**Содержание**

1. Постановка задачи 3

2. Структура таблиц в БД 4

3. Схема базы данных 5

4. Описание программных модулей 6

5. Листинг программы 9

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 о автомобилях

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Номер** | **Марка** | **Название** | **Тип кузова** | **Год выпуска** |
| 3 | Ford | Focus | седан | 2011 |
| 4 | УАЗ | Patriot | универсал | 2014 |
| 5 | Opel | Astra | хетчбэк | 2015 |

**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 «mechanic»

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название поля** | **Тип данных** | **Описание поля** |
| 1 | mechanic\_id | integer | Идентификатор поля. Первичный ключ |
| 2 | mechanic\_number | integer | Табельный номер механика |
| 3 | mechanic\_surname | string(255) | Фамилия механика |
| 4 | mechanic\_name | string(255) | Имя механика |
| 5 | mechanic\_patronymic | string(255) | Отчество механика |
| 6 | mechanic\_exp | integer | Стаж механика |
| 7 | mechanic\_rank | integer | Разряд механика |

Таблица 2.2 - Описание таблицы «car»

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название поля** | **Тип данных** | **Описание поля** |
| 1 | car\_id | integer | Идентификатор поля. Первичный ключ |
| 2 | car\_number | integer | Табельный номер машины |
| 3 | car\_mark | string(255) | Марка машины |
| 4 | car\_name | string(255) | Модель машины |
| 5 | car\_type | string(255) | Тип кузова |
| 6 | car\_year | integer | Год выпуска |

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 | Идентификатор механика |
| 3 | car\_id | integer | Идентификатор машины |
| 4 | repair\_date | string(255) | Дата начала работ |
| 5 | repair\_time | string(255) | Время затраченное на работу |
| 6 | repair\_cost | string(255) | Стоимость работы |

1. **Схема базы данных**

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

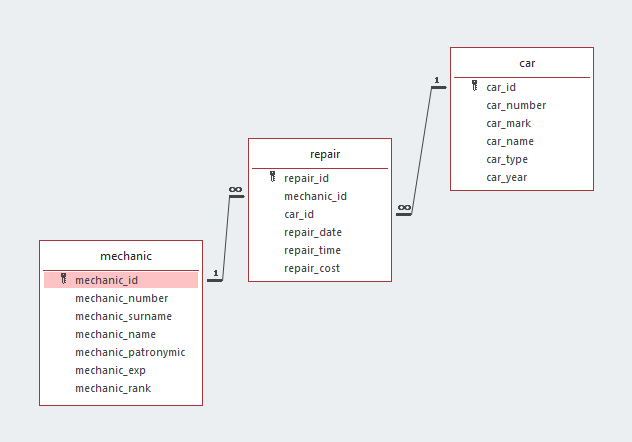


Рисунок 3.1 – Схема базы данных

1. **Описание программных модулей**

Форма меню программы представлена на рисунке 4.1.

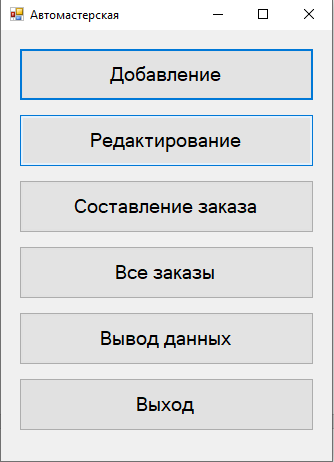


Рисунок 4.1 – Форма меню программы

Форма добавления данных в таблицы представлена на рисунке 4.2.

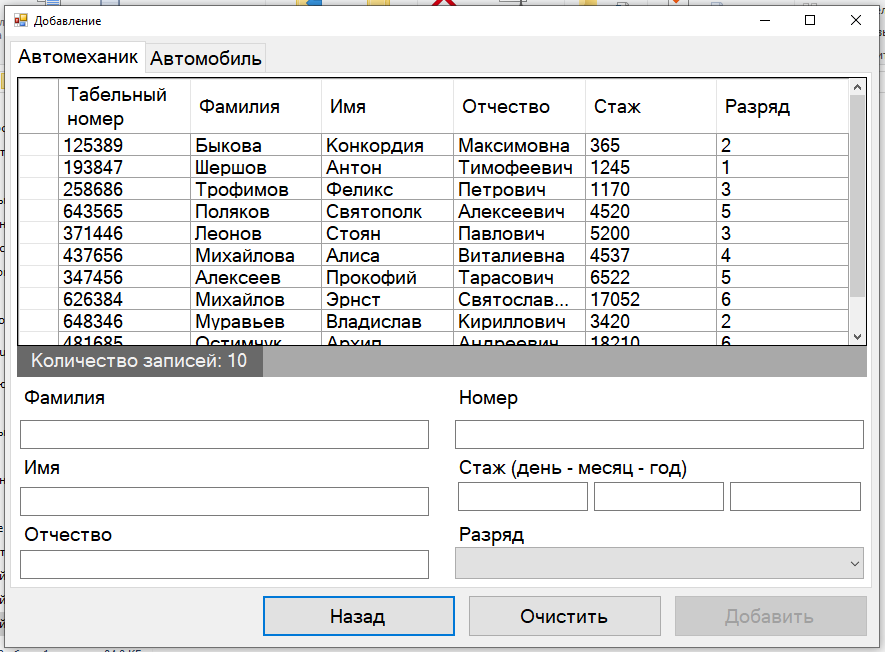


Рисунок 4.2 – Форма добавления информации в базу данных

Форма редактирования данных таблицы представлена на рисунке 4.3.

