

ĐỀ BÀI — ỨNG DỤNG QUẢN LÝ THỰC ĐƠN (C# Console, List, anonymous type, dynamic)

Mục tiêu

- Luyện **List** + **anonymous type** + `` (không tạo class dữ liệu).
- Vận dụng LINQ cơ bản: `FindIndex`, `FindAll`, `OrderBy`, `OrderByDescending` (nếu cần).
- Thao tác I/O console, kiểm tra nhập liệu an toàn.

I) Mô tả & phạm vi

Viết ứng dụng **Console C#** quản lý **MENU MÓN ĂN**: thêm món, hiển thị, xoá món, **cập nhật tên/giá theo mã, sắp xếp hiển thị theo tên/giá**, và thoát chương trình.

II) Cấu trúc dữ liệu

- Danh sách món: `List<object>`
- Mỗi phần tử là **anonymous object**: `new { ma = int, tenMon = string, gia = int }`
- Truy cập thuộc tính bằng **dynamic**: `((dynamic)x).ma`, `((dynamic)x).tenMon`, `((dynamic)x).gia`

```
var dsMon = new List<object>();  
// Thêm mẫu: dsMon.Add(new { ma = 1, tenMon = "cơm chiên", gia = 1000 });  
// Truy cập: var item = (dynamic)dsMon[0]; Console.WriteLine($"{item.ma} - {item.tenMon} - {item.gia}");
```

III) Giao diện & menu chức năng

```
1/ Thêm món ăn  
2/ Hiển thị menu  
3/ Xoá món ăn  
4/ Cập nhật món ăn (đổi TÊN và/hoặc GIÁ theo MÃ)  
5/ Sắp xếp & hiển thị theo TÊN (tăng dần)  
6/ Sắp xếp & hiển thị theo GIÁ (tăng dần)  
7/ Thoát
```

IV) Yêu cầu chức năng (chi tiết)

(1) Thêm món ăn

- Nhập **MÃ** (*int*) – **không trùng**.
- Nhập **TÊN MÓN** (*string*) – **không trùng tên** (không phân biệt hoa/thường).
- Nhập **GIÁ** (*int*) – **> 0**.
- Hợp lệ thì thêm: `dsMon.Add(new { ma, tenMon, gia });` nếu sai, thông báo lỗi (nêu rõ lý do).

Nhập **MÃ**: 1
Nhập **TÊN MÓN**: cơm chiên
Nhập **GIÁ**: 1000
=> Thêm thành công

(2) Hiển thị menu

- Nếu rỗng: in "Menu đang trống".
- Nếu có dữ liệu: in theo định dạng `i/ [ma] tenMon - gia` (i bắt đầu từ 1).

1/ [1] cơm chiên - 1000
2/ [2] cá chiên - 2000

(3) Xoá món ăn

- Hiển thị danh sách, yêu cầu nhập **CHỈ SỐ i** cần xoá.
- Hợp lệ khi `1 <= i <= dsMon.Count`.
- Hợp lệ => `RemoveAt(i-1)`; sai => báo lỗi, không xoá.

Chọn món ăn xoá (**CHỈ SỐ**): 2
=> Xoá thành công

(4) Cập nhật món ăn (theo **MÃ**)

- Nhập **MÃ** cần cập nhật => tìm bằng `FindIndex((dynamic x) => x.ma == ma)`.
- Nếu **không tìm thấy**: in "Không tìm thấy món tương ứng".
- Nếu **tìm thấy**: cho nhập **TÊN mới** (bỏ trống = giữ nguyên) và **GIÁ mới** (bỏ trống = giữ nguyên).
- Nếu nhập **TÊN mới** => không trùng tên món khác (không phân biệt hoa/thường).
- Nếu nhập **GIÁ mới** => phải là số nguyên dương `> 0`.
- Cập nhật bằng cách thay phần tử tại `index = new { ma = maCu, tenMon = tenMoiHoacCu, gia = giaMoiHoacCu }`.

Nhập **MÃ** cần cập nhật: 1
TÊN mới (bỏ trống nếu giữ nguyên): cơm chiên dương châu

GIÁ mới (bỏ trống nếu giữ nguyên): 1200
=> Cập nhật thành công

(5) Sắp xếp & hiển thị theo TÊN

- Tạo danh sách đã sắp xếp:

```
var ds = dsMon.OrderBy((dynamic x) => x.tenMon).ToList();
```

- In theo định dạng `i/ [ma] tenMon - gia`.

