Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое занятие 13\_3**»

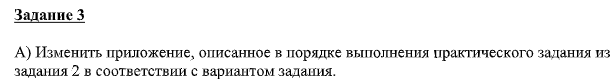
Выполнил: Юнусов Руслан Александрович

Группа: ПР-23

Преподаватель: Мирошниченко Г.В

2024

**Задание: Вариант 2**

****

**Входные и выходные данные**

carDictionary – Dictionary<TKey, TValue> - коллекция Dictionary, список автомобилей

currentKey – int, целочисленное значение – ключ для списка carDictionary

editKey – int, целочисленное значение – измененый ключ для списка carDicrionary

contextMenuStrip1 – ContextMenuStrip, контекстное меню, меню для удаления автомобиля

dataGridView1 – DataGridView, таблица, список всех автомобилей

modelEnter – string, строка, ввод модели автомобиля

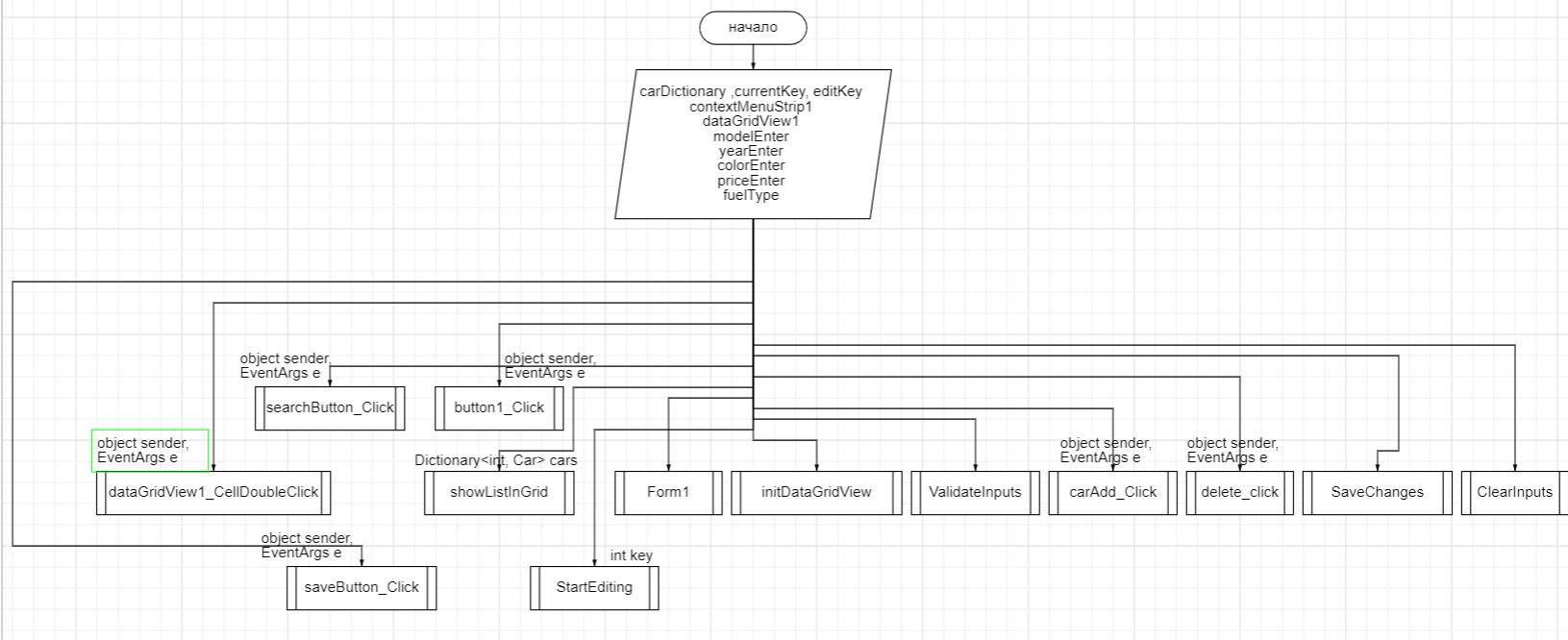
yearEnter – string, строка, ввод года выпуска автомобиля

