BÀI THỰC HÀNH NGÔN NGỮ LẬP TRÌNH C#

(Bài tập cho ứng dụng Console)

BÀI 1: XÂY DỰNG ỨNG DỤNG MÁY TÍNH ĐIỂM TRUNG BÌNH

Mục tiêu:

- Thực hành khai báo và sử dụng các kiểu dữ liệu cơ bản (double, string).
- Nắm vững kỹ thuật nhập/xuất dữ liệu trên Console (Console.WriteLine, Console.ReadLine).
- Thực hiện ép kiểu (parsing) từ string sang kiểu số.
- Áp dụng cấu trúc điều khiển rẽ nhánh if-else if-else để xử lý logic.
- Làm quen với việc định dạng chuỗi đầu ra (string formatting).

Mô tả bài toán:

Viết một ứng dụng Console cho phép người dùng nhập vào điểm của ba môn học: Toán, Văn, Anh. Chương trình sẽ tính điểm trung bình của ba môn này và dựa vào điểm trung bình để xếp loại học lực của sinh viên theo quy tắc sau:

- ĐTB >= 8.0: Giỏi
- 6.5 <= ĐTB < 8.0: Khá
- 5.0 <= ĐTB < 6.5: Trung bình
- ĐTB < 5.0: Yếu

Yêu cầu:

1. Chương trình phải hiển thị lời chào và hướng dẫn người dùng nhập điểm cho từng môn.
2. Sử dụng kiểu double để lưu trữ điểm số nhằm đảm bảo tính chính xác.
3. Sau khi tính toán, hiển thị điểm trung bình (làm tròn đến 2 chữ số thập phân) và xếp loại học lực tương ứng ra màn hình.

Gợi ý thực hiện:

- Sử dụng double.Parse() hoặc Convert.ToDouble() để chuyển đổi chuỗi nhập vào từ Console.ReadLine() sang kiểu double.
- Sử dụng chuỗi nội suy (interpolated string) với định dạng {variable:F2} để làm tròn và hiển thị số thập phân. Ví dụ: Console.WriteLine($"Điem trung binh của ban la: {diemTB:F2}");

Yêu cầu nâng cao (Tùy chọn):

- Sử dụng khối lệnh try-catch để xử lý ngoại lệ FormatException khi người dùng nhập vào không phải là số.
- Mở rộng chương trình để cho phép người dùng nhập vào một danh sách điểm số không giới hạn cho đến khi họ nhập một từ khóa để kết thúc (ví dụ: "done").

BÀI 2: XÂY DỰNG HỆ THỐNG QUẢN LÝ SINH VIÊN CƠ BẢN

Mục tiêu:

- Thiết kế và triển khai một lớp (class) đối tượng (SinhVien).
- Áp dụng nguyên lý đóng gói (Encapsulation) với private fields và public properties.
- Sử dụng cấu trúc dữ liệu List<T> để quản lý một tập hợp các đối tượng.
- Xây dựng một ứng dụng có menu chức năng sử dụng vòng lặp while và cấu trúc switch-case.
- Tách biệt logic thành các phương thức (methods) có chức năng rõ ràng.

Mô tả bài toán:

Viết một ứng dụng Console cho phép thực hiện các chức năng quản lý một danh sách sinh viên, bao gồm:

1. **Thêm mới một sinh viên:** Yêu cầu người dùng nhập Mã số sinh viên, Họ tên, và Điểm trung bình.
2. **Hiển thị danh sách sinh viên:** In ra thông tin của tất cả sinh viên trong danh sách theo một định dạng bảng.
3. **Tìm kiếm sinh viên theo MSSV:** Cho phép người dùng nhập một MSSV và hiển thị thông tin của sinh viên tương ứng nếu tìm thấy.
4. **Thoát chương trình.**

Yêu cầu:

1. Tạo một lớp SinhVien với các thuộc tính: MaSV (string), HoTen (string), DiemTB (double).
2. Sử dụng một List<SinhVien> toàn cục (hoặc được truyền qua lại giữa các phương thức) để lưu trữ danh sách.
3. Chương trình chính sẽ hiển thị một menu lặp lại cho đến khi người dùng chọn

chức năng “Thoat”.