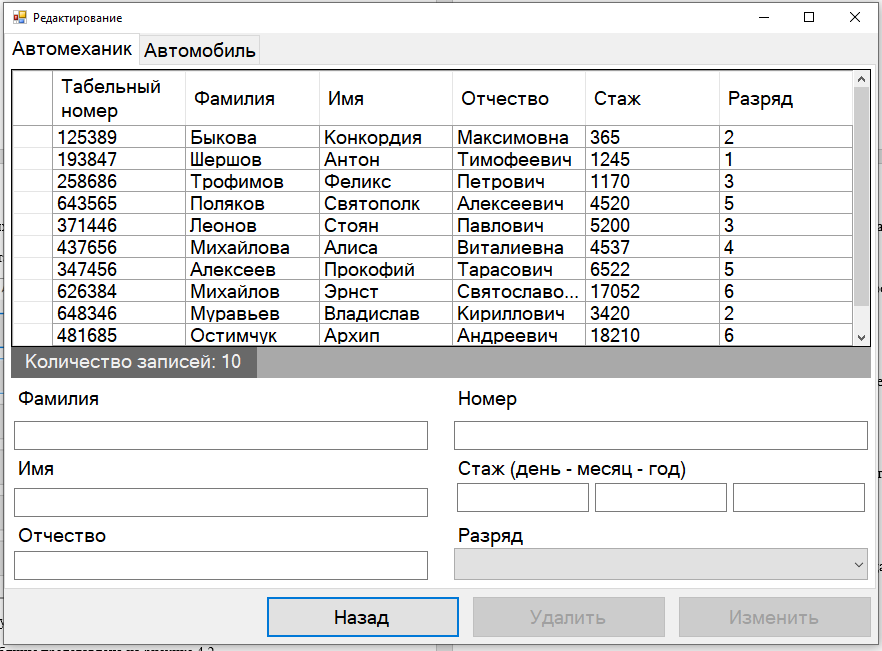


Рисунок 4.3 – Форма редактирования информации в базе данных

Форма составления заказа представлена на рисунке 4.4.

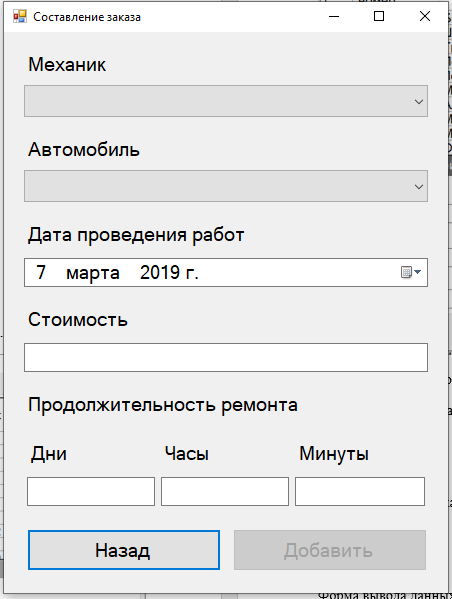


Рисунок 4.4 – Форма составления заказа

Форма просмотра заказов представлена на рисунке 4.5.

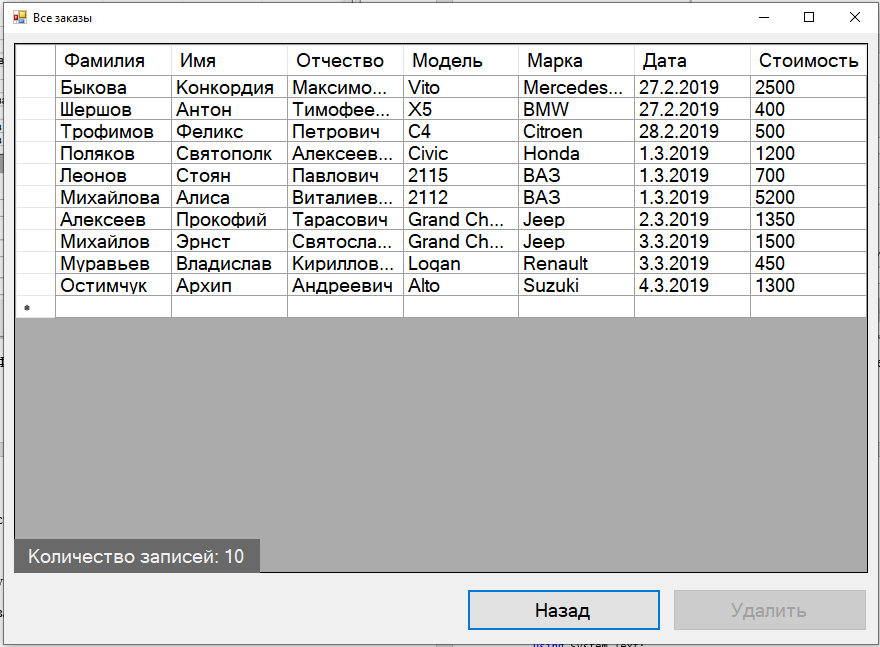


Рисунок 4.5 – Форма просмотра заказов

Форма вывода данных представлена на рисунке 4.6.