Chọn chức năng: 5
=> Danh sách (theo TÊN):
1/ [2] cá chiên - 2000
2/ [1] cơm chiên - 1000

(6) Sắp xếp & hiển thị theo GIÁ

- Tạo danh sách đã sắp xếp: `var ds = dsMon.OrderBy((dynamic x) => x.gia).ToList();`

- In theo định dạng `i/ [ma] tenMon - gia`.

Chọn chức năng: 6
=> Danh sách (theo GIÁ):
1/ [1] cơm chiên - 1000
2/ [2] cá chiên - 2000

(7) Thoát

- Người dùng nhập **7** để kết thúc vòng lặp; in lời chào và dừng chương trình.

Chọn chức năng: 7
=> Tạm biệt!

V) Kỹ thuật & kiểm tra nhập liệu

- **Số an toàn:** dùng `int.TryParse` cho **MÃ, GIÁ, CHỈ SỐ** để tránh crash; yêu cầu nhập lại khi sai.
- **Chuẩn hoá tên:** `ten = ten.Trim()`; so sánh tên bằng `ToLowerInvariant()`.
- **Kiểm tra trùng:**
- **MÃ:** `FindIndex((dynamic x) => x.ma == maMoi) == -1`
- **TÊN:** `FindIndex((dynamic x) => x.tenMon.ToLowerInvariant() == tenMoi.ToLowerInvariant()) == -1`
- **Cập nhật:** nếu người dùng để trống **TÊN/GIÁ** => giữ nguyên giá trị cũ.

VI) Chữ ký hàm gọi ý (*static class* Method)

```
public static void HienThiMenuChucNang();  
public static void ThemMonAn(ref List<object> dsMon);  
public static void HienThiMenu(List<object> dsMon);  
public static void XoaMonAn(ref List<object> dsMon);  
public static void CapNhatMonAn(ref List<object> dsMon);  
public static void SapXepTheoTen(List<object> dsMon);      // in theo tên (tăng  
dần)\npublic static void SapXepTheoGia(List<object> dsMon);      // in theo giá  
(tăng dần)
```

VII) Ví dụ tương tác (console)

```
1/ Thêm món ăn  
2/ Hiển thị menu  
3/ Xoá món ăn  
4/ Cập nhật món ăn (đổi TÊN/GIÁ theo MÃ)  
5/ Sắp xếp & hiển thị theo TÊN  
6/ Sắp xếp & hiển thị theo GIÁ  
7/ Thoát  
Chọn chức năng: 1  
Nhập MÃ: 1  
Nhập TÊN MÓN: cơm chiên  
Nhập GIÁ: 1000  
=> Thêm thành công  
  
Chọn chức năng: 1  
Nhập MÃ: 2  
Nhập TÊN MÓN: cá chiên  
Nhập GIÁ: 2000  
=> Thêm thành công  
  
Chọn chức năng: 5  
=> Danh sách (theo TÊN):  
1/ [2] cá chiên - 2000  
2/ [1] cơm chiên - 1000  
  
Chọn chức năng: 6  
=> Danh sách (theo GIÁ):  
1/ [1] cơm chiên - 1000  
2/ [2] cá chiên - 2000
```

Chọn chức năng: 7
=> Tạm biệt!

VIII) Tiêu chí chấm

- Đủ các chức năng **1–7**, dùng đúng **List + dynamic**.
 - Thêm: **không trùng MÃ/TÊN, GIÁ > 0**; thông báo lỗi rõ ràng.
 - Xóa đúng theo **CHỈ SỐ**; Cập nhật đúng theo **MÃ**; Sắp xếp đúng theo tiêu chí; Hiển thị đúng định dạng.
 - Không crash khi nhập sai (TryParse, kiểm tra chỉ số, validate cập nhật); mã rõ ràng.
-

IX) Mở rộng (tuỳ chọn)

- Tìm kiếm theo từ khoá tên (không phân biệt hoa/thường) bằng `FindAll`.
- Sắp xếp giảm dần (theo tên/giá) hoặc thêm tiêu chí phụ nếu trùng giá.
- Bỏ qua cập nhật nếu cả **TÊN** và **GIÁ** đều để trống.
- Chặn nhập tên chỉ gồm khoảng trắng (Trim trước khi kiểm tra).