4. Khi thêm mới, cần kiểm tra xem MaSV đã tồn tại trong danh sách hay chưa để đảm bảo tính duy nhất.

Gợi ý thực hiện:

- Xây dựng các phương thức riêng cho từng chức năng: ThemMoiSinhVien(), HienThiDanhSach(), TimKiemSinhVien().
- Để hiển thị danh sách dạng bảng, có thể sử dụng các ký tự |, - và định dạng chuỗi với string.Format() hoặc chuỗi nội suy để căn chỉnh các cột.
- Sử dụng phương thức Find() hoặc một vòng lặp foreach để tìm kiếm sinh viên trong List.

Yêu cầu nâng cao (Tùy chọn):

- Thêm chức năng **“Xóa sinh viên theo MSSV”** và **“Cập nhật thông tin sinh viên theo MSSV”**.
- Sử dụng LINQ (Language-Integrated Query) để thực hiện các thao tác tìm kiếm, sắp xếp (ví dụ: danhSach.FirstOrDefault(sv => sv.MaSV == mssvCanTim)).
-

BÀI 3: NÂNG CẤP HỆ THỐNG QUẢN LÝ SINH VIÊN VỚI LƯU TRỮ FILE

Mục tiêu:

- Làm việc với hệ thống file sử dụng namespace System.IO.
- Thực hiện các thao tác đọc/ghi file văn bản (File.ReadAllLines, File.WriteAllLines).
- Thực hành kỹ thuật chuyển đổi (serialization/deserialization) dữ liệu đối tượng sang định dạng chuỗi (CSV) và ngược lại.
- Triển khai xử lý ngoại lệ một cách toàn diện cho các thao tác với file và dữ liệu.
- Củng cố các kỹ năng về OOP và quản lý collections.

Mô tả bài toán:

Nâng cấp ứng dụng Quản lý Sinh viên từ Bài 2 với các yêu cầu sau:

1. **Tải dữ liệu khi khởi động:** Khi chương trình bắt đầu, nó phải kiểm tra sự tồn tại của một file *data.txt*. Nếu file tồn tại, chương trình sẽ đọc dữ liệu từ file và nạp vào List<SinhVien>.

2. **Lưu dữ liệu khi thoát:** Trước khi kết thúc chương trình (khi người dùng chọn chức năng "**Thoát**"), toàn bộ danh sách sinh viên hiện tại phải được ghi vào file data.txt, ghi đè lên nội dung cũ.

Yêu cầu:

1. Sử dụng định dạng **CSV (Comma-Separated Values)** để lưu trữ dữ liệu trong file data.txt. Mỗi dòng trong file tương ứng với một sinh viên, và các thuộc tính được ngăn cách bởi dấu phẩy.
   - *Ví dụ:* SV001,Nguyen Van A,8.5
2. Trong hàm Main, gọi một phương thức TaiDuLieu() để thực hiện việc nạp dữ liệu từ file.
3. Trong case của chức năng "**Thoát**", gọi một phương thức LuuDuLieu() trước khi kết thúc chương trình.
4. Sử dụng các khối try-catch để xử lý các ngoại lệ tiềm ẩn như FileNotFoundException (khi tải file lần đầu), IOException (lỗi đọc/ghi file), và FormatException (lỗi khi phân tích dữ liệu từ file).

Gợi ý thực hiện:

- Khi đọc file, sử dụng một vòng lặp foreach trên mảng các dòng đọc được từ File.ReadAllLines(). Trong mỗi vòng lặp, sử dụng phương thức string.Split(',') để tách chuỗi thành các thuộc tính.
- Khi ghi file, sử dụng một vòng lặp foreach trên List<SinhVien>. Trong mỗi vòng lặp, tạo một chuỗi CSV từ các thuộc tính của đối tượng sinh viên (ví dụ: string line = $"{sv.MaSV},{sv.HoTen},{sv.DiemTB}";) và thêm vào một List<string>. Cuối cùng, dùng File.WriteAllLines() để ghi toàn bộ danh sách chuỗi này vào file.

Yêu cầu nâng cao (Tùy chọn):

- Thêm chức năng **"Sắp xếp danh sách sinh viên"** theo điểm trung bình giảm dần hoặc theo họ tên (alphabetical) và hiển thị kết quả.
- Sử dụng LINQ để thực hiện các truy vấn phức tạp hơn, ví dụ: "Liệt kê các sinh viên có điểm trung bình trên 8.0 và có họ là 'Nguyen'".

--- Hết ---