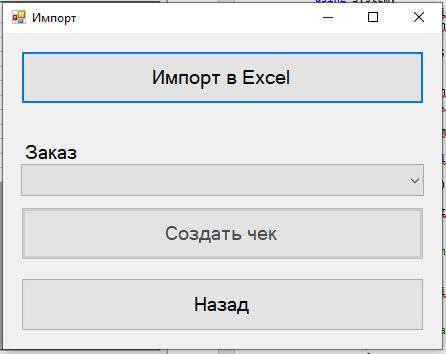


Рисунок 4.6 – Форма вывода данных

1. **Листинг программы**

**menu.cs** – форма меню

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 carShowroom

{

public partial class menu : Form

{

public menu()

{

InitializeComponent();

}

// Закрытие программы

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

{

Application.Exit();

}

// Переход на форму добавления

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

{

add add = new add();

add.Show();

Hide();

}

// Полное закрытие программы

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

{

Environment.Exit(0);

}

// Переход на форму редактирования

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

{

edit edit = new edit();

edit.Show();

Hide();

}

// Переход на форму добавления заказа

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

{

addOrder addOrder = new addOrder();

addOrder.Show();

Hide();

}

// Переход на форму просмотра заказов

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

{

allOrder allOrder = new allOrder();

allOrder.Show();

Hide();

}

// Переход на форму импорта данных

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

{

import import = new import();

import.Show();

Hide();

}

}

}

**add.cs** – форма добавления

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.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace carShowroom

{

public partial class add : Form

{

public add()

{

InitializeComponent();

}

// Полное закрытие программы

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

{

Environment.Exit(0);

}

// Кнопка назад

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

{

menu menu = new menu();

menu.Show();

Hide();

}

// Подключение класса с функциями

Function MainFunc = new Function();

// Загрузка формы добавления

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

{

updateTables();

}

// Обновление таблиц

private void updateTables()

{

try

{

// Таблица механиков

OleDbDataAdapter data1 = new OleDbDataAdapter(MainFunc.getAll("mechanic"));

DataTable table1 = new DataTable();

data1.Fill(table1);

table1.Columns["mechanic\_id"].ColumnName = "ID";

table1.Columns["mechanic\_number"].ColumnName = "Табельный номер";

table1.Columns["mechanic\_surname"].ColumnName = "Фамилия";

table1.Columns["mechanic\_name"].ColumnName = "Имя";

table1.Columns["mechanic\_patronymic"].ColumnName = "Отчество";

table1.Columns["mechanic\_exp"].ColumnName = "Стаж";

table1.Columns["mechanic\_rank"].ColumnName = "Разряд";

dataGridView1.DataSource = table1;

dataGridView1.Columns[0].Visible = false;

label11.Text = "Количество записей: " + table1.Rows.Count.ToString();

// Таблица авто

OleDbDataAdapter data2 = new OleDbDataAdapter(MainFunc.getAll("car"));

DataTable table2 = new DataTable();

data2.Fill(table2);

table2.Columns["car\_id"].ColumnName = "ID";

table2.Columns["car\_number"].ColumnName = "Табельный номер";

table2.Columns["car\_mark"].ColumnName = "Марка";

table2.Columns["car\_name"].ColumnName = "Модель";

table2.Columns["car\_type"].ColumnName = "Тип кузова";

table2.Columns["car\_year"].ColumnName = "Год";

dataGridView2.DataSource = table2;

dataGridView2.Columns[0].Visible = false;

label13.Text = "Количество записей: " + table2.Rows.Count.ToString();

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

// Проверка полей при изменении

private void textBox4\_KeyUp(object sender, KeyEventArgs e) { checkInputMechanic(); }

private void comboBox1\_TextChanged(object sender, EventArgs e) { checkInputMechanic(); }

private void textBox9\_KeyUp(object sender, KeyEventArgs e) { checkInputCar(); }

private void comboBox2\_TextChanged(object sender, EventArgs e) { checkInputCar(); }

// Проверка полей механика

private bool checkInputMechanic()

{

try

{

// Проверка есть ли пустые поля

if (textBox1.Text != "" && textBox2.Text != "" && textBox4.Text != "" && textBox5.Text != "" && textBox6.Text != "" && comboBox1.Text != "")

{

// Проверка полей на правильность ввода

if (MainFunc.stringTest(textBox1.Text, @"^[a-zA-Zа-яА-Я]\*$") && MainFunc.stringTest(textBox2.Text, @"^[a-zA-Zа-яА-Я]\*$") && MainFunc.stringTest(textBox6.Text, @"^[a-zA-Zа-яА-Я]\*$") && MainFunc.stringTest(textBox4.Text, @"^[0-9]\*$") && MainFunc.stringTest(textBox3.Text, @"^[0-9]\*$") && MainFunc.stringTest(textBox5.Text, @"^[0-9]\*$") && MainFunc.stringTest(textBox11.Text, @"^[0-9]\*$"))

{

button1.Enabled = true;

return true;

}

else

{

button1.Enabled = false;

}

}

else

{

button1.Enabled = false;

}

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return false;

}

// Отчистка полей механика

private void clearInputMechanic()

{

textBox4.Text = "";

textBox1.Text = "";

textBox2.Text = "";

textBox6.Text = "";

textBox3.Text = "";

textBox5.Text = "";

textBox11.Text = "";

}

// Добавление механика

private void addMechanic()

{

try

{

Random rnd = new Random();

