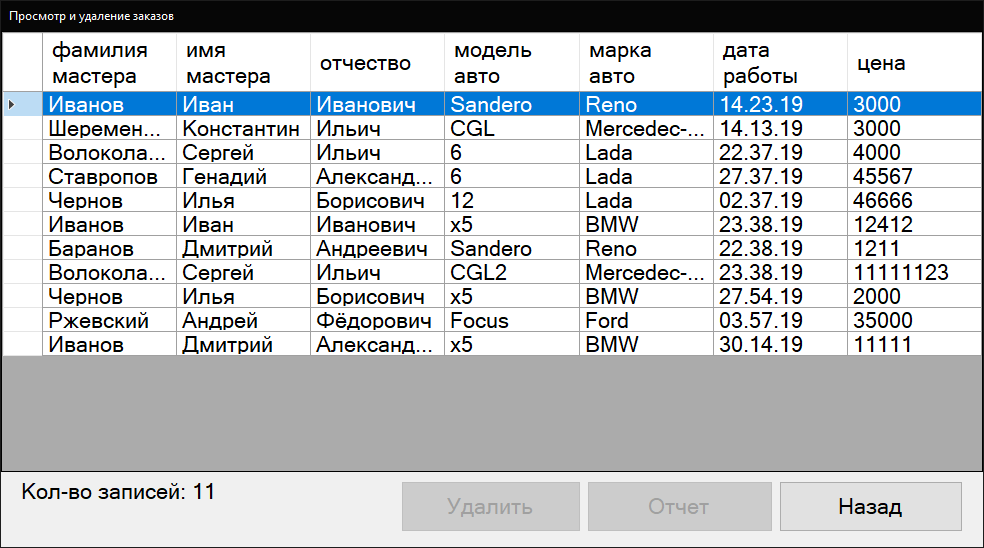


Рисунок 4.27 – Форма «Просмотр и удаление заказов»

При выборе в имеющейся таблице строки активируются кнопки «Удалить» и «Отчет», в соответствии с рисунком 4.28, при нажатии на кнопку удалить выделенная запись удалится и появиться диалоговое окно, уведомляющее об этом, в соответствии с рисунком 4.29. При нажатии на кнопку «Отчет» в папке с программой появится файл формата docx с названием «Отчет», в соответствии с рисунками 4.30 и 4.31.



Рисунок 4.28 – Активация кнопок «Удалить» и «Отчет»

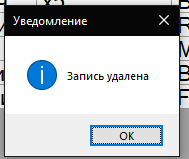


Рисунок 4.29 – Диалоговое окно, уведомляющее об успешном удалении данных

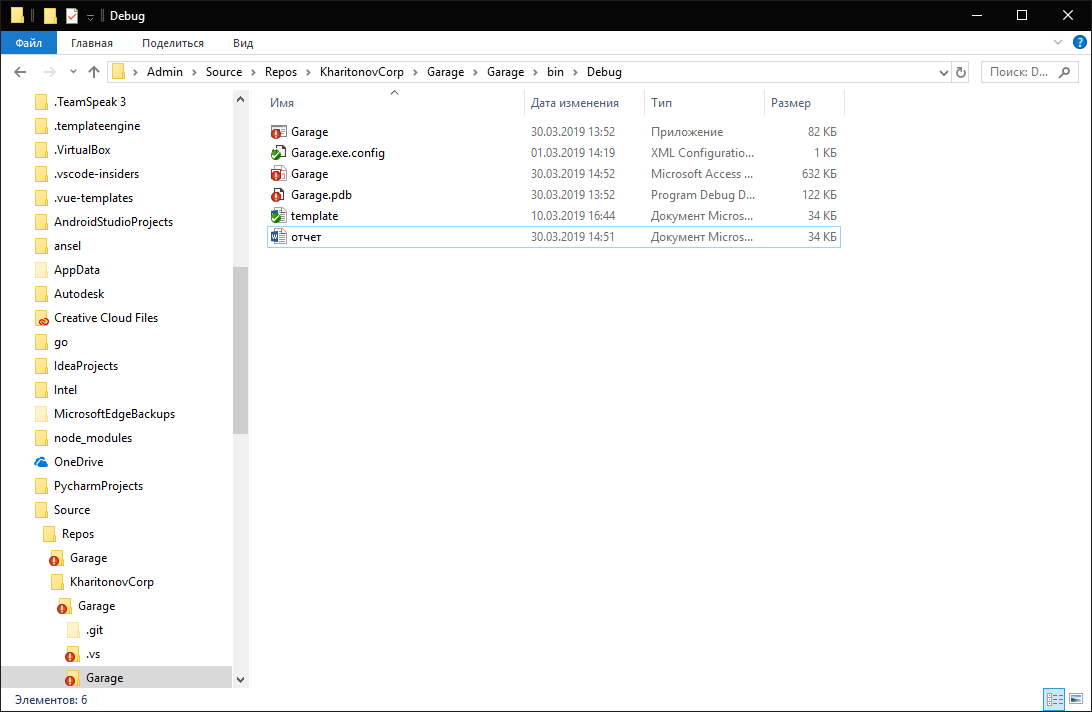


Рисунок 4.30 – «отчет» в файловой системе windows

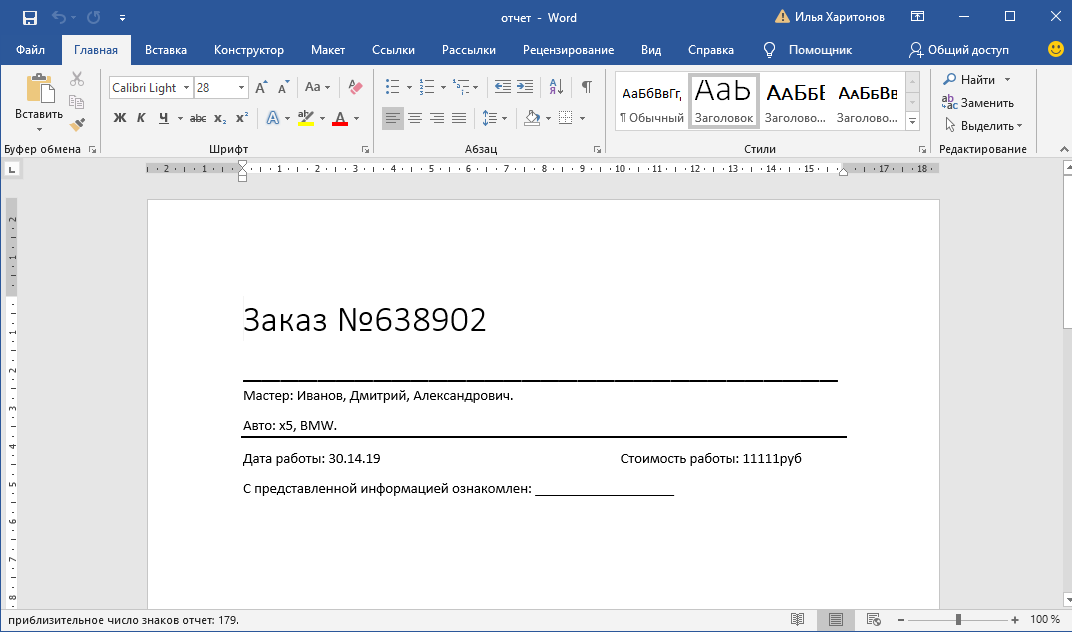


Рисунок 4.31 – Отчет

При нажатии на кнопку «Назад», в соответствии с рисунком 4.32 осуществляется переход на главную форму.



