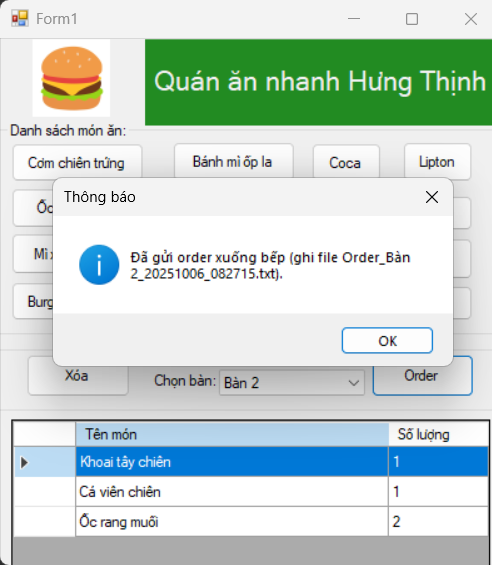
Nguyễn Trần Hữu Đức 11\_ĐH\_CNPM1 MSSV: 1150080049

Môn: Phương pháp phát triển phần mềm hướng đối tượng

Bài tập tuần 6







Source code

using System;

using System.Collections.Generic;

using System.IO;

using System.Windows.Forms;

namespace NguyenTranHuuDuc\_1150080049\_BTtuan6

{

public partial class Form1 : Form

{

// Danh sách món đã chọn (tên món -> số lượng)

private Dictionary<string, int> orderList = new Dictionary<string, int>();

public Form1()

{

InitializeComponent();

// đảm bảo Form1\_Load được gọi (tránh trường hợp Designer chưa gán)

this.Load += Form1\_Load;

}

private void Form1\_Load(object sender, EventArgs e)

{

// kiểm tra control cơ bản có tồn tại không

if (cbBan == null || dgvOrder == null)

{

MessageBox.Show("Không tìm thấy cbBan hoặc dgvOrder. Vào Designer kiểm tra tên (Name) của các control.", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

// Thêm bàn vào ComboBox

cbBan.Items.Clear();

cbBan.Items.AddRange(new string[] { "Bàn 1", "Bàn 2", "Bàn 3", "Bàn 4" });

if (cbBan.Items.Count > 0) cbBan.SelectedIndex = 0;

// Cấu hình DataGridView (tạo cột)

dgvOrder.SuspendLayout();

dgvOrder.Columns.Clear();

var colTenMon = new DataGridViewTextBoxColumn

{

Name = "TenMon",

HeaderText = "Tên món",

AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill,

ReadOnly = true

};

var colSoLuong = new DataGridViewTextBoxColumn

{

Name = "SoLuong",

HeaderText = "Số lượng",

Width = 80,

ReadOnly = true

};

dgvOrder.Columns.Add(colTenMon);

dgvOrder.Columns.Add(colSoLuong);

dgvOrder.AllowUserToAddRows = false;

dgvOrder.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

// programmatic adding vẫn hoạt động tốt; giữ ReadOnly = false để an toàn nếu cần chỉnh chương trình

dgvOrder.ReadOnly = false;

dgvOrder.ResumeLayout();

// Gán sự kiện cho các Button món ăn (an toàn: nếu button không tồn tại thì bỏ qua)

AttachButtonHandler("btnComChien", (s, ev) => AddItem("Cơm chiên trứng"));

AttachButtonHandler("btnBanhMy", (s, ev) => AddItem("Bánh mì ốp la"));

AttachButtonHandler("btnCoca", (s, ev) => AddItem("Coca"));

AttachButtonHandler("btnLipton", (s, ev) => AddItem("Lipton"));

AttachButtonHandler("btnOcRang", (s, ev) => AddItem("Ốc rang muối"));

AttachButtonHandler("btnKhoaiTay", (s, ev) => AddItem("Khoai tây chiên"));

AttachButtonHandler("btn7Up", (s, ev) => AddItem("7 Up"));

AttachButtonHandler("btnCam", (s, ev) => AddItem("Cam"));

AttachButtonHandler("btnMyXao", (s, ev) => AddItem("Mỳ xào hải sản"));

AttachButtonHandler("btnCaVien", (s, ev) => AddItem("Cá viên chiên"));

AttachButtonHandler("btnPepsi", (s, ev) => AddItem("Pepsi"));

AttachButtonHandler("btnCafe", (s, ev) => AddItem("Cafe"));

AttachButtonHandler("btnBurger", (s, ev) => AddItem("Burger bò nướng"));

AttachButtonHandler("btnDuiGa", (s, ev) => AddItem("Đùi gà rán"));

AttachButtonHandler("btnBunBo", (s, ev) => AddItem("Bún bò Huế"));

// Nút chức năng

AttachButtonHandler("btnXoa", BtnXoa\_Click);

AttachButtonHandler("btnOrder", BtnOrder\_Click);

}

// Gán event cho Button an toàn (tìm control theo tên trong tất cả các container)

private void AttachButtonHandler(string controlName, EventHandler handler)

{

var found = this.Controls.Find(controlName, true);

if (found.Length == 0) return; // nếu không thấy, bỏ qua (tránh crash)

if (found[0] is Button btn)

{

// loại bỏ handler trùng lặp (an toàn khi load lại)

btn.Click -= handler;

btn.Click += handler;

}

}

// Hàm thêm món

private void AddItem(string tenMon)

{

// debug tạm: bỏ dòng MessageBox sau khi chắc chắn hoạt động

// MessageBox.Show("AddItem: " + tenMon);

if (orderList.ContainsKey(tenMon))

orderList[tenMon]++;

else

orderList[tenMon] = 1;

LoadDataToGrid();

}

// Hiển thị danh sách món trên DataGridView

private void LoadDataToGrid()

{

if (dgvOrder == null) return;

dgvOrder.Rows.Clear();

foreach (var item in orderList)

{

dgvOrder.Rows.Add(item.Key, item.Value);

}

}

// Xóa món đang chọn

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

{

if (dgvOrder == null) return;

if (dgvOrder.SelectedRows.Count > 0)

{

var cell = dgvOrder.SelectedRows[0].Cells["TenMon"].Value;

if (cell != null)

{

string tenMon = cell.ToString();

if (orderList.ContainsKey(tenMon))

{

orderList.Remove(tenMon);

LoadDataToGrid();

}

}

}

else

{

MessageBox.Show("Vui lòng chọn 1 món để xóa.", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

// Ghi đơn hàng xuống file txt

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

{

if (orderList.Count == 0)

{

MessageBox.Show("Chưa chọn món nào!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

return;

}

string ban = cbBan.SelectedItem?.ToString() ?? cbBan.Text ?? "Unknown";

string fileName = $"Order\_{ban}\_{DateTime.Now:yyyyMMdd\_HHmmss}.txt";

try

{

using (StreamWriter sw = new StreamWriter(fileName))

{

sw.WriteLine($"Bàn: {ban}");

sw.WriteLine("Danh sách món ăn:");

foreach (var item in orderList)

{

sw.WriteLine($"{item.Key} - {item.Value}");

}

}

MessageBox.Show($"Đã gửi order xuống bếp (ghi file {fileName}).", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

orderList.Clear();

LoadDataToGrid();

}

catch (Exception ex)

{

MessageBox.Show("Lỗi khi ghi file: " + ex.Message, "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}