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

if (checkInputMechanic())

{

int exp = Convert.ToInt32(textBox3.Text) + Convert.ToInt32(textBox5.Text) \* 30 + Convert.ToInt32(textBox11.Text) \* 365;

// Добавление записи

MainFunc.sql("INSERT INTO mechanic VALUES (" +

temp + ", " +

textBox4.Text + ", " +

"'" + textBox1.Text + "', " +

"'" + textBox2.Text + "', " +

"'" + textBox6.Text + "', " +

"'" + exp + "', " +

comboBox1.Text +

");");

clearInputMechanic();

// Обновление таблиц

updateTables();

MessageBox.Show(this, "Запись добавлена!", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

// Проверка полей автомобиля

private bool checkInputCar()

{

try

{

// Проверка есть ли пустые поля

if (textBox7.Text != "" && textBox8.Text != "" && textBox9.Text != "" && textBox10.Text != "" && comboBox2.Text != "")

{

if (MainFunc.stringTest(textBox9.Text, @"^[a-zA-Zа-яА-Я\-]\*$") && MainFunc.stringTest(textBox7.Text, @"^[0-9]\*$") && MainFunc.stringTest(textBox8.Text, @"^[0-9]\*$"))

{

button1.Enabled = true;

return true;

}

else

{

button1.Enabled = false;

}

}

else

{

button1.Enabled = false;

}

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return false;

}

// Отчистка полей машины

private void clearInputCar()

{

textBox8.Text = "";

textBox9.Text = "";

textBox10.Text = "";

textBox7.Text = "";

}

// Добавление машины

private void addCar()

{

try

{

Random rnd = new Random();

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

if (checkInputCar())

{

// Добавление записи

MainFunc.sql("INSERT INTO car VALUES (" +

temp + ", " +

textBox8.Text + ", " +

"'" + textBox9.Text + "', " +

"'" + textBox10.Text + "', " +

"'" + comboBox2.Text + "', " +

textBox7.Text +

");");

clearInputCar();

// Обновление таблиц

updateTables();

MessageBox.Show(this, "Запись добавлена!", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

// Кнопка добавления

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

{

// Определяем что добавляем

if (tabControl1.SelectedIndex == 0)

{

// Добавляем механика

addMechanic();

}

else if (tabControl1.SelectedIndex == 1)

{

// Добавляем машину

addCar();

}

}

private void tabControl1\_Selected(object sender, TabControlEventArgs e)

{

if (tabControl1.SelectedIndex == 0) { checkInputMechanic(); }

else if (tabControl1.SelectedIndex == 1) { checkInputCar(); }

}

}

}

**edit.cs** – форма редактирования

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 carShowroom

{

public partial class edit : Form

{

public edit()

{

InitializeComponent();

}

// ID выбранной записи

private string cur\_id = "";

// Полное закрытие программы

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

{

Environment.Exit(0);

}

// Открыть форму меню

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

{

menu menu = new menu();

menu.Show();

Hide();

}

// Подключение класса с функциями

Function MainFunc = new Function();

// Загрузка формы

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

{

updateTables();

}

// Обновление таблиц

private void updateTables()

{

try

{

// Таблица механиков

OleDbDataAdapter data1 = new OleDbDataAdapter(MainFunc.getAll("mechanic"));

DataTable table1 = new DataTable();

data1.Fill(table1);

table1.Columns["mechanic\_id"].ColumnName = "ID";

table1.Columns["mechanic\_number"].ColumnName = "Табельный номер";

table1.Columns["mechanic\_surname"].ColumnName = "Фамилия";

table1.Columns["mechanic\_name"].ColumnName = "Имя";

table1.Columns["mechanic\_patronymic"].ColumnName = "Отчество";

table1.Columns["mechanic\_exp"].ColumnName = "Стаж";

table1.Columns["mechanic\_rank"].ColumnName = "Разряд";

dataGridView1.DataSource = table1;

dataGridView1.Columns[0].Visible = false;

label11.Text = "Количество записей: " + table1.Rows.Count.ToString();

// Таблица авто

OleDbDataAdapter data2 = new OleDbDataAdapter(MainFunc.getAll("car"));

DataTable table2 = new DataTable();

data2.Fill(table2);

table2.Columns["car\_id"].ColumnName = "ID";

table2.Columns["car\_number"].ColumnName = "Табельный номер";

table2.Columns["car\_mark"].ColumnName = "Марка";

table2.Columns["car\_name"].ColumnName = "Модель";

table2.Columns["car\_type"].ColumnName = "Тип кузова";

table2.Columns["car\_year"].ColumnName = "Год";

dataGridView2.DataSource = table2;

dataGridView2.Columns[0].Visible = false;

label13.Text = "Количество записей: " + table2.Rows.Count.ToString();

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

// Выбор записи в таблице автомехаников

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

{

// Получение ID записи

cur\_id = dataGridView1.SelectedCells[0].Value.ToString();

// Подстановка данных в поля

textBox4.Text = dataGridView1.SelectedCells[1].Value.ToString();

textBox1.Text = dataGridView1.SelectedCells[2].Value.ToString();

textBox2.Text = dataGridView1.SelectedCells[3].Value.ToString();

textBox6.Text = dataGridView1.SelectedCells[4].Value.ToString();

int exp = Convert.ToInt32(dataGridView1.SelectedCells[5].Value.ToString());

int exp\_year = Convert.ToInt32(exp / 365);

int exp\_month = Convert.ToInt32((exp - exp\_year \* 365) / 30);

int exp\_day = exp - exp\_year \* 365 - exp\_month \* 30;

textBox3.Text = exp\_day.ToString();

textBox5.Text = exp\_month.ToString();

textBox11.Text = exp\_year.ToString();