Рисунок 4.32 – Кнопка «Назад»

При нажатии на кнопку «Информация о разработчике», в соответствии с рисунком 4.33, будет осуществлен переход на форму с информацией о разработчике данной программы, в соответствии с рисунком 4.34.

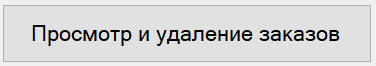


Рисунок 4.33 – Кнопка «Информация о разработчике»

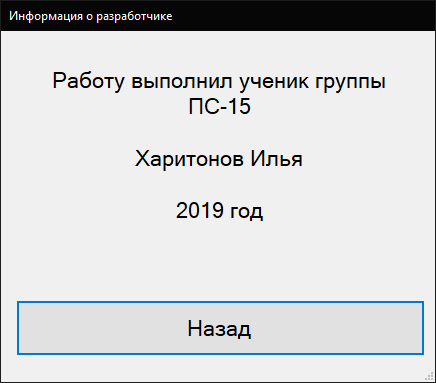


Рисунок 4.34 – Форма с информацией о разработчике

При нажатии на кнопку «Экспорт данных», в соответствии с рисунком 4.35, откроется лист Microsoft Excel со всей информацией из базы данных, в соответствии с рисунком 4.36 и диалоговое окно с предложением сохранить файл, в соответствии с рисунком 4.37.

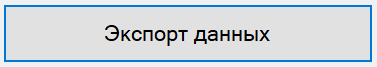


Рисунок 4.35 – Кнопка «Экспорт данных»

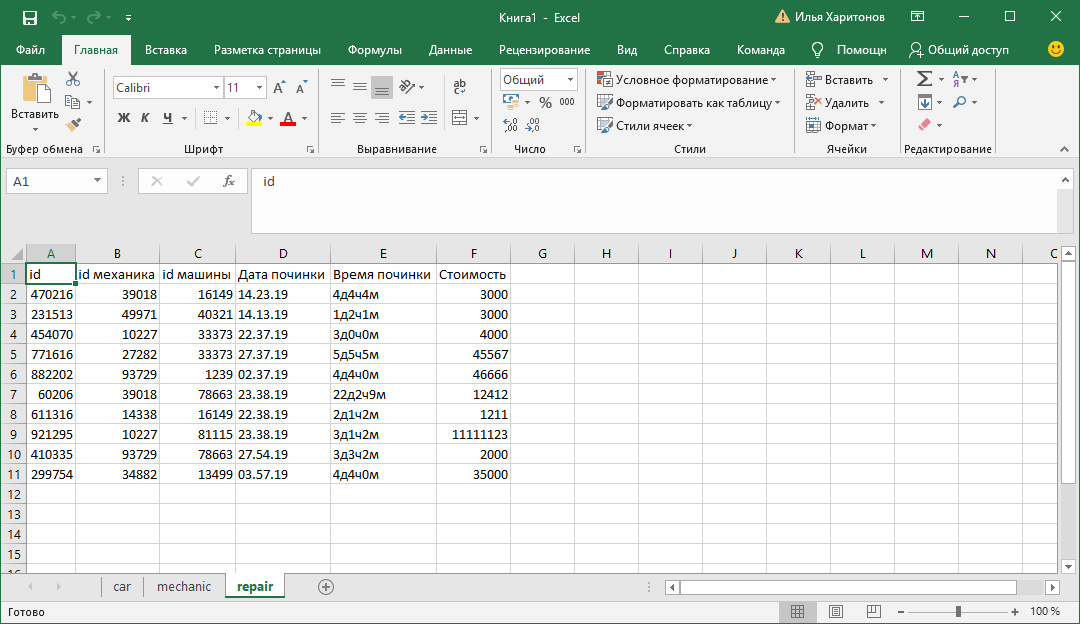


Рисунок 4.36 – База данных в excel

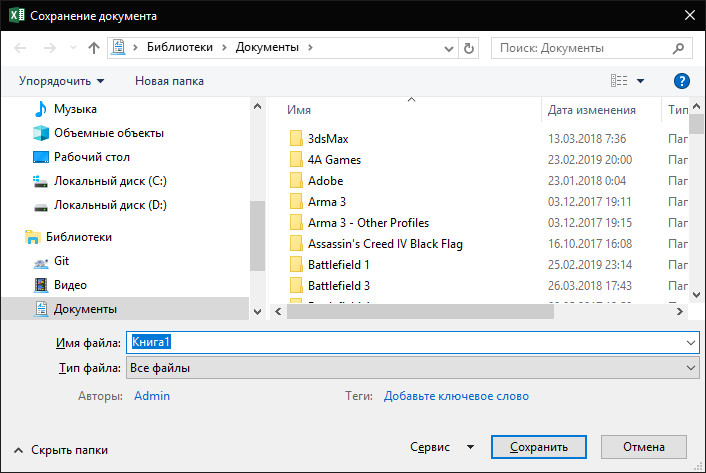


Рисунок 4.37 – Предложение сохранить excel файл

При нажатии на кнопку «Выход», в соответствии с рисунком 4.38, происходит завершение работы программы.

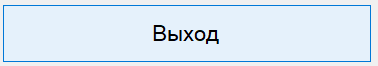


Рисунок 4.38 – Кнопка завершения работы

1. **Листинг программы**

Form1.cs – главная форма

// This is an open source non-commercial project. Dear PVS-Studio, please check it.

// PVS-Studio Static Code Analyzer for C, C++, C#, and Java: http://www.viva64.com

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Garage

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

InitializeInterface();

}

private void InitializeInterface()

{

this.Text = "СУБД Автомастерская";

label1.Text = "Меню навигации:";

button1.Text = "Добавление данных";

button2.Text = "Редактирование и удаление данных";

button3.Text = "Составление заказа";

button4.Text = "Просмотр и удаление заказов";

button5.Text = "Экспорт данных";

button6.Text = "Информация о разработчике";

button7.Text = "Выход";

}

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

{

var result = new DialogResult();

result = MessageBox.Show("Вы действительно желаете выйти?", "Внимание!", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

Application.Exit();

}

}

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

{

this.Hide();

Form AddForm = new Form2();

AddForm.Show();

}

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

{

this.Hide();

Form EditDelForm = new Form3();

EditDelForm.Show();

}

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

{

this.Hide();

Form AddDealForm = new Form4();

AddDealForm.Show();

}

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

{

this.Hide();

Form DelDealForm = new Form5();

DelDealForm.Show();

}

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

{

Form OutputInfoForm = new Form6();

OutputInfoForm.ShowDialog();

}

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

{

Form CreatorInfoForm = new Form7();

CreatorInfoForm.ShowDialog();

}

}

}

Form2.cs – форма добавления информации

// This is an open source non-commercial project. Dear PVS-Studio, please check it.