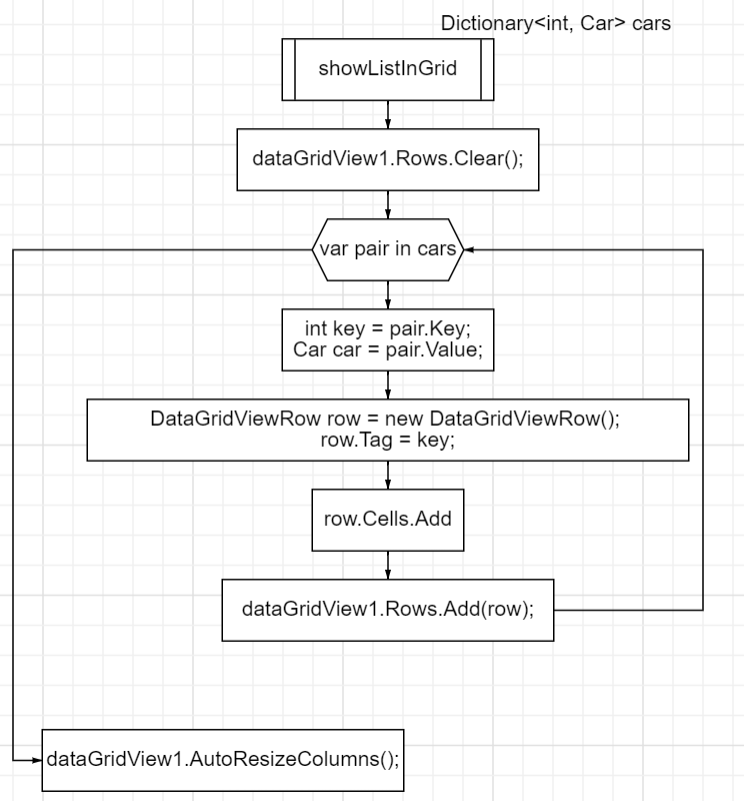
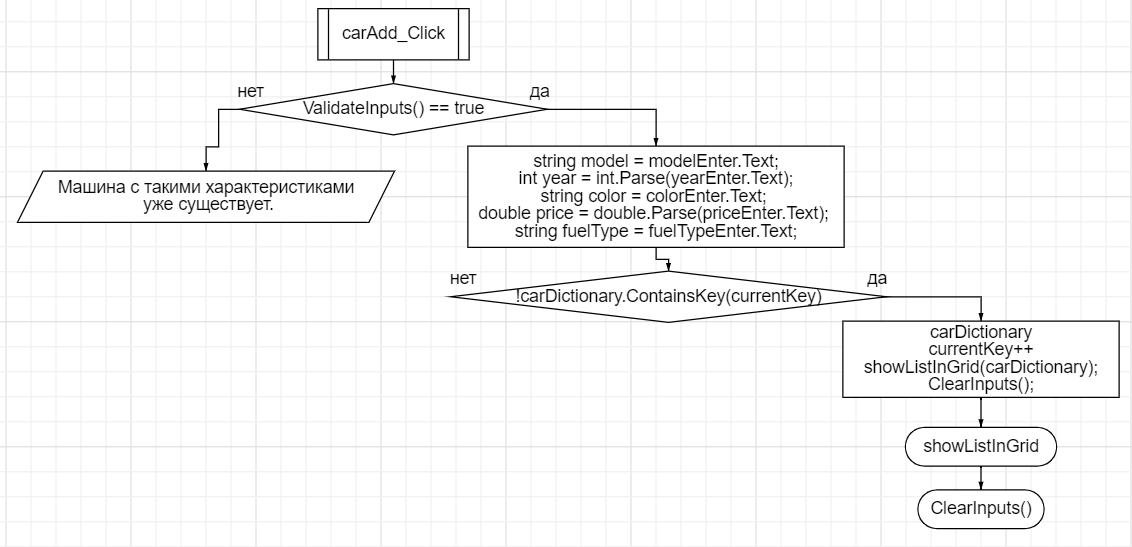
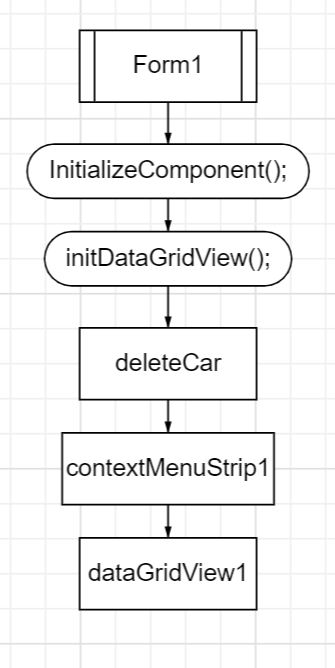
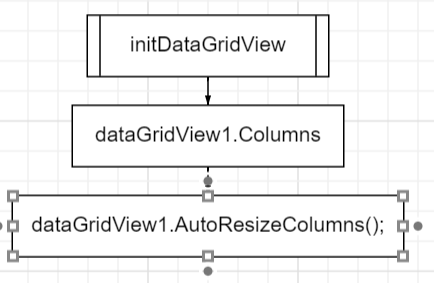
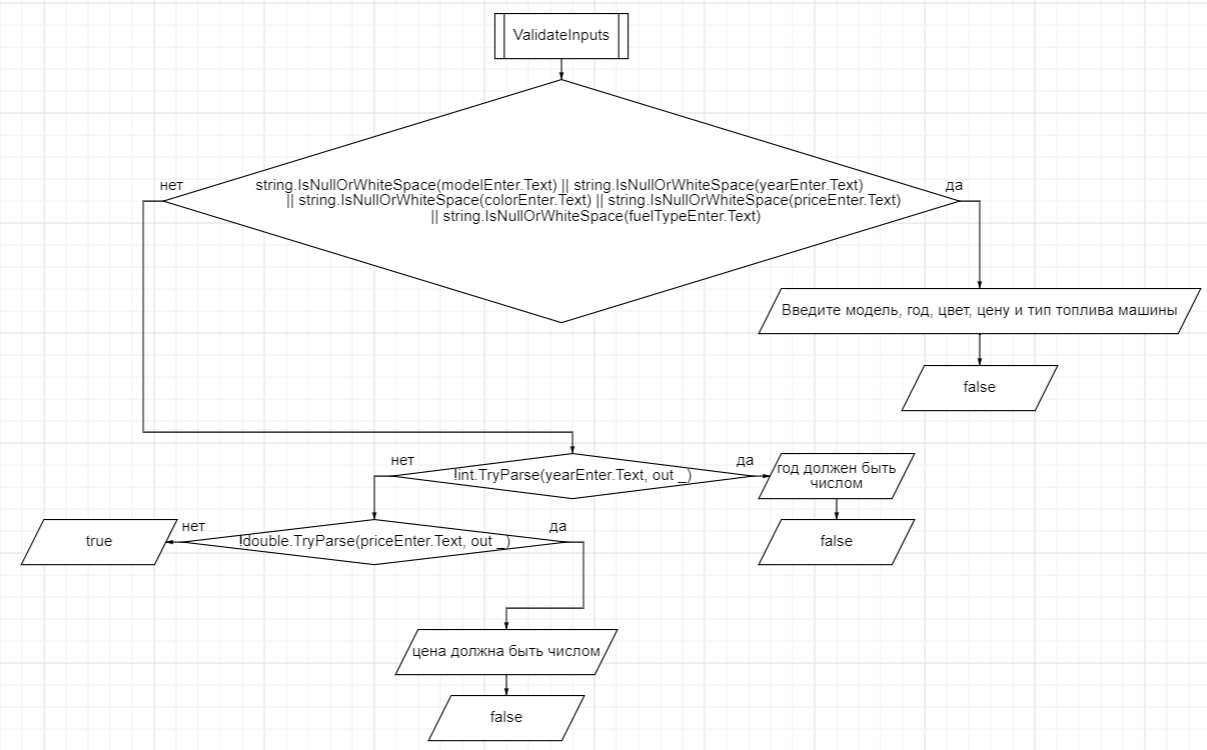
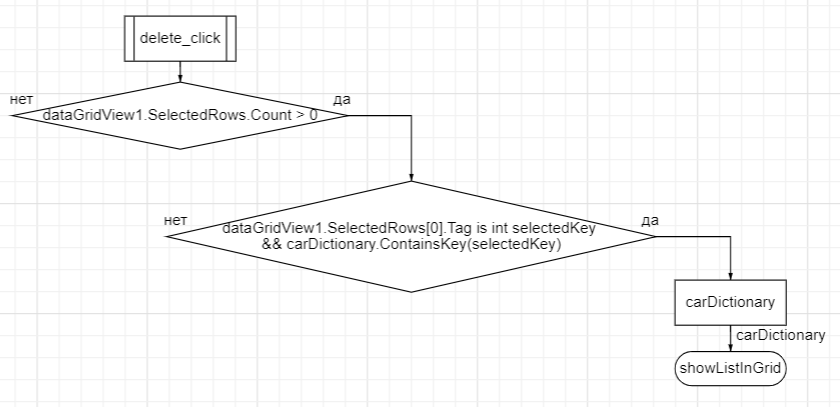
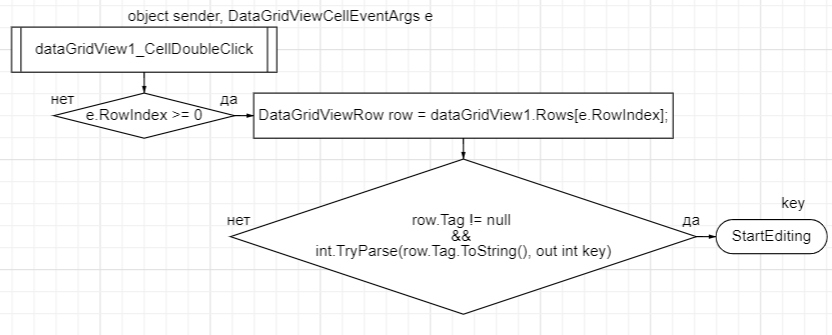