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

// Активация кнопок

button1.Enabled = true;

button3.Enabled = true;

// Проверка полей на валидность

checkInputMechanic();

}

// Выбор записи в таблице автомобилей

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

{

// Получение ID записи

cur\_id = dataGridView2.SelectedCells[0].Value.ToString();

// Подстановка данных в поля

textBox8.Text = dataGridView2.SelectedCells[1].Value.ToString();

textBox9.Text = dataGridView2.SelectedCells[2].Value.ToString();

textBox10.Text = dataGridView2.SelectedCells[3].Value.ToString();

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

textBox7.Text = dataGridView2.SelectedCells[5].Value.ToString();

// Активация кнопок

button1.Enabled = true;

button3.Enabled = true;

// Проверка полей на валидность

checkInputCar();

}

// Проверка полей при изменении

private void textBox4\_KeyUp(object sender, KeyEventArgs e) { checkInputMechanic(); }

private void comboBox1\_TextChanged(object sender, EventArgs e) { checkInputMechanic(); }

private void textBox9\_KeyUp(object sender, KeyEventArgs e) { checkInputCar(); }

private void comboBox2\_TextChanged(object sender, EventArgs e) { checkInputCar(); }

// Проверка полей механика

private bool checkInputMechanic()

{

try

{

// Проверка есть ли пустые поля

if (textBox1.Text != "" && textBox2.Text != "" && textBox4.Text != "" && textBox5.Text != "" && textBox6.Text != "" && comboBox1.Text != "")

{

// Проверка полей на правильность ввода

if (MainFunc.stringTest(textBox1.Text, @"^[a-zA-Zа-яА-Я]\*$") && MainFunc.stringTest(textBox2.Text, @"^[a-zA-Zа-яА-Я]\*$") && MainFunc.stringTest(textBox6.Text, @"^[a-zA-Zа-яА-Я]\*$") && MainFunc.stringTest(textBox4.Text, @"^[0-9]\*$") && MainFunc.stringTest(textBox3.Text, @"^[0-9]\*$") && MainFunc.stringTest(textBox5.Text, @"^[0-9]\*$") && MainFunc.stringTest(textBox11.Text, @"^[0-9]\*$"))

{

button1.Enabled = true;

return true;

}

else

{

button1.Enabled = false;

}

}

else

{

button1.Enabled = false;

}

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return false;

}

// Отчистка полей механика

private void clearInputMechanic()

{

cur\_id = "";

textBox4.Text = "";

textBox1.Text = "";

textBox2.Text = "";

textBox6.Text = "";

textBox3.Text = "";

textBox5.Text = "";

textBox11.Text = "";

button1.Enabled = false;

button3.Enabled = false;

}

// Изменение механиков

private void editMechanic()

{

try

{

if (checkInputMechanic())

{

int exp = Convert.ToInt32(textBox3.Text) + Convert.ToInt32(textBox5.Text) \* 30 + Convert.ToInt32(textBox11.Text) \* 365;

// Добавление записи

MainFunc.sql("UPDATE mechanic SET " +

"mechanic\_number = " + textBox4.Text + ", " +

"mechanic\_surname = '" + textBox1.Text + "', " +

"mechanic\_name = '" + textBox2.Text + "', " +

"mechanic\_patronymic = '" + textBox6.Text + "', " +

"mechanic\_exp = '" + exp + "', " +

"mechanic\_rank = " + comboBox1.Text + " " +

"WHERE mechanic\_id = " + cur\_id + ";");

clearInputMechanic();

// Обновление таблиц

updateTables();

}

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

// Проверка полей автомобиля

private bool checkInputCar()

{

try

{

// Проверка есть ли пустые поля

if (textBox7.Text != "" && textBox8.Text != "" && textBox9.Text != "" && textBox10.Text != "" && comboBox2.Text != "")

{

if (MainFunc.stringTest(textBox9.Text, @"^[a-zA-Zа-яА-Я\-]\*$") && MainFunc.stringTest(textBox7.Text, @"^[0-9]\*$") && MainFunc.stringTest(textBox8.Text, @"^[0-9]\*$"))

{

button1.Enabled = true;

return true;

}

else

{

button1.Enabled = false;

}

}

else

{

button1.Enabled = false;

}

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return false;

}

// Отчистка полей машины

private void clearInputCar()

{

cur\_id = "";

textBox8.Text = "";

textBox9.Text = "";

textBox10.Text = "";

textBox7.Text = "";

button1.Enabled = false;

button3.Enabled = false;

}

// Изменение данных машины

private void editCar()

{

try

{

if (checkInputCar())

{

// Добавление записи

MainFunc.sql("UPDATE car SET " +

"car\_number = " + textBox8.Text + ", " +

"car\_mark = '" + textBox9.Text + "', " +

"car\_name = '" + textBox10.Text + "', " +

"car\_type = '" + comboBox2.Text + "', " +

"car\_year = " + textBox7.Text + " " +

"WHERE car\_id = " + cur\_id + ";");

clearInputCar();

// Обновление таблиц

updateTables();