// PVS-Studio Static Code Analyzer for C, C++, C#, and Java: http://www.viva64.com

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.OleDb;

namespace Garage

{

//Форма добавления данных

public partial class Form2 : Form

{

//Инициализация переменных используемых в дальнейшем

string car = "car";

string mechanic = "mechanic";

//Инициализия элементов и компонентов формы

public Form2()

{

InitializeComponent();

InitializeInterface();

DataGridREView(car);

DataGridREView(mechanic);

}

//Метод инициализации элементов интерфейса

private void InitializeInterface()

{

//Названия формы и контролов

this.Text = "Добавление данных";

tabPage1.Text = "Механики";

tabPage2.Text = "Машины";

//Контрол механики

button1.Text = "Назад";

button2.Text = "Добавить";

button3.Text = "Очистить поля ввода";

label1.Text = "Номер механика";

label2.Text = "Фамилия";

label3.Text = "Имя";

label4.Text = "Отчество";

label5.Text = "Стаж, года";

label6.Text = "Разряд";

label12.Text = "месяца";

label13.Text = "дни";

string[] ranks = { "1", "2", "3", "4", "5", "6" };

comboBox1.Items.AddRange(ranks);

//Контрол машины

button6.Text = "Назад";

button5.Text = "Добавить";

button4.Text = "Очистить поля ввода";

label7.Text = "Номер авто";

label8.Text = "Марка";

label9.Text = "Модель";

label10.Text = "Год выпуска";

label11.Text = "Тип кузова";

string[] car\_types = { "Седан", "Хэтчбэк", "Универсал", "Лифтбэк", "Купе", "Кабриолет", "Родстер", "Тарга" };

comboBox2.Items.AddRange(car\_types);

//Получение количества записей

GetNoteCount(car);

GetNoteCount(mechanic);

}

//Метод очистки текстовых полей

private void ClearTextBox(string table)

{

//Если контрол механики

if (table == "mechanic")

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

textBox10.Text = "";

textBox11.Text = "";

comboBox1.Text = "";

}

//Если контрол машины

else if (table == "car")

{

textBox6.Text = "";

textBox7.Text = "";

textBox8.Text = "";

textBox9.Text = "";

comboBox2.Text = "";

}

}

//Метод получения количества записей

private void GetNoteCount(string table)

{

//Если контрол мехиники

if (table == "mechanic")

{

try

{

//Подключение к бд

string con = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection oleDbConn = new OleDbConnection(con);

oleDbConn.Open();

//Создание запроса на получение количества записей

OleDbCommand sql = new OleDbCommand("SELECT COUNT(\*) FROM " + table + ";", oleDbConn);

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

//Ввод количества записей в текстовый элемент на форму

label14.Text = "Кол-во записей: " + (int)sql.ExecuteScalar();

oleDbConn.Close();

}

catch (Exception ex)

{

//Уведомление об ошибке

MessageBox.Show(ex.ToString());

}

}

//Если контрол машины

if (table == "car")

{

try

{

//Подключение к бд

string con = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection oleDbConn = new OleDbConnection(con);

oleDbConn.Open();

//Создание запроса на получение количества записей

OleDbCommand sql = new OleDbCommand("SELECT COUNT(\*) FROM " + table + ";", oleDbConn);

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

//Ввод количества записей в текстовый элемент на форме

label15.Text = "Кол-во записей: " + (int)sql.ExecuteScalar();

oleDbConn.Close();

}

catch (Exception ex)

{

//Уведомление об ошибке

MessageBox.Show(ex.ToString());

}

}

}

//Преобразование стажа в единый текст

private string GetMechanicExp()

{

string time = textBox5.Text + " " + textBox10.Text + " " + textBox11.Text;

return time;

}

//Метод добавления записей

private void AddData(string table)

{

try

{

//Получение случайного числа, используемого, как id

Random rnd = new Random();

int temp = rnd.Next(0, 99999);

//Подключегие к бд

string con = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection oleDbConn = new OleDbConnection(con);

DataTable dt = new DataTable();

oleDbConn.Open();

//Если контрол механики

if (table == "mechanic")

{

//Создание запроса на добавление записи

OleDbCommand sql = new OleDbCommand("INSERT INTO " + table + " (mechanic\_id, mechanic\_number, mechanic\_surname, mechanic\_name, mechanic\_patronymic, mechanic\_exp, mechanic\_rank) VALUES (" + Convert.ToInt32(temp) + " , '" + textBox1.Text + "' , '" + textBox2.Text + "' , '" + textBox3.Text + "' , '" + textBox4.Text + "' , '" + GetMechanicExp() + "' , '" + comboBox1.Text + "')");

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

}

//Елси контрол машины

else if (table == "car")

{

//Создание запроса на добавление записиы

OleDbCommand sql = new OleDbCommand("INSERT INTO " + table + " (car\_id, car\_number, car\_mark, car\_name , car\_type, car\_year) VALUES (" + Convert.ToInt32(temp) + " , '" + textBox6.Text + "' , '" + textBox7.Text + "' , '" + textBox8.Text + "' , '" + comboBox2.Text + "' , '" + textBox9.Text + "')");

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

}

//Закрытие подключения

oleDbConn.Close();

}

catch (Exception ex)

{

//Уведомление об ошибке

MessageBox.Show(ex.ToString());

}

//Обновление количества записей

GetNoteCount(car);

GetNoteCount(mechanic);

}

//Получение данных в DataGridView

private void DataGridREView(string table)

{

try

{

//Подключение к бд

string con = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection oleDbConn = new OleDbConnection(con);

DataTable dt = new DataTable();

oleDbConn.Open();

//Создание запроса на получение информации

OleDbCommand sql = new OleDbCommand("SELECT \* FROM " + table + ";");

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(sql);

da.Fill(dt);

//Ввод информации в контрол машины

if (table == "car")

{

dt.Columns["car\_id"].ColumnName = "id";

dt.Columns["car\_number"].ColumnName = "номер авто";

dt.Columns["car\_mark"].ColumnName = "марка";

dt.Columns["car\_name"].ColumnName = "модель";

dt.Columns["car\_type"].ColumnName = "тип";

dt.Columns["car\_year"].ColumnName = "год выпуска";

dataGridView2.DataSource = dt;

dataGridView2.Columns[0].Visible = false;

}

//Ввод информации в контрол мехники

if (table == "mechanic")

{

dt.Columns["mechanic\_id"].ColumnName = "id";

dt.Columns["mechanic\_number"].ColumnName = "номер";

dt.Columns["mechanic\_surname"].ColumnName = "фамилия";

dt.Columns["mechanic\_name"].ColumnName = "имя";

dt.Columns["mechanic\_patronymic"].ColumnName = "отчество";

dt.Columns["mechanic\_exp"].ColumnName = "стаж";

dt.Columns["mechanic\_rank"].ColumnName = "разряд";

dataGridView1.DataSource = dt;

dataGridView1.Columns[0].Visible = false;

}

//Закрытие подключения

oleDbConn.Close();

}

catch (Exception ex)

