**21110324 - Lương Đăng Khôi**

Màn hình menu:

A screenshot of a computer

Description automatically generated

Chức năng các option:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Source Code:

using System;

using System.Globalization;

using System.Collections.Generic;

using AttributeData;

using System.Runtime.InteropServices;

using System.Threading;

delegate void MilkActionDelegate();

namespace MilkManagement

{

using MainData;

class Program

{

[DllImport("User32.dll", CharSet = CharSet.Auto)]

public static extern int MessageBox(int hWnd, string text, string caption, int options);

static List<Milk> milkList = new List<Milk>

{

new Milk("Dairy Pure", "01/10/2023", "01/04/2024", 100),

new Milk("Organic Valley", "15/08/2023", "15/02/2024", 150),

new Milk("Horizon Organic", "05/09/2023", "05/03/2024", 120)

};

static void Main(string[] args)

{

bool isRunning = true;

while (isRunning)

{

// Menu options

Console.WriteLine("==========0==========");

Console.WriteLine("Menu:");

Console.WriteLine("1. Add Milk");

Console.WriteLine("2. Delete Milk");

Console.WriteLine("3. Display Milk");

Console.WriteLine("4. Clear Terminal");

Console.WriteLine("5. Exit Program");

Console.Write("Choose an option: ");

string choice = Console.ReadLine();

switch (choice)

{

case "1":

AddMilk();

break;

case "2":

DeleteMilk();

break;

case "3":

// Display current Milk objects

Console.WriteLine("Current Milk Data:");

foreach (var milk in milkList)

{

milk.MilkInfoOutput();

}

break;

case "4":

ClearTerminal();

break;

case "5":

// Exit the program

isRunning = false;

break;

default:

Console.WriteLine("Invalid option. Please try again.");

isRunning = false;

break;

}

}

}

public static void AddMilk()

{

var newMilk = new Milk();

newMilk.MilkInfoInput();

milkList.Add(newMilk);

Console.WriteLine("Adding milk...");

Thread.Sleep(1);

Console.WriteLine("Milk item added successfully.\n");

}

public static void DeleteMilk()

{

Console.Write("Enter the Milk ID or Name to delete:");

string input = Console.ReadLine();

// Tìm đối tượng Milk có ID hoặc Tên khớp với input

Milk milkToDelete = milkList.Find(m => m.ValMilkID.Equals(input, StringComparison.OrdinalIgnoreCase) ||

m.ValMilkName.Equals(input, StringComparison.OrdinalIgnoreCase));

if (milkToDelete != null)

{

milkList.Remove(milkToDelete);

Console.WriteLine("Deleting milk...");

Thread.Sleep(1);

Console.WriteLine("Milk successfully deleted.");

}

else

{

Console.WriteLine("Milk not found.");

}

}

public static void ClearTerminal()

{

Console.Clear();

}

}

}

namespace MainData

{

//public delegate void MilkActionDelegate();

interface IMilkAction

{

void MilkInfoInput();

void MilkInfoOutput();

}

public class Milk : IMilkAction

{

private string MilkName;

private string MilkID;

private DateTime ProductionDate;

private DateTime ExpiredDate;

private int Quantity;

//public MilkActionDelegate InputDelegate;

//public MilkActionDelegate OutputDelegate;

public Milk(string milkName = "N/A", string productionDate = "01/01/1010", string expiredDate = "01/01/1010", int quantity = 0)

{

this.MilkName = milkName;

this.ProductionDate = DateTime.ParseExact(productionDate, "dd/MM/yyyy", CultureInfo.InvariantCulture);

this.ExpiredDate = DateTime.ParseExact(expiredDate, "dd/MM/yyyy", CultureInfo.InvariantCulture);

this.MilkID = String.Format("MILK{0}", this.ProductionDate.ToString("ddMMyyyy"));

this.Quantity = quantity;

MilkActionDelegate InputDelegate = new MilkActionDelegate(MilkInfoInput);

MilkActionDelegate OutputDelegate = new MilkActionDelegate(MilkInfoOutput);

//InputDelegate = MilkInfoInput;

//OutputDelegate = MilkInfoOutput;

}

public void MilkInfoInput()

{

Console.WriteLine("Enter Milk Name: ");

ValMilkName = Console.ReadLine();

Console.WriteLine("Enter Production Date (dd/MM/yyyy): ");

ValProductionDate = Console.ReadLine();

Console.WriteLine("Enter Expired Date (dd/MM/yyyy): ");

ValExpiredDate = Console.ReadLine();

Console.WriteLine("Enter Quantity: ");

ValQuantity = int.Parse(Console.ReadLine());

}

public void MilkInfoOutput()

{

string output = $"Milk ID: {ValMilkID}\n" +

$"Milk Name: {ValMilkName}\n" +

$"Production Date: {ValProductionDate}\n" +

$"Expired Date: {ValExpiredDate}\n" +

$"Quantity: {ValQuantity}\n";

Console.WriteLine(output);

}

public string ValMilkID => MilkID; // this method get only

public string ValMilkName

{

get { return MilkName; }

set { MilkName = value; }

}

public string ValProductionDate

{

get { return ProductionDate.ToString("dd/MM/yyyy"); }

set

{

ProductionDate = ParseDate(value);

MilkID = String.Format("MILK{0}", ProductionDate.ToString("ddMMyyyy"));

}

}

public string ValExpiredDate

{

get { return ExpiredDate.ToString("dd/MM/yyyy"); }

set { ExpiredDate = ParseDate(value); }

}

public int ValQuantity

{

get { return Quantity; }

set { Quantity = value; }

}

private DateTime ParseDate(string dateStr)

{

DateTime result;

string[] formats = { "dd/MM/yyyy" };

if (DateTime.TryParseExact(dateStr, formats, CultureInfo.InvariantCulture, DateTimeStyles.None, out result))

{

return result;

}

else

{

throw new FormatException($"'{dateStr}' is invalid.");

}

}

}

}

namespace AttributeData

{

[AttributeUsage(AttributeTargets.All, AllowMultiple = true)]

public class MilkMoreInfo : System.Attribute

{

public string Manufacturer { get; set; }

public string CompanyName { get; set; }

public MilkMoreInfo(string manuFacturer = "", string companyName = "")

{

this.Manufacturer = manuFacturer;

this.CompanyName = companyName;

}

}

}