}

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

// Кнопка изменения

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

{

DialogResult result = MessageBox.Show(this, "Вы точно хотите изменить запись?", "Изменение записи", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

// Определяем что добавляем

if (tabControl1.SelectedIndex == 0)

{

// Добавляем механика

editMechanic();

}

else if (tabControl1.SelectedIndex == 1)

{

// Добавляем машину

editCar();

}

}

}

// Кнопка удаления

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

{

try

{

DialogResult result = MessageBox.Show(this, "Вы точно хотите удалить запись?", "Удаление записи", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

// Определяем что добавляем

if (tabControl1.SelectedIndex == 0)

{

MainFunc.sql("DELETE FROM mechanic WHERE mechanic\_id = " + cur\_id + ";");

clearInputMechanic();

// Обновление таблиц

updateTables();

}

else if (tabControl1.SelectedIndex == 1)

{

MainFunc.sql("DELETE FROM car WHERE car\_id = " + cur\_id + ";");

clearInputCar();

// Обновление таблиц

updateTables();

}

}

} catch

{

MessageBox.Show(this, "Запись, которую вы хотите удалить, используется.", "Ошибка удаления", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**addOrder.cs** – форма добавления заказа

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 carShowroom

{

public partial class addOrder : Form

{

public addOrder()

{

InitializeComponent();

}

// Полное закрытие программы

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

{

Environment.Exit(0);

}

// Открыть форму меню

private void button2\_Click\_1(object sender, EventArgs e)

{

menu menu = new menu();

menu.Show();

Hide();

}

// Подключение класса с функциями

Function MainFunc = new Function();

// Загрузка формы добавление заказа

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

{

// Заполнение поля выбора механика

OleDbDataAdapter data1 = new OleDbDataAdapter(MainFunc.getAll("mechanic"));

DataTable table1 = new DataTable();

data1.Fill(table1);

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

{

string item = table1.Rows[curRow][0].ToString() + ": " + table1.Rows[curRow][2].ToString() + " " + table1.Rows[curRow][3].ToString() + " " + table1.Rows[curRow][4].ToString();

comboBox1.Items.Add(item);

}

// Заполнение поля выбора машины

OleDbDataAdapter data2 = new OleDbDataAdapter(MainFunc.getAll("car"));

DataTable table2 = new DataTable();

data2.Fill(table2);

for (int curRow = 0; curRow < table2.Rows.Count; curRow++)

{

string item = table2.Rows[curRow][0].ToString() + ": " + table2.Rows[curRow][2].ToString() + " " + table2.Rows[curRow][3].ToString();

comboBox2.Items.Add(item);

}

dateTimePicker1.MinDate = DateTime.Today;

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e){ checkInput(); }

private void textBox2\_KeyUp(object sender, KeyEventArgs e) { checkInput(); }

// Проверка полей

private bool checkInput()

{

try

{

// Проверка есть ли пустые поля

if (textBox2.Text != "" && textBox3.Text != "" && textBox4.Text != "" && textBox5.Text != "" && comboBox1.Text != "" && comboBox2.Text != "")

{

// Проверка полей на правильность ввода

if (MainFunc.stringTest(textBox2.Text, @"^[0-9]\*$") && MainFunc.stringTest(textBox2.Text, @"^[0-9]\*$") && MainFunc.stringTest(textBox3.Text, @"^[0-9]\*$") && MainFunc.stringTest(textBox4.Text, @"^[0-9]\*$") && MainFunc.stringTest(textBox5.Text, @"^[0-9]\*$"))

{

button1.Enabled = true;

return true;

}

else

{

button1.Enabled = false;

}

}

else

{

button1.Enabled = false;

}

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return false;

}

// Отчистка полей

private void clearInput()

{

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

}

// Кнопка добавления

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

{

try

{

Random rnd = new Random();

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

if (checkInput())

{

// Добавление записи

MainFunc.sql("INSERT INTO repair VALUES (" +

temp + ", " +

comboBox1.Text.Split(':')[0] + ", " +

comboBox2.Text.Split(':')[0] + ", " +

"'" + dateTimePicker1.Value.Day + "." + dateTimePicker1.Value.Month + "." + dateTimePicker1.Value.Year + "', " +

"'" + textBox3.Text + " д. " + textBox4.Text + " ч. " + textBox5.Text + " мин.', " +

textBox2.Text +

");");

clearInput();

MessageBox.Show(this, "Запись добавлена!", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**allOrder.cs** – форма просмотра заказов

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 carShowroom

{

public partial class allOrder : Form

{

public allOrder()

{

InitializeComponent();

}

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

{

menu menu = new menu();

menu.Show();

Hide();

}

// Полное закрытие программы

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

{

Environment.Exit(0);

}

// Подключение класса с функциями

Function MainFunc = new Function();

private string cur\_id = "";

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

{

updateTables();

}

private void updateTables()

{

try

{

// Таблица заказов

OleDbDataAdapter data1 = new OleDbDataAdapter(MainFunc.sql("SELECT repair\_id, (SELECT mechanic\_surname FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [m\_surname], (SELECT mechanic\_name FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [m\_name], (SELECT mechanic\_patronymic FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [m\_patronymic], (SELECT car\_name FROM car WHERE repair.car\_id = car.car\_id) AS [model\_car], (SELECT car\_mark FROM car WHERE repair.car\_id = car.car\_id) AS [mark\_car], repair\_date, repair\_cost FROM repair;"));