{

//Уведомление об ошибке

MessageBox.Show(ex.ToString());

}

}

//Кнопка перехода на главную форму

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

{

GoBackToMainForm();

}

//Метод перехода на голавную форму

private void GoBackToMainForm()

{

this.Close();

Form MainForm = new Form1();

MainForm.Show();

}

//Кнопка очистки полей ввода

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

{

//Подтверждение запроса

var result = new DialogResult();

result = MessageBox.Show("Вы действительно очистить поля ввода?", "Внимание!", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

//Очистка полей вода, уведомление

ClearTextBox(mechanic);

MessageBox.Show("Поля очищены", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

//Кнопка добавления записи

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

{

//Проверка заполненности полей

var result = TextCheck(mechanic);

if (result == 1)

{

return;

}

else if (result == 0)

{

//Добавление записи, очистка полей, обновление DataGridView, уведомление

AddData(mechanic);

ClearTextBox(mechanic);

DataGridREView(mechanic);

MessageBox.Show("Запись добавлена", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

//Кнопка добавление записи

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

{

//Проверка заполненности полей

var result = TextCheck(car);

if (result == 1)

{

return;

}

else if (result == 0)

{

//Добавление записи в бд, очистка полей ввода, обновление DataGridView, уведомление

AddData(car);

ClearTextBox(car);

DataGridREView(car);

MessageBox.Show("Запись добавлена", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

//Кнопка возврата к главному меню

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

{

GoBackToMainForm();

}

//Кнопка очистки полей ввода

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

{

//Подтверждение запроса

var result = new DialogResult();

result = MessageBox.Show("Вы действительно очистить поля ввода?", "Внимание!", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

//Очистка полей ввода, уведомление

ClearTextBox(car);

MessageBox.Show("Поля очищены", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

//Метод запрета ввода

private void FullNoInput(object sender, KeyPressEventArgs e)

{

char ch = e.KeyChar;

if (!Char.IsDigit(ch) && ch != 9)

{

e.Handled = true;

}

if (!Char.IsDigit(e.KeyChar)) return;

else

e.Handled = true;

}

//Метод запрета ввода текстовых символов

private void TextNoInput(object sender, KeyPressEventArgs e)

{

char ch = e.KeyChar;

if (!Char.IsDigit(ch) && ch != 9)

{

e.Handled = true;

}

}

//Метод запрета ввода чисел

private void NumberNoInput (object sender, KeyPressEventArgs e)

{

if (!Char.IsDigit(e.KeyChar)) return;

else

e.Handled = true;

}

//Метод проверки заполненности текстовых полей

public int TextCheck(string table)

{

//Елси контрол механик

if (table == "mechanic")

{

if (textBox1.Text == "")

{

MessageBox.Show("Заполните корректно поле Номер механика!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox2.Text == "")

{

MessageBox.Show("Заполните корректно поле Фамилия!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox3.Text == "")

{

MessageBox.Show("Заполните корректно поле Имя!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox4.Text == "")

{

MessageBox.Show("Заполните корректно поле Отчество!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox5.Text == "" || Convert.ToInt32(textBox5.Text) > 120 || textBox10.Text == "" || Convert.ToInt32(textBox10.Text) > 12 || textBox11.Text == "" || Convert.ToInt32(textBox11.Text) > 31)

{

MessageBox.Show("Заполните корректно поле Стаж!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (comboBox1.Text == "")

{

MessageBox.Show("Заполните поле Разряд!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

return 0;

}

//Если контрол машина

else if (table == "car")

{

if (textBox6.Text == "")

{

MessageBox.Show("Заполните корректно поле Номер авто!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox7.Text == "")

{

MessageBox.Show("Заполните корректно поле Марка!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox8.Text == "")

{

MessageBox.Show("Заполните корректно поле Модель!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox9.Text == "" || Convert.ToInt32(textBox9.Text) < 1805 || Convert.ToInt32(textBox9.Text) > 2019)

{

MessageBox.Show("Заполните корректно поле Год выпуска!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (comboBox2.Text == "")

{

MessageBox.Show("Заполните поле Тип кузова!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

return 0;

}

return 1;

}

}

}

Form3.cs – форма удаления и редактирования информации

// This is an open source non-commercial project. Dear PVS-Studio, please check it.

// PVS-Studio Static Code Analyzer for C, C++, C#, and Java: http://www.viva64.com

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.OleDb;

namespace Garage

{

//форма редактирования и удаления данных

public partial class Form3 : Form

{

//Инициализация переменных используемых в дальнейшем

string car = "car";

string mechanic = "mechanic";

string temp = "";

//Инициализация элементов и компонентов формы

public Form3()

{

InitializeComponent();

InitializeInterface();

DataGridREView(car);

DataGridREView(mechanic);

}

//Метод инициализации элементов формы

private void InitializeInterface()

{

//Название формы и контролов

this.Text = "Редактирование и удаление данных";

tabPage1.Text = "Механики";

tabPage2.Text = "Машины";

//Контрол механиков

button1.Text = "Назад";

button2.Text = "Редактировать";

button3.Text = "Очистить поля ввода";

button8.Text = "Удалить";

label1.Text = "Номер механика";

label2.Text = "Фамилия";

label3.Text = "Имя";

label4.Text = "Отчество";

label5.Text = "Стаж, года";

label6.Text = "Разряд";

label12.Text = "месяца";

label13.Text = "дни";

string[] ranks = { "1", "2", "3", "4", "5", "6" };

comboBox1.Items.AddRange(ranks);

//Контрол механиков

button6.Text = "Назад";

button5.Text = "Редактировать";

button4.Text = "Очистить поля ввода";

button7.Text = "Удалить";

label7.Text = "Номер авто";

label8.Text = "Марка";

label9.Text = "Модель";

label10.Text = "Год выпуска";

label11.Text = "Тип кузова";

string[] car\_types = { "Седан", "Хэтчбэк", "Универсал", "Лифтбэк", "Купе", "Кабриолет", "Родстер", "Тарга" };

comboBox2.Items.AddRange(car\_types);

//Отключение кнопок удаления и редактирования

button2.Enabled = false;

button8.Enabled = false;

button5.Enabled = false;

button7.Enabled = false;

//Получение количества записей

GetNoteCount(car);

GetNoteCount(mechanic);

}

//Метод счета количества записей

private void GetNoteCount(string table)

{

if (table == "mechanic")

{

try

{

//Подключение к бд

string con = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection oleDbConn = new OleDbConnection(con);

oleDbConn.Open();

//Отправление запроса на получение количества записей

OleDbCommand sql = new OleDbCommand("SELECT COUNT(\*) FROM " + table + ";", oleDbConn);

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

//Ввод количества записи в текстовый элемент на форме

label14.Text = "Кол-во записей: " + (int)sql.ExecuteScalar();

//Закрытие подключения

oleDbConn.Close();

}

catch (Exception ex)

{

//Уведомление об ошибке

MessageBox.Show(ex.ToString());

}

}

if (table == "car")

{

try

{

//Подключение к бд

string con = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection oleDbConn = new OleDbConnection(con);

oleDbConn.Open();