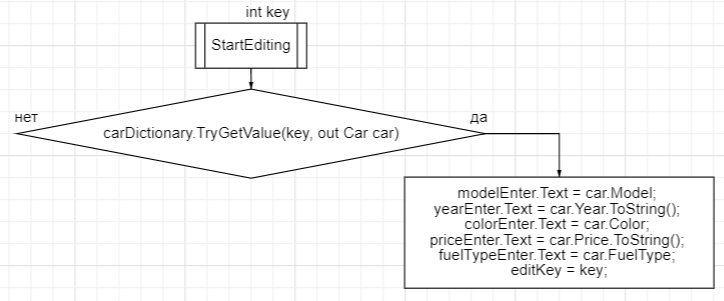
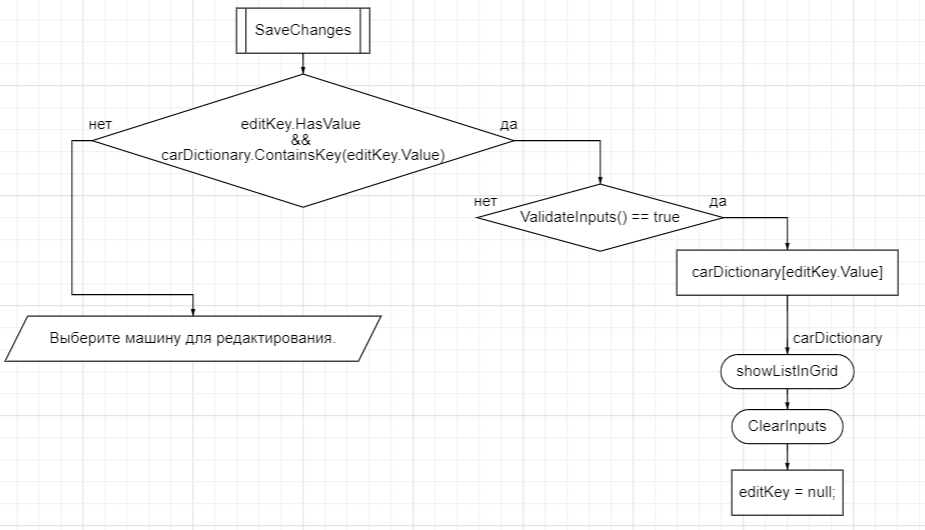
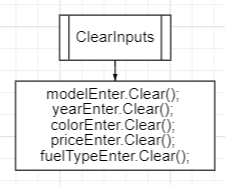
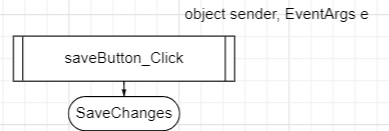
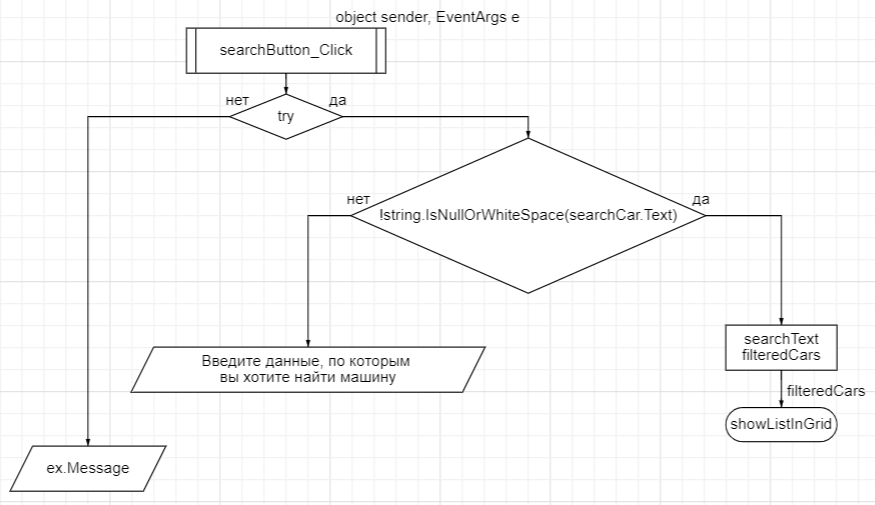
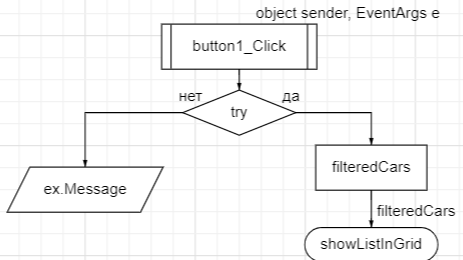
colorEnter – string, строка, ввод цвета автомобиля