DataTable table1 = new DataTable();

data1.Fill(table1);

table1.Columns["repair\_id"].ColumnName = "ID";

table1.Columns["m\_surname"].ColumnName = "Фамилия";

table1.Columns["m\_name"].ColumnName = "Имя";

table1.Columns["m\_patronymic"].ColumnName = "Отчество";

table1.Columns["model\_car"].ColumnName = "Модель";

table1.Columns["mark\_car"].ColumnName = "Марка";

table1.Columns["repair\_date"].ColumnName = "Дата";

table1.Columns["repair\_cost"].ColumnName = "Стоимость";

dataGridView1.DataSource = table1;

dataGridView1.Columns[0].Visible = false;

label13.Text = "Количество записей: " + table1.Rows.Count.ToString();

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

cur\_id = dataGridView1.SelectedCells[0].Value.ToString();

button1.Enabled = true;

}

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

{

try

{

DialogResult result = MessageBox.Show(this, "Вы точно хотите удалить запись?", "Удаление записи", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

MainFunc.sql("DELETE FROM repair WHERE repair\_id = " + cur\_id + ";");

cur\_id = "";

button1.Enabled = false;

updateTables();

}

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

**import.cs** – форма импорта данных

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;

using Word = Microsoft.Office.Interop.Word;

using Excel = Microsoft.Office.Interop.Excel;

namespace carShowroom

{

public partial class import : Form

{

public import()

{

InitializeComponent();

}

// Подключение класса с функциями

Function MainFunc = new Function();

// Проверка поля на правильность ввода

private bool checkInput()

{

try

{

// Проверка есть ли пустые поля

if (comboBox1.Text != "")

{

button1.Enabled = true;

return true;

}

else

{

button1.Enabled = false;

}

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

return false;

}

// Создание чека

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

{

try

{

// Проверка полей

if (checkInput())

{

// Получение данных из таблицы

OleDbDataAdapter data1 = new OleDbDataAdapter(MainFunc.sql("SELECT " +

"repair\_id, " +

"(SELECT mechanic\_surname FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [m\_surname], " +

"(SELECT mechanic\_name FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [m\_name], " +

"(SELECT mechanic\_patronymic FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [m\_patronymic], " +

"(SELECT car\_name FROM car WHERE repair.car\_id = car.car\_id) AS [model\_car], " +

"(SELECT car\_mark FROM car WHERE repair.car\_id = car.car\_id) AS [mark\_car], " +

"repair\_date, " +

"repair\_cost " +

"FROM repair " +

"WHERE repair\_id = " + comboBox1.Text.Split(':')[0] + ";"));

DataTable table1 = new DataTable();

data1.Fill(table1);

// Создание объекта ворда

Word.Application wordApp = new Word.Application();

// Получение шаблона чека

Word.Document doc = wordApp.Documents.Add(Environment.CurrentDirectory + "\\check.docx");

// Показать документ

wordApp.Visible = true;

// Вывод данных в документ

doc.Bookmarks["number"].Range.Text = table1.Rows[0][0].ToString();

doc.Bookmarks["date"].Range.Text = table1.Rows[0][6].ToString();

doc.Bookmarks["worker"].Range.Text = table1.Rows[0][1].ToString() + " " + table1.Rows[0][2].ToString() + " " + table1.Rows[0][3].ToString();

doc.Bookmarks["car"].Range.Text = table1.Rows[0][5].ToString() + " " + table1.Rows[0][4].ToString();

doc.Bookmarks["orderName"].Range.Text = "Ремонт автомобиля " + table1.Rows[0][5].ToString() + " " + table1.Rows[0][4].ToString();

doc.Bookmarks["price"].Range.Text = table1.Rows[0][7].ToString();

}

}

catch

{

MessageBox.Show(this, "Произошла критическая ошибка!", "Критическая ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

// Полное закрытие программы

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

{

Environment.Exit(0);

}

// Загрузка формы импорта

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

{

// Получение данных о заказах

OleDbDataAdapter data1 = new OleDbDataAdapter(MainFunc.sql("SELECT " +

"repair\_id, " +

"(SELECT mechanic\_surname FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [m\_surname], " +

"(SELECT mechanic\_name FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [m\_name], " +

"(SELECT mechanic\_patronymic FROM mechanic WHERE repair.mechanic\_id = mechanic.mechanic\_id) AS [m\_patronymic], " +

"(SELECT car\_name FROM car WHERE repair.car\_id = car.car\_id) AS [model\_car], " +

"(SELECT car\_mark FROM car WHERE repair.car\_id = car.car\_id) AS [mark\_car] " +

"FROM repair"));

DataTable table1 = new DataTable();

data1.Fill(table1);

// Добавление записей в поле выбора

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

{

string item = table1.Rows[curRow][0].ToString() + ": " + table1.Rows[curRow][1].ToString() + " " + table1.Rows[curRow][2].ToString().Substring(0, 1) + ". " + table1.Rows[curRow][3].ToString().Substring(0, 1) + ". - " + table1.Rows[curRow][4].ToString() + " " + table1.Rows[curRow][5].ToString();

comboBox1.Items.Add(item);

}

}

// Открыть форму меню

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

{

menu menu = new menu();

menu.Show();

Hide();

}

// Проверка правильности ввода при изменении поля выбора

private void comboBox1\_TextChanged(object sender, EventArgs e)

{

checkInput();