//Отправдение запроса на получение количества записей

OleDbCommand sql = new OleDbCommand("SELECT COUNT(\*) FROM " + table + ";", oleDbConn);

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

//Вывод количества записей в текстовый элемент на форму

label15.Text = "Кол-во записей: " + (int)sql.ExecuteScalar();

//Закрытие подключения

oleDbConn.Close();

}

catch (Exception ex)

{

//Уведомление об ошибке

MessageBox.Show(ex.ToString());

}

}

}

//Метод очистки полей ввода

private void ClearTextBox(string table)

{

//контрола с мехниками

if (table == "mechanic")

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

textBox10.Text = "";

textBox11.Text = "";

comboBox1.Text = "";

}

//контрола с машинами

else if (table == "car")

{

textBox6.Text = "";

textBox7.Text = "";

textBox8.Text = "";

textBox9.Text = "";

comboBox2.Text = "";

}

}

//Метод удаления данных

private void DeleteData(string table)

{

try

{

//Подключение к бд

string con = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection oleDbConn = new OleDbConnection(con);

DataTable dt = new DataTable();

oleDbConn.Open();

//Если удаление данных из контрола механиков

if (table == "mechanic")

{

//Отправление запроса на удаление записи

OleDbCommand sql = new OleDbCommand("DELETE FROM " + table + " WHERE mechanic\_id=" + temp + ";");

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

}

//Если удаление данных из контрола машин

else if (table == "car")

{

//Отправление запроса на удаление записи

OleDbCommand sql = new OleDbCommand("DELETE FROM " + table + " WHERE car\_id=" + temp + ";");

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

}

//Закрытие подключения

oleDbConn.Close();

}

catch (Exception ex)

{

//Уведомление об ошибке

MessageBox.Show(ex.ToString());

}

//Обновление количества записей

GetNoteCount(car);

GetNoteCount(mechanic);

}

//Преобразование стажа из текста в массив

private string[] GetExpArray(string exp)

{

string[] ExpArray = exp.Split(new Char[] { ' ' });

return ExpArray;

}

//Ввод данных в текстовые поля на форме и активация кнопок при нажатии на заглавную часть строки DataGridView

private void dataGridView1\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

try

{

temp = dataGridView1.SelectedCells[0].Value.ToString();

textBox1.Text = dataGridView1.SelectedCells[1].Value.ToString();

textBox2.Text = dataGridView1.SelectedCells[2].Value.ToString();

textBox3.Text = dataGridView1.SelectedCells[3].Value.ToString();

textBox4.Text = dataGridView1.SelectedCells[4].Value.ToString();

textBox5.Text = GetExpArray(dataGridView1.SelectedCells[5].Value.ToString())[0];

textBox10.Text = GetExpArray(dataGridView1.SelectedCells[5].Value.ToString())[1];

textBox11.Text = GetExpArray(dataGridView1.SelectedCells[5].Value.ToString())[2];

comboBox1.Text = dataGridView1.SelectedCells[6].Value.ToString();

//Активация кнопок удаления и редактирования

button2.Enabled = true;

button8.Enabled = true;

}

catch (Exception)

{

}

}

//Ввод данных в текстовые поля на форме и активация кнопок при нажатии на заглавную часть строки DataGridView

private void dataGridView2\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

try

{

temp = dataGridView2.SelectedCells[0].Value.ToString();

textBox6.Text = dataGridView2.SelectedCells[1].Value.ToString();

textBox7.Text = dataGridView2.SelectedCells[2].Value.ToString();

textBox8.Text = dataGridView2.SelectedCells[3].Value.ToString();

comboBox2.Text = dataGridView2.SelectedCells[4].Value.ToString();

textBox9.Text = dataGridView2.SelectedCells[5].Value.ToString();

//Активация кнопок удаления и редактирования

button5.Enabled = true;

button7.Enabled = true;

}

catch (Exception)

{

}

}

//Преобразование стажа в текст

private string GetMechanicExp()

{

string time = textBox5.Text + " " + textBox10.Text + " " + textBox11.Text;

return time;

}

//Обновление записи в бд

private void UpdateData(string table)

{

try

{

//Подключение к бд

string con = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection oleDbConn = new OleDbConnection(con);

DataTable dt = new DataTable();

oleDbConn.Open();

//Если контрол механиков

if (table == "mechanic")

{

//Запрос на обновление записи в бд

OleDbCommand sql = new OleDbCommand("UPDATE " + table + " SET mechanic\_number='" + textBox1.Text + "', mechanic\_surname='" + textBox2.Text + "', mechanic\_name='" + textBox3.Text + "' , mechanic\_patronymic='" + textBox4.Text + "' , mechanic\_exp= '" + GetMechanicExp() + "', mechanic\_rank='" + comboBox1.Text + "' WHERE mechanic\_id=" + temp + ";");

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

}

//Если контрол машин

else if (table == "car")

{

//Запрос на обновление записи в бд

OleDbCommand sql = new OleDbCommand("UPDATE " + table + " SET car\_number='" + textBox6.Text + "', car\_mark='" + textBox7.Text + "', car\_name='" + textBox8.Text + "' , car\_type='" + comboBox2.Text + "', car\_year=" + Convert.ToInt32(textBox9.Text) + " WHERE car\_id=" + temp + ";");

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

}

//Закрытие подключения

oleDbConn.Close();

}

catch (Exception ex)

{

//Уведомление об ошибке

MessageBox.Show(ex.ToString());

}

//Обновление счета записей на форме

GetNoteCount(car);

GetNoteCount(mechanic);

}

//Метод получения в DataGridView данных из бд

private void DataGridREView(string table)

{

try

{

//Подключение к бд

string con = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection oleDbConn = new OleDbConnection(con);

DataTable dt = new DataTable();

oleDbConn.Open();

//Отправление запроса на получение информации из бд

OleDbCommand sql = new OleDbCommand("SELECT \* FROM " + table + ";");

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(sql);

da.Fill(dt);

//Если контрол машин

if (table == "car")

{

dt.Columns["car\_id"].ColumnName = "id";

dt.Columns["car\_number"].ColumnName = "номер авто";

dt.Columns["car\_mark"].ColumnName = "марка";

dt.Columns["car\_name"].ColumnName = "модель";

dt.Columns["car\_type"].ColumnName = "тип";

dt.Columns["car\_year"].ColumnName = "год выпуска";

dataGridView2.DataSource = dt;

dataGridView2.Columns[0].Visible = false;

}

//Если контрол механиков

if (table == "mechanic")

{

dt.Columns["mechanic\_id"].ColumnName = "id";

dt.Columns["mechanic\_number"].ColumnName = "номер";

dt.Columns["mechanic\_surname"].ColumnName = "фамилия";

dt.Columns["mechanic\_name"].ColumnName = "имя";

dt.Columns["mechanic\_patronymic"].ColumnName = "отчество";

dt.Columns["mechanic\_exp"].ColumnName = "стаж";

dt.Columns["mechanic\_rank"].ColumnName = "разряд";

dataGridView1.DataSource = dt;

dataGridView1.Columns[0].Visible = false;