priceEnter – string, строка, ввод цены автомобиля

fuelType – string, строка, ввод типа топлива автомобиля

**Блок-схема:**

****

**** ****           

**Листинг программы (если есть)**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

namespace \_13\_2prakt

{

public partial class Form1 : Form

{

private Dictionary<int, Car> carDictionary = new Dictionary<int, Car>();

private int currentKey = 0;

private int? editKey = null;

public Form1()

{

InitializeComponent();

initDataGridView();

var deleteCar = new ToolStripMenuItem("удалить");

deleteCar.Click += delete\_click;

contextMenuStrip1.Items.Add(deleteCar);

dataGridView1.ContextMenuStrip = contextMenuStrip1;

dataGridView1.CellDoubleClick += dataGridView1\_CellDoubleClick;

}

private void initDataGridView()

{

dataGridView1.Columns.AddRange(

new DataGridViewTextBoxColumn { HeaderText = "Модель", ValueType = typeof(string) },

new DataGridViewTextBoxColumn { HeaderText = "Год", ValueType = typeof(int) },

new DataGridViewTextBoxColumn { HeaderText = "Цвет", ValueType = typeof(string) },

new DataGridViewTextBoxColumn { HeaderText = "Цена", ValueType = typeof(double) },

new DataGridViewTextBoxColumn { HeaderText = "Топливо", ValueType = typeof(string) }

);

dataGridView1.AutoResizeColumns();

}

private void showListInGrid(Dictionary<int, Car> cars)

{

dataGridView1.Rows.Clear();

foreach (var pair in cars)

{

int key = pair.Key;

Car car = pair.Value;

DataGridViewRow row = new DataGridViewRow();

row.Tag = key;

row.Cells.Add(new DataGridViewTextBoxCell { Value = car.Model });

row.Cells.Add(new DataGridViewTextBoxCell { Value = car.Year });

row.Cells.Add(new DataGridViewTextBoxCell { Value = car.Color });

row.Cells.Add(new DataGridViewTextBoxCell { Value = car.Price });

row.Cells.Add(new DataGridViewTextBoxCell { Value = car.FuelType });

dataGridView1.Rows.Add(row);

}

}

private bool ValidateInputs()

{

if (string.IsNullOrWhiteSpace(modelEnter.Text) || string.IsNullOrWhiteSpace(yearEnter.Text)

|| string.IsNullOrWhiteSpace(colorEnter.Text) || string.IsNullOrWhiteSpace(priceEnter.Text)

|| string.IsNullOrWhiteSpace(fuelTypeEnter.Text))

{

MessageBox.Show("Введите модель, год, цвет, цену и тип топлива машины");

return false;

}

if (!int.TryParse(yearEnter.Text, out \_))

{

MessageBox.Show("год должен быть целым числом");

return false;

}

if (!double.TryParse(priceEnter.Text, out \_))

{

MessageBox.Show("цена должна быть числом");

return false;

}

return true;

}

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

{

if (ValidateInputs())

{

string model = modelEnter.Text;

int year = int.Parse(yearEnter.Text);

string color = colorEnter.Text;

double price = double.Parse(priceEnter.Text);

string fuelType = fuelTypeEnter.Text;

if (!carDictionary.ContainsKey(currentKey))

{

carDictionary.Add(currentKey, new Car(model, year, color, price, fuelType));

currentKey++;

showListInGrid(carDictionary);

ClearInputs();

}

else

{

MessageBox.Show("Машина с такими характеристиками уже существует.");

}

}

}

private void delete\_click(object sender, EventArgs e)

{

if (dataGridView1.SelectedRows.Count > 0)

{

if (dataGridView1.SelectedRows[0].Tag is int selectedKey && carDictionary.ContainsKey(selectedKey))

{

carDictionary.Remove(selectedKey);

showListInGrid(carDictionary);

}

}

}

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

{

if (e.RowIndex >= 0)

{

DataGridViewRow row = dataGridView1.Rows[e.RowIndex];

if (row.Tag != null && int.TryParse(row.Tag.ToString(), out int key))

{

StartEditing(key);

}

}

}

private void StartEditing(int key)

{

if (carDictionary.TryGetValue(key, out Car car))

{

modelEnter.Text = car.Model;

yearEnter.Text = car.Year.ToString();

colorEnter.Text = car.Color;

priceEnter.Text = car.Price.ToString();

fuelTypeEnter.Text = car.FuelType;

editKey = key;

}

}

private void SaveChanges()

{

if (editKey.HasValue && carDictionary.ContainsKey(editKey.Value))

{

if (ValidateInputs())

{

carDictionary[editKey.Value] = new Car(

modelEnter.Text,

int.Parse(yearEnter.Text),

colorEnter.Text,

double.Parse(priceEnter.Text),

fuelTypeEnter.Text);

showListInGrid(carDictionary);

ClearInputs();

editKey = null;

}

}

else

{

MessageBox.Show("Выберите машину для редактирования.");

}

}

private void ClearInputs()

{

modelEnter.Clear();

yearEnter.Clear();

colorEnter.Clear();

priceEnter.Clear();

fuelTypeEnter.Clear();

}

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

{

SaveChanges();

}

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

{

try

{

if (!string.IsNullOrWhiteSpace(searchCar.Text))

{

var searchText = searchCar.Text.ToLower();

var filteredCars = carDictionary.Where(s =>

s.Value.Model.ToLower().Contains(searchText) ||

s.Value.Year.ToString().ToLower().Contains(searchText) ||

s.Value.Color.ToLower().Contains(searchText) ||

s.Value.Price.ToString().ToLower().Contains(searchText) ||

s.Value.FuelType.ToLower().Contains(searchText)).ToDictionary(pair => pair.Key, pair => pair.Value);

showListInGrid(filteredCars);

}

else

{

MessageBox.Show("Введите данные, по которым вы хотите найти машину");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

try

{

var filteredCars = carDictionary.OrderBy(s => s.Value.Model).