}

// Импорт всех таблиц в Excel

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

{

// Соединение с базой

string connect = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=" + MainFunc.db\_name + ";";

OleDbConnection ODConnect = new OleDbConnection(connect);

ODConnect.Open();

// Создание Excel документа

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

// Добавление листа

var workBook = excelApp.Workbooks.Add(Type.Missing);

// Получение листа

var mechanic = workBook.ActiveSheet;

// Имя листа

mechanic.Name = "mechanic";

// Объект таблицы

var mechanicCells = mechanic.Cells;

// Шапка

mechanic.Cells[1, 1] = "mechanic\_id";

mechanic.Cells[1, 2] = "mechanic\_number";

mechanic.Cells[1, 3] = "mechanic\_surname";

mechanic.Cells[1, 4] = "mechanic\_name";

mechanic.Cells[1, 5] = "mechanic\_patronymic";

mechanic.Cells[1, 6] = "mechanic\_exp";

mechanic.Cells[1, 7] = "mechanic\_rank";

// Получение данных

OleDbCommand mechanicData = new OleDbCommand("SELECT \* FROM mechanic;");

mechanicData.Connection = ODConnect;

mechanicData.ExecuteNonQuery();

OleDbDataReader readerMechanic = mechanicData.ExecuteReader();

// Добовление данных в таблицу

int temp = 2;

while (readerMechanic.Read())

{

mechanic.Cells[temp, 1] = readerMechanic[0].ToString();

mechanic.Cells[temp, 2] = readerMechanic[1].ToString();

mechanic.Cells[temp, 3] = readerMechanic[2].ToString();

mechanic.Cells[temp, 4] = readerMechanic[3].ToString();

mechanic.Cells[temp, 5] = readerMechanic[4].ToString();

mechanic.Cells[temp, 6] = readerMechanic[5].ToString();

mechanic.Cells[temp, 7] = readerMechanic[6].ToString();

temp++;

}

mechanic.Columns.AutoFit();

mechanic.Rows.AutoFit();

// Добавление листа

var car = workBook.Sheets.Add(After: workBook.ActiveSheet);

// Имя листа

car.Name = "car";

// Объект таблицы

var carCells = car.Cells;

// Шапка

car.Cells[1, 1] = "car\_id";

car.Cells[1, 2] = "car\_number";

car.Cells[1, 3] = "car\_mark";

car.Cells[1, 4] = "car\_name";

car.Cells[1, 5] = "car\_type";

car.Cells[1, 6] = "car\_year";

// Получение данных

OleDbCommand carData = new OleDbCommand("SELECT \* FROM car;");

carData.Connection = ODConnect;

carData.ExecuteNonQuery();

OleDbDataReader readerCar = carData.ExecuteReader();

// Добовление данных в таблицу

temp = 2;

while (readerCar.Read())

{

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

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

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

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

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

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

temp++;

}

car.Columns.AutoFit();

car.Rows.AutoFit();

// Добавление листа

var repair = workBook.Sheets.Add(After: workBook.ActiveSheet);

// Имя листа

repair.Name = "repair";

// Объект таблицы

var repairCells = repair.Cells;

// Шапка

repair.Cells[1, 1] = "repair\_id";

repair.Cells[1, 2] = "mechanic\_id";

repair.Cells[1, 3] = "car\_id";

repair.Cells[1, 4] = "repair\_date";

repair.Cells[1, 5] = "repair\_time";

repair.Cells[1, 6] = "repair\_cost";

// Получение данных

OleDbCommand repairData = new OleDbCommand("SELECT \* FROM repair;");

repairData.Connection = ODConnect;

repairData.ExecuteNonQuery();

OleDbDataReader readerRepair = repairData.ExecuteReader();

// Добовление данных в таблицу

temp = 2;

while (readerRepair.Read())

{

repair.Cells[temp, 1] = readerRepair[0].ToString();

repair.Cells[temp, 2] = readerRepair[1].ToString();

repair.Cells[temp, 3] = readerRepair[2].ToString();

repair.Cells[temp, 4] = readerRepair[3].ToString();

repair.Cells[temp, 5] = readerRepair[4].ToString();

repair.Cells[temp, 6] = readerRepair[5].ToString();

temp++;

}

repair.Columns.AutoFit();

repair.Rows.AutoFit();

excelApp.Visible = true;

ODConnect.Close();

}

}

}

**Function.cs** – основные функции программы

using System;

using System.Collections.Generic;

using System.Data.OleDb;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

namespace carShowroom

{

class Function

{

public string db\_name = "db.mdb";

// Функция SQL запроса

public OleDbCommand sql(string query)

{

string connect = "Provider= Microsoft.Jet.OLEDB.4.0; Data Source=" + this.db\_name + ";";

OleDbConnection ODConnect = new OleDbConnection(connect);

ODConnect.Open();

OleDbCommand sql = new OleDbCommand(query);

sql.Connection = ODConnect;

sql.ExecuteNonQuery();

ODConnect.Close();

return sql;

}

// Функция получения данных таблицы

public OleDbCommand getAll(string table)

{

return this.sql("SELECT \* FROM " + table + ";");

}

// Проверка строки на соответствие регулярному выражению

public bool stringTest(string str, string reg)

{

Regex regex = new Regex(reg);

MatchCollection matches = regex.Matches(str);

if (matches.Count > 0)

{

return true;

}

return false;

}

}

}