}

//Закрытие подключения

oleDbConn.Close();

}

catch (Exception ex)

{

//Уведомление об ошибке

MessageBox.Show(ex.ToString());

}

}

//Метод возврата на главную форму

private void GoBackToMainForm()

{

this.Close();

Form MainForm = new Form1();

MainForm.Show();

}

//Кнопку возврата на главную форму

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

{

GoBackToMainForm();

}

//Кнопка возврата на главную форму

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

{

GoBackToMainForm();

}

//Кнопка очистки полей ввода

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

{

//Подтверждение запроса

var result = new DialogResult();

result = MessageBox.Show("Вы действительно очистить поля ввода?", "Внимание!", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

//Очистка полей ввода и уведомление об этом

ClearTextBox(mechanic);

MessageBox.Show("Поля очищены", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

//Кнопка очистки полей ввода

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

{

//Подтверждение запроса

var result = new DialogResult();

result = MessageBox.Show("Вы действительно очистить поля ввода?", "Внимание!", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

//Очистка полей ввода и уведомление об этом

ClearTextBox(car);

MessageBox.Show("Поля очищены", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

//Кнопка изменения записи

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

{

//Проверка заполнености полей ввода

var result = TextCheck(mechanic);

if (result == 1)

{

return;

}

else if (result == 0)

{

//Изменение записи, очистка полей ввода, обновление DataGridView, уведомление

UpdateData(mechanic);

ClearTextBox(mechanic);

DataGridREView(mechanic);

MessageBox.Show("Запись изменена", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

//Деактивация кнопок

button2.Enabled = false;

button8.Enabled = false;

}

//Кнопка удаления записи

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

{

//Подтверждение удаления записи

var result = new DialogResult();

result = MessageBox.Show("Вы действительно желаете удалить запись?", "Внимание!", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

//Удаление записи, очистка полей ввода, обновление DataGridView,уведомление

DeleteData(mechanic);

ClearTextBox(mechanic);

DataGridREView(mechanic);

MessageBox.Show("Запись удалена", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

//Деактивация кнопок

button2.Enabled = false;

button8.Enabled = false;

}

//Кнопка изменения записи

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

{

//Проверка заполненности полей

var result = TextCheck(car);

if (result == 1)

{

return;

}

else if (result == 0)

{

//Изменение записи, очистка полей ввода, обновление DataGridView, уведомление

UpdateData(car);

ClearTextBox(car);

DataGridREView(car);

MessageBox.Show("Запись изменена", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

//Деактивация кнопок

button5.Enabled = false;

button7.Enabled = false;

}

//Кнопка удаления записей

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

{

//Подтверждение запроса

var result = new DialogResult();

result = MessageBox.Show("Вы действительно желаете удалить запись?", "Внимание!", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

//Удаление записи, очистка полей ввода, обновление DataGridView, уведомление

DeleteData(car);

ClearTextBox(car);

DataGridREView(car);

MessageBox.Show("Запись удалена", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

//Деактивация кнопок

button5.Enabled = false;

button7.Enabled = false;

}

//Метод запрета ввода в текстовые поля

private void FullNoInput(object sender, KeyPressEventArgs e)

{

char ch = e.KeyChar;

if (!Char.IsDigit(ch) && ch != 9)

{

e.Handled = true;

}

if (!Char.IsDigit(e.KeyChar)) return;

else

e.Handled = true;

}

//Метод запрета ввода текста в поле ввода

private void TextNoInput(object sender, KeyPressEventArgs e)

{

char ch = e.KeyChar;

if (!Char.IsDigit(ch) && ch != 9)

{

e.Handled = true;

}

}

//Метод запрета ввода чисел в поле ввода

private void NumberNoInput(object sender, KeyPressEventArgs e)

{

if (!Char.IsDigit(e.KeyChar)) return;

else

e.Handled = true;

}

//Проверка заполнености полей ввода

public int TextCheck(string table)

{

//Если контрол механики

if (table == "mechanic")

{

if (textBox1.Text == "")

{

MessageBox.Show("Заполните корректно поле Номер механика!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox2.Text == "")

{

MessageBox.Show("Заполните корректно поле Фамилия!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox3.Text == "")

{

MessageBox.Show("Заполните корректно поле Имя!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox4.Text == "")

{

MessageBox.Show("Заполните корректно поле Отчество!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox5.Text == "" || Convert.ToInt32(textBox5.Text) > 120 || textBox10.Text == "" || Convert.ToInt32(textBox10.Text) > 12 || textBox11.Text == "" || Convert.ToInt32(textBox11.Text) > 31)

{

MessageBox.Show("Заполните корректно поле Стаж!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (comboBox1.Text == "")

{

MessageBox.Show("Заполните поле Разряд!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

return 0;

}

//Если контрол машины

else if (table == "car")

{

if (textBox6.Text == "")

{

MessageBox.Show("Заполните корректно поле Номер авто!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox7.Text == "")

{

MessageBox.Show("Заполните корректно поле Марка!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox8.Text == "")

{

MessageBox.Show("Заполните корректно поле Модель!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox9.Text == "" || Convert.ToInt32(textBox9.Text) < 1805 || Convert.ToInt32(textBox9.Text) > 2019)

{

MessageBox.Show("Заполните корректно поле Год выпуска!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (comboBox2.Text == "")

{

MessageBox.Show("Заполните поле Тип кузова!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

return 0;

}

return 1;

}

}

}

Form4.cs – Форма составления информации

// This is an open source non-commercial project. Dear PVS-Studio, please check it.

// PVS-Studio Static Code Analyzer for C, C++, C#, and Java: http://www.viva64.com

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.OleDb;

namespace Garage

{

public partial class Form4 : Form

{

string repair = "repair";

string repair\_date = "";

public Form4()

{

InitializeComponent();

InitializeInterface();

comboboxInput();

}

private void InitializeInterface()

{

this.Text = "Составление заказа";

button1.Text = "Назад";

button2.Text = "Добавить";

button3.Text = "Очистить поля ввода";

label1.Text = "Выберите механика";

label2.Text = "Выберите авто";

label3.Text = "Выберите дату работы";

label4.Text = "Введите стоимость";

label5.Text = "Введите время ремонта";

label6.Text = "дни";

label7.Text = "часы";

label8.Text = "минуты";

}

private void ClearTextBox(string table)

{

if (table == "repair")

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

comboBox1.Text = "";

comboBox2.Text = "";

dateTimePicker1.Text = "";

}

}

private void AddData(string table)

{

try

{

string A = comboBox1.Text;

string[] a = A.Split(new Char[] { ' ' });

string B = comboBox2.Text;

string[] b = B.Split(new Char[] { ' ' });

int days = Convert.ToInt32(textBox2.Text);

int hours = Convert.ToInt32(textBox3.Text);

int minuts = Convert.ToInt32(textBox4.Text);

string repair\_time = GetRepairTime(days, hours, minuts);

string repair\_cost = textBox1.Text;

int id = 0;

Random rnd = new Random();

id = rnd.Next(1, 999999);