ThenBy(s => s.Value.Year).

ThenBy(s => s.Value.Color).

ThenBy(s => s.Value.Price).

ThenBy(s => s.Value.FuelType).ToDictionary(pair => pair.Key, pair => pair.Value);

showListInGrid(filteredCars);

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_13\_2prakt

{

public class Car

{

public string Model;

public int Year;

public string Color;

public double Price;

public string FuelType;

public Car(string model, int year, string color, double price, string fuelType)

{

Model = model;

Year = year;

Color = color;

Price = price;

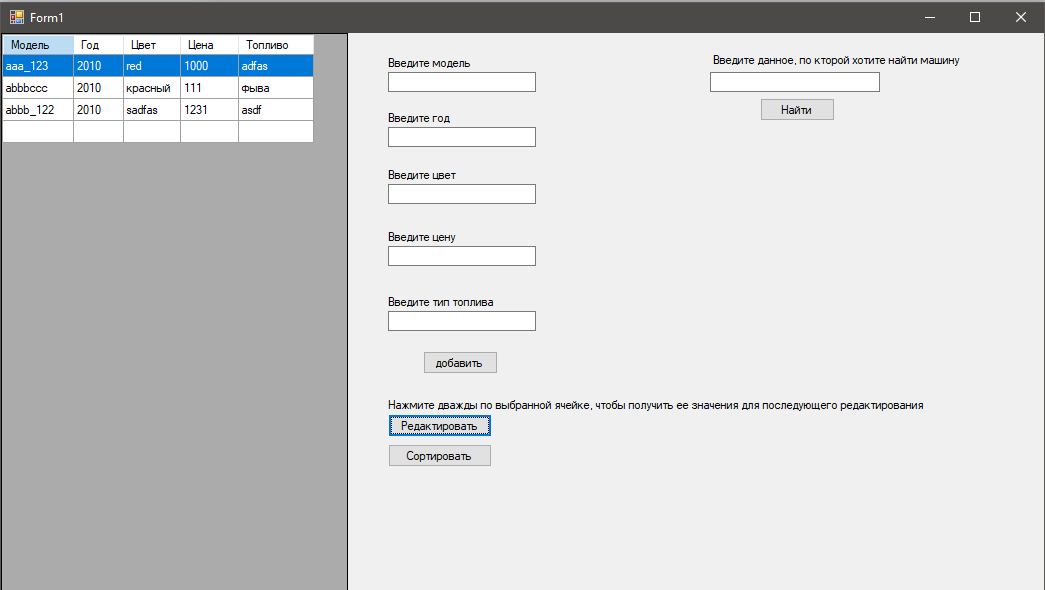
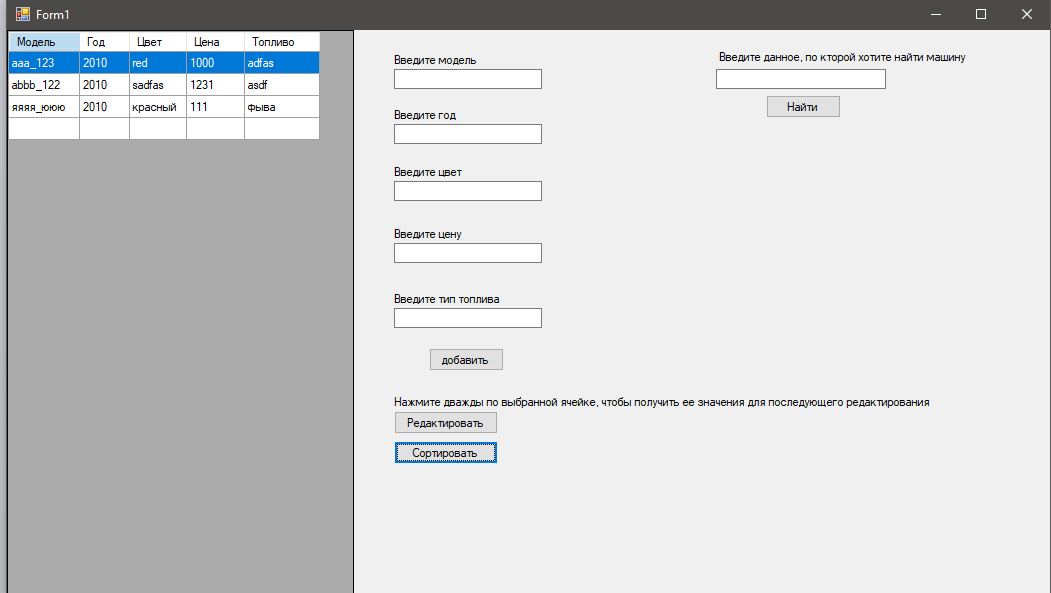
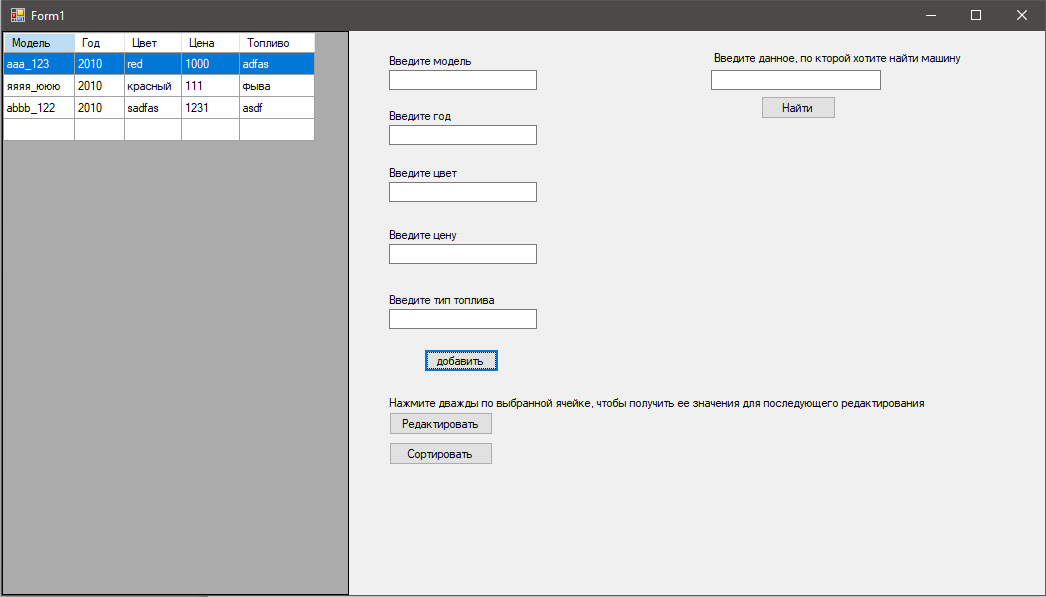
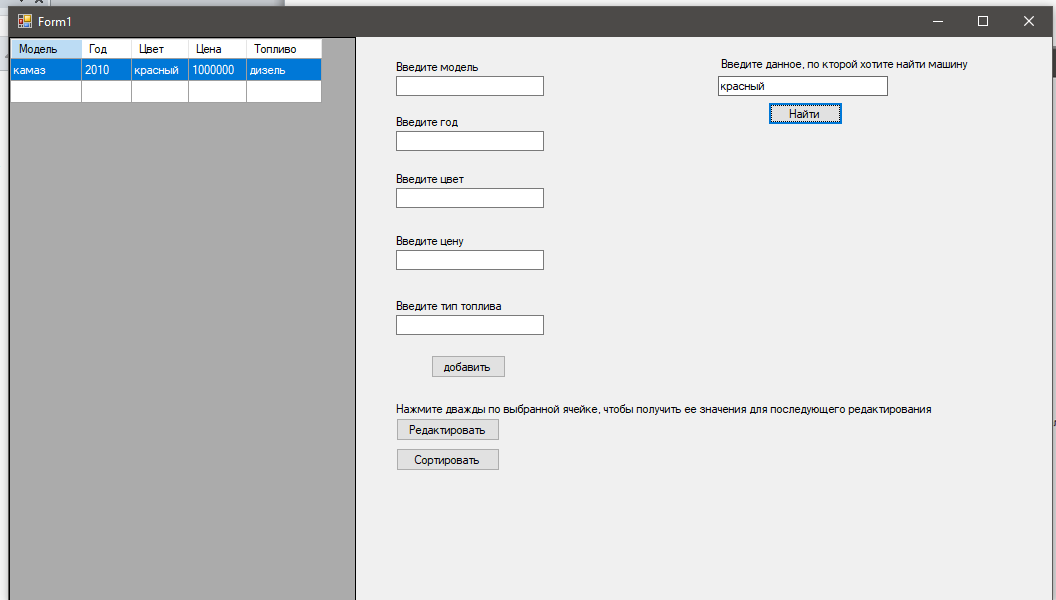
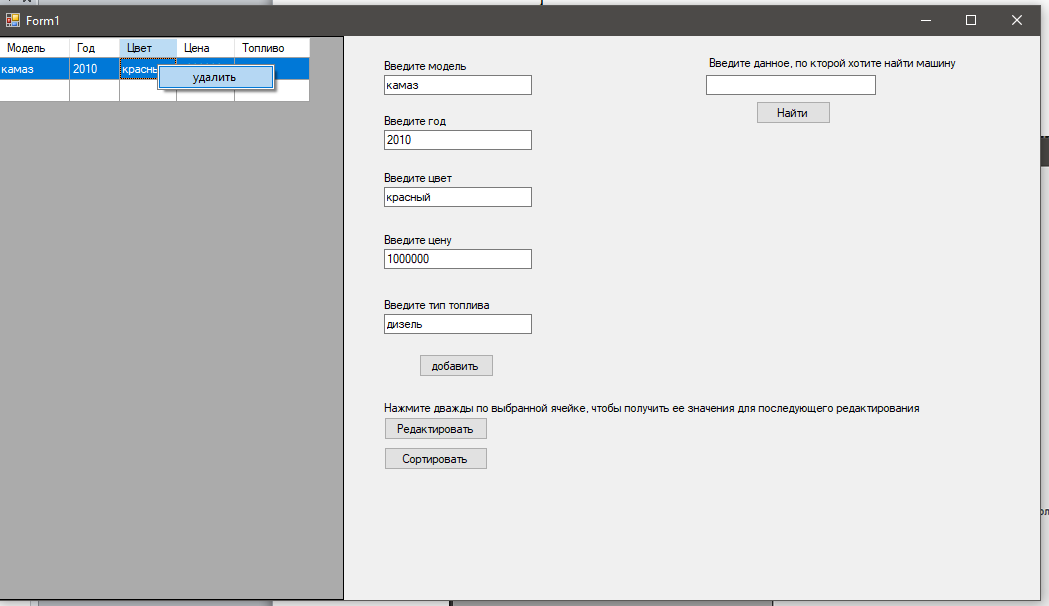
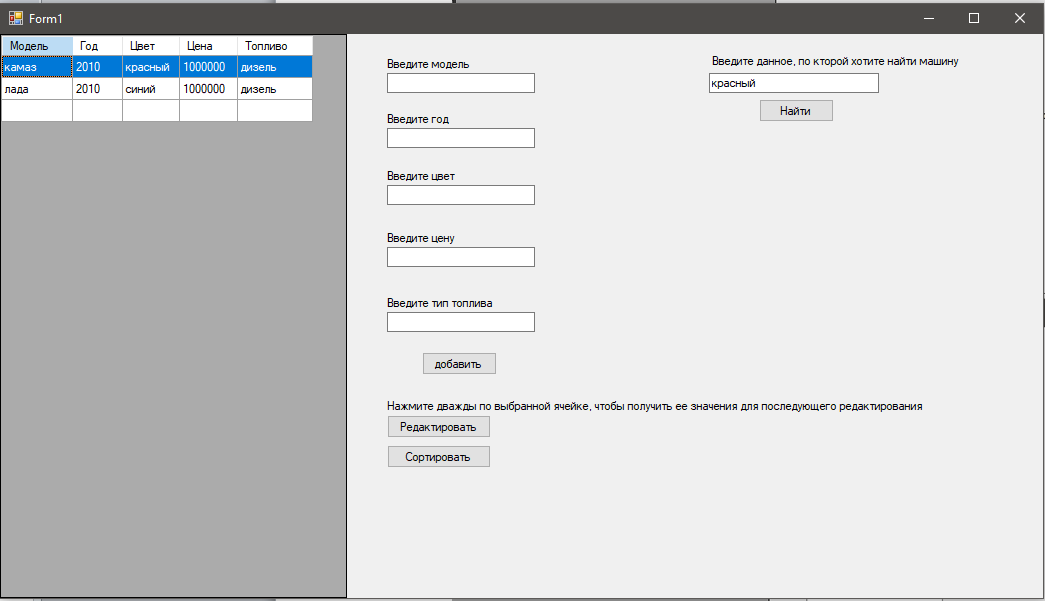
FuelType = fuelType;

