Họ và tên: Hồ Nguyên Hà . Mã SV: 20it043

Bài Tập WDF Automobile

**AutomobileLibrary**

**BiussinessObject/Car.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AutomobileLibrary.BiussinessObject

{

public class Car

{

public int CarId { get; set; }

public string CarName { get; set; }

public string Manufacturer { get; set; }

public decimal Price { get; set; }

public int ReleaseYear { get; set; }

}

}

**DataAccess/CarDBContext.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using AutomobileLibrary.BiussinessObject;

namespace AutomobileLibrary.DataAccess

{

public class CarDBContext

{

private static List<Car> CarList = new List<Car>()

{

new Car{CarId=1,CarName="CRV",Manufacturer="honda",Price=300000,ReleaseYear=2020 },

new Car{CarId=2,CarName="Ford raptor",Manufacturer="Ford",Price=400000, ReleaseYear=2020 }

};

private static CarDBContext instance = null;

private static readonly object instanceLock = new object();

private CarDBContext() { }

public static CarDBContext Instance

{

get {

lock (instanceLock)

{

if (instance == null)

{

instance = new CarDBContext();

}

}

return instance; }

}

public List<Car> GetCarList => CarList;

public Car GetCarById(int carID)

{

Car car = CarList.SingleOrDefault(pro => pro.CarId == carID);

return car;

}

public void AddNew(Car car)

{

Car pro = GetCarById(car.CarId);

if(pro == null)

{

CarList.Add(car);

}

else

{

throw new Exception("Car is already exists");

}

}

public void Update(Car car)

{

Car c =GetCarById(car.CarId);

if (c !=null)

{

var index = CarList.IndexOf(c);

CarList[index] = car;

}

else

{

throw new Exception("car does not already exists");

}

}

public void Remove(int CarId)

{

Car p = GetCarById(CarId);

if(p != null)

{

CarList.Remove(p);

}

else

{

throw new Exception("Car does not already exists");

}

}

}

}

**Repository/CarRepository.cs**

using AutomobileLibrary.BiussinessObject;

using AutomobileLibrary.DataAccess;

namespace AutomobileLibrary.Repository

{

public class CarRepository : ICarRepository

{

public Car GetCarByID(int carId) => CarDBContext.Instance.GetCarById(carId);

public IEnumerable<Car> GetCars() => (IEnumerable<Car>)CarDBContext.Instance.GetCarList;

public void InsertCar(Car car) => CarDBContext.Instance.AddNew(car);

public void DeleteCar(int carId) => CarDBContext.Instance.Remove(carId);

public void UpdateCar(Car car) => CarDBContext.Instance.Update(car);

}

}

**Repository/ICarRepository.cs**

using AutomobileLibrary.BiussinessObject;

using System.Collections;

namespace AutomobileLibrary.Repository

{

public interface ICarRepository

{

IEnumerable<Car> GetCars();

Car GetCarByID(int carId);

void InsertCar(Car car);

void UpdateCar(Car car);

void DeleteCar(int carId);

}

}

**AutomobileWinApp**

**frmCarDetails.cs**

using AutomobileLibrary.BiussinessObject;

using AutomobileLibrary.Repository;

namespace AutomobileWinApp

{

public partial class frmCarDetails : Form

{

public frmCarDetails()

{

InitializeComponent();

}

public ICarRepository CarRepository { get; set; }

public bool InsertOrUpdate { get; set; }

public Car CarInfo { get; set; }

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

{

try

{

var car = new Car

{

CarId = int.Parse(txtCarID.Text),

CarName = txtCarName.Text,

Manufacturer = cboManufacturer.Text,

Price = decimal.Parse(txtPrice.Text),

ReleaseYear = int.Parse(txtReleaseYear.Text)

};

if(InsertOrUpdate == false)

{

CarRepository.InsertCar(car);

}

else

{

CarRepository.UpdateCar(car);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message,InsertOrUpdate==false?"Add a new car": "Update a car");

}

}

private void btnCancel\_Click(object sender, EventArgs e) => Close();

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

{

cboManufacturer.SelectedIndex = 0;

txtCarID.Enabled = !InsertOrUpdate;

if(InsertOrUpdate == true)

{

txtCarID.Text = CarInfo.CarId.ToString();

txtCarName.Text = CarInfo.CarName;

txtPrice.Text = CarInfo.Price.ToString();

txtReleaseYear.Text = CarInfo.ReleaseYear.ToString();

cboManufacturer.Text = CarInfo.Manufacturer.Trim();

}

}

}

}

**frmCarManagement.cs**

using AutomobileLibrary.BiussinessObject;

using AutomobileLibrary.Repository;

namespace AutomobileWinApp

{

public partial class frmCarManagement : Form

{

public frmCarManagement()

{

InitializeComponent();

}

ICarRepository carRepository = new CarRepository();

BindingSource source;

private Action<object, DataGridViewCellEventArgs> dgvCarList\_CellDoubleClick;

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

{

LoadCarList();

}

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

{

frmCarDetails frmCarDetails = new frmCarDetails

{

Text = "Add car",

InsertOrUpdate = false,

CarRepository = carRepository,

};

if(frmCarDetails.ShowDialog() == DialogResult.OK)

{

LoadCarList();

source.Position = source.Position - 1;

}

}

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

{

try

{

var car = GetCarObject();

carRepository.DeleteCar(car.CarId);

LoadCarList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message,"delete a car");

}

}

private void btnClose\_Click(object sender, EventArgs e) => Close();

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

{

btnDelete.Enabled = false;

dgvCarList\_CellDoubleClick += DgvCarList\_CellDoubleClick;

}

private void DgvCarList\_CellDoubleClick(object arg1, DataGridViewCellEventArgs e)

{

frmCarDetails frmCarDetails = new frmCarDetails {

Text = "Update car",

InsertOrUpdate = true,

CarInfo = GetCarObject(),

CarRepository = carRepository,

};

if(frmCarDetails.ShowDialog() == DialogResult.OK)

{

LoadCarList();

source.Position = source.Count -1;

}

}

private void LoadCarList()

{

var cars = carRepository.GetCars();

try

{

source = new BindingSource();

source.DataSource = cars;

txtCarID.DataBindings.Clear();

txtCarName.DataBindings.Clear();

txtManufacturer.DataBindings.Clear();

txtPrice.DataBindings.Clear();

txtReleaseYear.DataBindings.Clear();

txtCarID.DataBindings.Add("text", source, "CarID");

txtCarName.DataBindings.Add("text", source, "Carname");

txtManufacturer.DataBindings.Add("text", source, "Manufacturer");

txtPrice.DataBindings.Add("text", source, "Price");

txtReleaseYear.DataBindings.Add("text", source, "ReleaseYear");

dgvCarList.DataSource = null;

dgvCarList.DataSource = source;

if(cars.Count() ==0){

ClearText();

btnDelete.Enabled = false;

}

else

{

btnDelete.Enabled = true;

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Load Car list");

}

}

private void ClearText()

{

txtCarID.Text = string.Empty;

txtManufacturer.Text = string.Empty;

txtCarName.Text = string.Empty;

txtReleaseYear.Text = string.Empty;

txtPrice.Text = string.Empty;

}

private Car GetCarObject()

{

Car car = null;

try

{

car = new Car

{

CarId = int.Parse(txtCarID.Text),

CarName = txtCarID.Text,

Manufacturer = txtManufacturer.Text,

Price = decimal.Parse(txtPrice.Text),

ReleaseYear = int.Parse(txtReleaseYear.Text)

};

}

catch (Exception ex)

{

MessageBox.Show(ex.Message,"Get car");

}

return car;

}

}

}