DateTime SelectedDate = dateTimePicker1.Value;

var DateRepair = SelectedDate;

repair\_date = DateRepair.ToString("dd/mm/yy");

string con1 = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection OleDbcon1 = new OleDbConnection(con1);

OleDbcon1.Open();

OleDbCommand sql1 = new OleDbCommand("INSERT INTO " + table

+ " (repair\_id, mechanic\_id, car\_id, repair\_date, repair\_time, repair\_cost) VALUES (" +

Convert.ToInt32(id) + "," + Convert.ToInt32(a[0]) + "," + Convert.ToInt32(b[0]) + ",'" + repair\_date + "','" + repair\_time + "'," +

Convert.ToInt32(repair\_cost) + ");", OleDbcon1);

sql1.ExecuteNonQuery();

OleDbcon1.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

private void comboboxInput()

{

try

{

string con1 = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection OleDbcon1 = new OleDbConnection(con1);

OleDbcon1.Open();

OleDbCommand sql1 = new OleDbCommand("SELECT mechanic\_id AS [id], mechanic\_surname AS [фамилия], mechanic\_name AS [имя], mechanic\_patronymic AS [отчество] FROM mechanic;", OleDbcon1);

sql1.ExecuteNonQuery();

OleDbDataReader reader = sql1.ExecuteReader();

while (reader.Read())

{

comboBox1.Items.Add(reader["id"] + " " + reader["фамилия"] + " " + reader["имя"].ToString() + " " + reader["отчество"].ToString());

}

reader.Close();

OleDbcon1.Close();

string con2 = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection OleDbcon2 = new OleDbConnection(con2);

OleDbcon2.Open();

OleDbCommand sql2 = new OleDbCommand("SELECT car\_id AS [id], car\_number AS [номер], car\_mark AS [марка], car\_name AS [модель] FROM car;", OleDbcon2);

sql2.ExecuteNonQuery();

OleDbDataReader reader2 = sql2.ExecuteReader();

while (reader2.Read())

{

comboBox2.Items.Add(reader2["id"] + " " + reader2["номер"] + " " + reader2["марка"].ToString() + " " + reader2["модель"].ToString());

}

reader2.Close();

OleDbcon2.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

private string GetRepairTime(int days, int hours, int minuts)

{

string minut\_time = days.ToString() + "д" + hours.ToString() + "ч" + minuts.ToString() + "м";

return minut\_time;

}

private void GoBackToMainForm()

{

this.Close();

Form MainForm = new Form1();

MainForm.Show();

}

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

{

GoBackToMainForm();

}

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

{

var result = TextCheck(repair);

if (result == 1)

{

return;

}

else if (result == 0)

{

AddData(repair);

ClearTextBox(repair);

MessageBox.Show("Запись добавлена", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

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

{

var result = new DialogResult();

result = MessageBox.Show("Вы действительно очистить поля ввода?", "Внимание!", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

ClearTextBox(repair);

MessageBox.Show("Поля очищены", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

private void FullNoInput(object sender, KeyPressEventArgs e)

{

char ch = e.KeyChar;

if (!Char.IsDigit(ch) && ch != 9)

{

e.Handled = true;

}

if (!Char.IsDigit(e.KeyChar)) return;

else

e.Handled = true;

}

private void TextNoInput(object sender, KeyPressEventArgs e)

{

char ch = e.KeyChar;

if (!Char.IsDigit(ch) && ch != 9)

{

e.Handled = true;

}

}

private void NumberNoInput(object sender, KeyPressEventArgs e)

{

if (!Char.IsDigit(e.KeyChar)) return;

else

e.Handled = true;

}

public int TextCheck(string table)

{

if (table == "repair")

{

if (comboBox1.Text == "")

{

MessageBox.Show("Заполните поле Выберите механика!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (comboBox2.Text == "")

{

MessageBox.Show("Заполните поле Выберите авто!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (dateTimePicker1.Text == "")

{

MessageBox.Show("Заполните корректно поле Выберите дату работы!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox1.Text == "")

{

MessageBox.Show("Заполните корректно поле Введите стоимость!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

if (textBox2.Text == "" || textBox3.Text == "" || textBox4.Text == "" || Convert.ToInt32(textBox2.Text) > 31 || Convert.ToInt32(textBox3.Text) > 24 || Convert.ToInt32(textBox4.Text) > 60)

{

MessageBox.Show("Заполните корректно поля Введите время ремонта!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return 1;

}

return 0;

}

return 1;

}

}

}

Form5.cs – Форма просмотра и удаления заказов

// This is an open source non-commercial project. Dear PVS-Studio, please check it.

// PVS-Studio Static Code Analyzer for C, C++, C#, and Java: http://www.viva64.com

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.OleDb;

using Word = Microsoft.Office.Interop.Word;

using System.IO;

namespace Garage

{

//Форма вывода в word и удаления заказов

public partial class Form5 : Form

{

private readonly string FileName = Directory.GetCurrentDirectory() + @"\\template.docx";

//Инициализация переменных, используемых в дальнейшем

string repair = "repair";

string temp = "";

//Инициализация компонентов формы и интерфейса

public Form5()

{

InitializeComponent();

InitializeInterface();

DataGridREView(repair);

GetNoteCount(repair);

}

//Метод инициализации интерфейса

private void InitializeInterface()

{

this.Text = "Просмотр и удаление заказов";

button1.Text = "Назад";

button2.Text = "Удалить";

button3.Text = "";

button3.Text = "Отчет";

label1.Text = "Кол-во записей: ";

button2.Enabled = false;

button3.Enabled = false;

}

//Метод получения количество записей

private void GetNoteCount(string table)

{

try

{

string con = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection oleDbConn = new OleDbConnection(con);

oleDbConn.Open();

OleDbCommand sql = new OleDbCommand("SELECT COUNT(\*) FROM " + table + ";", oleDbConn);

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

label1.Text = "Кол-во записей: " + (int)sql.ExecuteScalar();

//Закрытие подключения

oleDbConn.Close();

}

catch (Exception ex)

{

//Сообщение об ошибке

MessageBox.Show(ex.ToString());

}

}

//Метод получения данных для DataGridView

private void DataGridREView(string table)

{

try

{

//Подключение к бд

string con = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection oleDbConn = new OleDbConnection(con);

DataTable dt = new DataTable();

oleDbConn.Open();

//Отправление запроса

OleDbCommand sql = new OleDbCommand("SELECT repair\_id AS [id], (SELECT mechanic\_surname FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [фамилия мастера], (SELECT mechanic\_name FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [имя мастера], (SELECT mechanic\_patronymic FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [отчество], (SELECT car\_name FROM car WHERE repair.car\_id = car.car\_id) AS [модель авто], (SELECT car\_mark FROM car WHERE repair.car\_id = car.car\_id) AS [марка авто], repair\_date AS [дата работы], repair\_cost AS [цена] FROM " + table + ";", oleDbConn);

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

//Заполнение полей DataGridView

OleDbDataAdapter da = new OleDbDataAdapter(sql);

da.Fill(dt);

dataGridView1.DataSource = dt;

dataGridView1.Columns[0].Visible = false;

//Закрытие подключения

oleDbConn.Close();

}

catch (Exception ex)