}

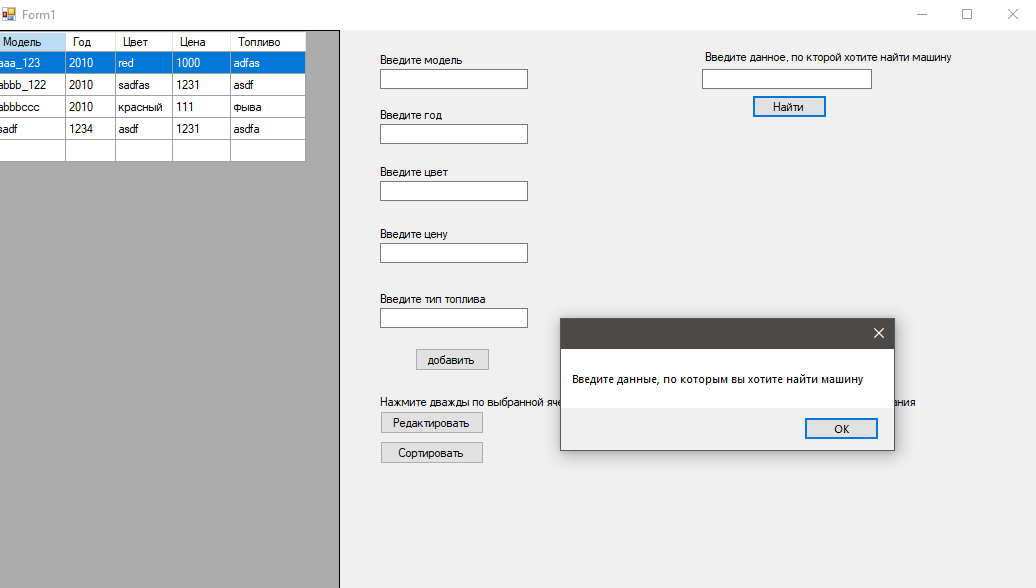
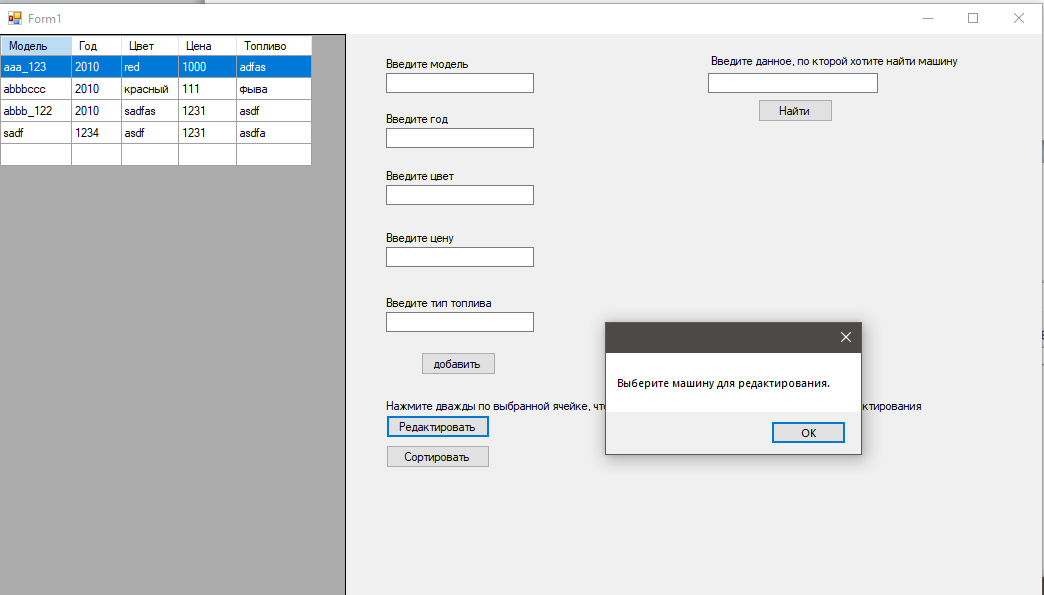
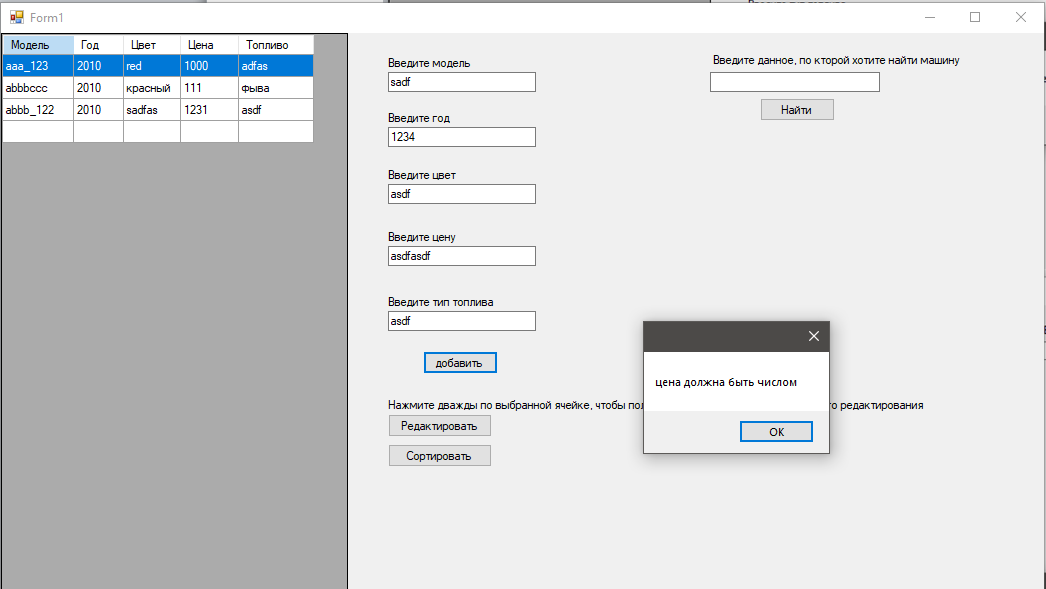
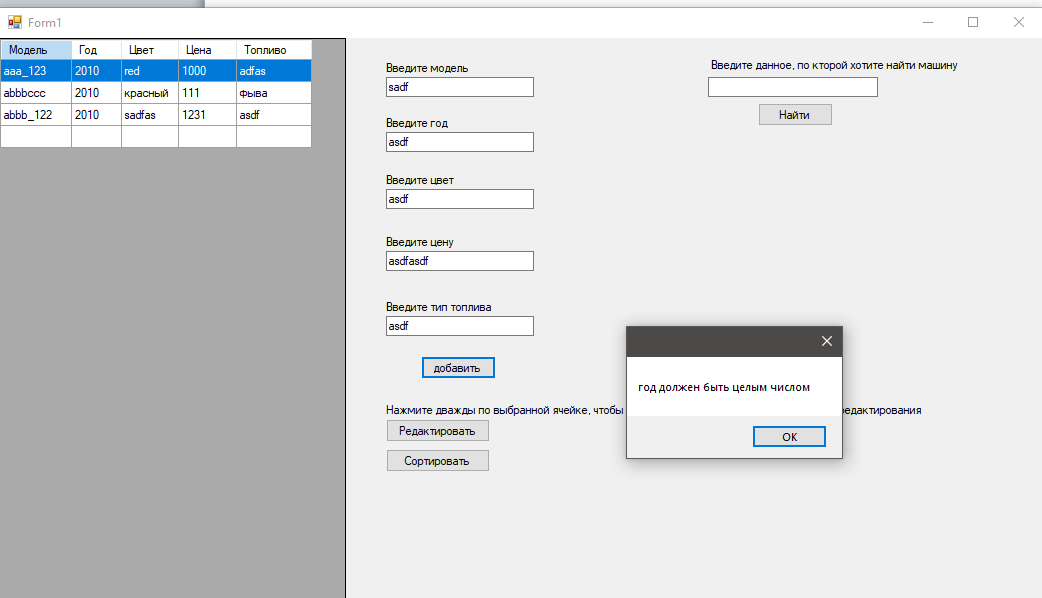