{

//Сообщение об ошибке

MessageBox.Show(ex.ToString());

}

}

//Кнопка перехода к главному меню

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

{

this.Hide();

Form MainForm = new Form1();

MainForm.Show();

}

//Кнопка удаления данных

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

{

//Получение согласия на удаление данных

var result = new DialogResult();

result = MessageBox.Show("Вы действительно желаете удалить запись?", "Внимание!", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

//Удаление данных, обновление DataGridView и счетчика записей в таблице

DeleteData(repair);

DataGridREView(repair);

GetNoteCount(repair);

MessageBox.Show("Запись удалена", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

button2.Enabled = false;

button3.Enabled = false;

}

//Метод получение id записи и активация кнопок

private void dataGridView1\_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

//Активация кнопок

button2.Enabled = true;

button3.Enabled = true;

try

{

//Получение id

temp = dataGridView1.SelectedCells[0].Value.ToString();

}

catch (Exception)

{

}

}

//Метод удаления записи из таблицы

private void DeleteData(string table)

{

try

{

//Подключение к бд

string con = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection oleDbConn = new OleDbConnection(con);

DataTable dt = new DataTable();

oleDbConn.Open();

//Отправление запроса на удаление записи

OleDbCommand sql = new OleDbCommand("DELETE FROM " + table + " WHERE repair\_id=" + Convert.ToInt32(temp) + ";");

sql.Connection = oleDbConn;

sql.ExecuteNonQuery();

oleDbConn.Close();

}

catch (Exception ex)

{

//Уведомление об ошибке

MessageBox.Show(ex.ToString());

}

}

//Медот вывода данных в word

private void WordExport()

{

//Подключение к бд

string con1 = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=Garage.mdb;";

OleDbConnection OleDbcon1 = new OleDbConnection(con1);

OleDbcon1.Open();

//Отправка запроса на получение данных

OleDbCommand sql1 = new OleDbCommand("SELECT repair\_id AS [id], (SELECT mechanic\_surname FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [фамилия мастера], (SELECT mechanic\_name FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [имя мастера], (SELECT mechanic\_patronymic FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [отчество], (SELECT car\_name FROM car WHERE repair.car\_id = car.car\_id) AS [модель авто], (SELECT car\_mark FROM car WHERE repair.car\_id = car.car\_id) AS [марка авто], repair\_date AS [дата работы], repair\_cost AS [цена] FROM repair WHERE repair\_id = " + temp +";", OleDbcon1);

sql1.ExecuteNonQuery();

OleDbDataReader reader = sql1.ExecuteReader();

//Инициализация элемента word и его визуализация

var wordApp = new Word.Application();

wordApp.Visible = false;

var wordDocument = wordApp.Documents.Open(FileName);

while (reader.Read())

{

//Ввод полученной из бд информации в word

wordDocument.Bookmarks["id"].Range.Text = reader["id"].ToString();

wordDocument.Bookmarks["surname"].Range.Text = reader["фамилия мастера"].ToString();

wordDocument.Bookmarks["name"].Range.Text = reader["имя мастера"].ToString();

wordDocument.Bookmarks["patronimyc"].Range.Text = reader["отчество"].ToString();

wordDocument.Bookmarks["model"].Range.Text = reader["модель авто"].ToString();

wordDocument.Bookmarks["mark"].Range.Text = reader["марка авто"].ToString();

wordDocument.Bookmarks["date"].Range.Text = reader["дата работы"].ToString();

wordDocument.Bookmarks["cost"].Range.Text = reader["цена"].ToString();

}

//Просмотр документа

wordApp.Visible = true;

//Прекращение ввода

reader.Close();

//Закрытие подключения

OleDbcon1.Close();

}

//Кнопка вывода в word

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

{

try

{

WordExport();

//Деактивация кнопок

button2.Enabled = false;

button3.Enabled = false;

}

catch (Exception ex)

{

//Уведомление об ошибке

MessageBox.Show(ex.ToString());

}

}

}

}

Form6.cs – Форма экспорта данных excel.

// This is an open source non-commercial project. Dear PVS-Studio, please check it.

// PVS-Studio Static Code Analyzer for C, C++, C#, and Java: http://www.viva64.com

using Microsoft.Office.Interop.Excel;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Garage

{

public partial class Form6 : Form

{

public Form6()

{

InitializeComponent();

InitializeInterface();

}

private void InitializeInterface()

{

this.Text = "Экспорт данных в Excel";

button1.Text = "Назад";

button2.Text = "Отчет";

}

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

{

GoToMainForm();

}

private void GoToMainForm()

{

this.Hide();

Form MainForm = new Form1();

}

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

{

excel\_output();

}

private void excel\_output()

{

string connectString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=Garage.mdb;";

OleDbConnection myConnection = new OleDbConnection(connectString);

myConnection.Open();

Microsoft.Office.Interop.Excel.Application ExcelApp = new Microsoft.Office.Interop.Excel.Application();

ExcelApp.Workbooks.Add(Type.Missing);

////////////////////////////////////////////////////////////////////////////

var car = (Microsoft.Office.Interop.Excel.Worksheet)ExcelApp.Sheets[1];

car.Name = "car";

var cellsD = (Microsoft.Office.Interop.Excel.Range)car.Cells;

cellsD.HorizontalAlignment = Microsoft.Office.Interop.Excel.Constants.xlCenter;

car.Cells[1, 1] = "id";

car.Cells[1, 2] = "Номер машины";

car.Cells[1, 3] = "Модель машины";

car.Cells[1, 4] = "Марка машины";

car.Cells[1, 5] = "Тип машины";

car.Cells[1, 6] = "Год выпуска";

string queryD = "SELECT \* FROM car";

OleDbCommand commandD = new OleDbCommand(queryD, myConnection);

OleDbDataReader readerD = commandD.ExecuteReader();

int temp = 2;

while (readerD.Read())

{

car.Cells[temp, 1] = readerD[0].ToString();

car.Cells[temp, 2] = readerD[1].ToString();

car.Cells[temp, 3] = readerD[2].ToString();

car.Cells[temp, 4] = readerD[3].ToString();

car.Cells[temp, 5] = readerD[4].ToString();

car.Cells[temp, 6] = readerD[5].ToString();

temp++;

}

ExcelApp.Visible = true;

readerD.Close();

car.Columns.AutoFit();

car.Rows.AutoFit();

///////////////////////////////////////////////////////////////////////////

}

}

}

Form7.cs – Форма предоставления информации о разработчике

// This is an open source non-commercial project. Dear PVS-Studio, please check it.

// PVS-Studio Static Code Analyzer for C, C++, C#, and Java: http://www.viva64.com

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Garage

{

public partial class Form7 : Form

{

//Форма представления информации об разработчике

//Готова

public Form7()

{

InitializeComponent();

this.Text = "Информация о разработчике";

label1.Text = "Работу выполнил ученик группы ПС-15\n\nХаритонов Илья\n\n2019 год";

button1.Text = "Назад";

}

private void GoBackToMainForm()

{

this.Close();

Form MainForm = new Form1();

}

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

{

GoBackToMainForm();

}

}

}