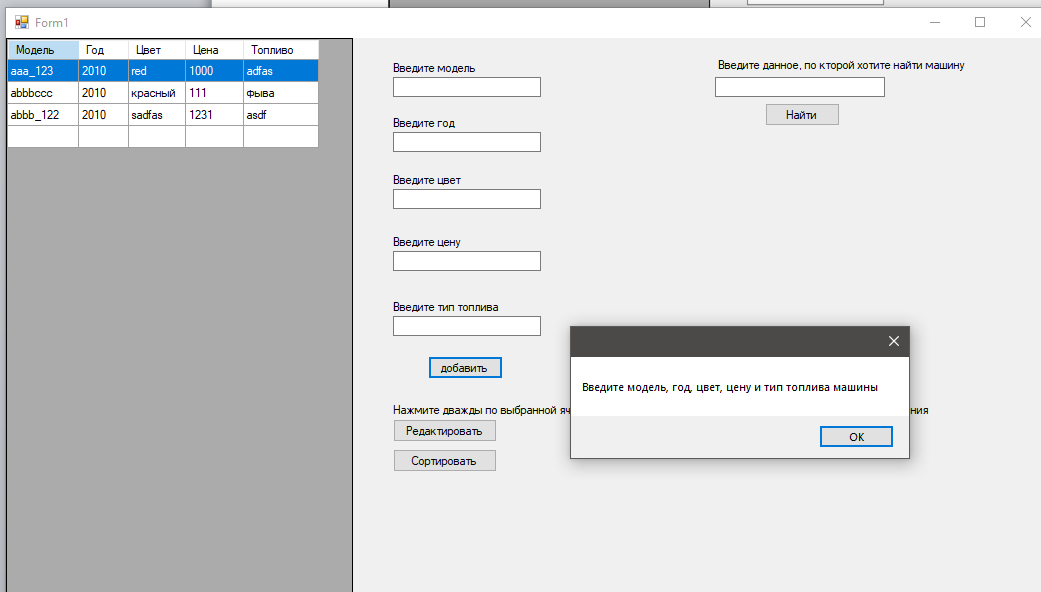
}

}

**Скрин-шот экранов**

****

**Тестовые ситуации**

****

**Вывод**

Я укрепил свои знания в ООП