## PHT C# [06] - XỬ LÝ LỖI VÀ DỰ ÁN TỔNG HỢP

### 🎯 KIẾN THỨC CỐT LỐI

| **Khái Niệm** | **Ý Nghĩa** | **Ví Dụ** |
| --- | --- | --- |
| **Exception** | Lỗi xảy ra lúc chạy chương trình | DivideByZeroException |
| **try-catch** | Bắt lỗi và xử lý | try { } catch { } |
| **finally** | Code luôn chạy, dù có lỗi hay không | Đóng file, giải phóng tài nguyên |
| **Validation** | Kiểm tra dữ liệu đầu vào | Kiểm tra age > 0 |
| **throw** | Ném ra một exception | throw new Exception("Lỗi"); |

### 📝 DỰ ÁN TỔNG HỢP: HỆ THỐNG QUẢN LÝ SINH VIÊN

**Yêu cầu chức năng:**

✅ 1. Danh sách sinh viên (tối đa 50)

✅ 2. Thêm sinh viên mới (có validation)

✅ 3. Xóa sinh viên theo ID

✅ 4. Cập nhật điểm cho sinh viên

✅ 5. Tính điểm trung bình

✅ 6. Tìm điểm cao nhất / thấp nhất

✅ 7. Tìm sinh viên theo ID

✅ 8. Xử lý lỗi (try-catch)

**TODO 6.1: [TẠO LỚP STUDENT]**

// Lớp Student chứa thông tin và điểm của 1 sinh viên

public class Student

{

public string StudentId { get; set; }

public string Name { get; set; }

public double Score { get; set; }

// Constructor

public Student(string id, string name, double score)

{

if (string.IsNullOrWhiteSpace(id))

                throw new ArgumentException("StudentId không được rỗng");

            if (string.IsNullOrWhiteSpace(name))

                throw new ArgumentException("Name không được rỗng");

            if (score < 0 || score > 10)

                throw new ArgumentException("Score phải từ 0 đến 10");

            StudentId = id;

            Name = name;

            Score = score;

}

// Phương thức in thông tin

public void Display()

{

Console.WriteLine($"ID: {StudentId} | Tên: {Name} | Điểm: {Score}");

}

}

**TODO 6.2: [TẠO LỚP STUDENT MANAGER]**

public class StudentManager

{

private Student[] students = new Student[50];

private int count = 0; // Số lượng sinh viên hiện tại

public void AddStudent(string id, string name, double score)

{

    if (count >= students.Length)

    {

        Console.WriteLine("Danh sách sinh viên đã đầy!");

        return;

    }

    if (FindStudentById(id) != null)

    {

        Console.WriteLine("Sinh viên đã tồn tại!");

        return;

    }

    students[count] = new Student(id, name, score);

    count++;

    Console.WriteLine("Thêm sinh viên thành công!");

}

        // Xóa sinh viên theo ID

        public void RemoveStudent(string id)

        {

            for (int i = 0; i < count; i++)

            {

                if (students[i].StudentId == id)

                {

                    for (int j = i; j < count - 1; j++)

                        students[j] = students[j + 1];

                    students[--count] = null;

                    Console.WriteLine("Xóa sinh viên thành công!");

                    return;

                }

            }

            Console.WriteLine("Không tìm thấy sinh viên!");

        }

        // Cập nhật điểm

        public void UpdateScore(string id, double newScore)

        {

            Student s = FindStudentById(id);

            if (s == null)

            {

                Console.WriteLine("Không tìm thấy sinh viên!");

                return;

            }

            if (newScore < 0 || newScore > 10)

            {

                Console.WriteLine("Điểm phải từ 0 đến 10!");

                return;

            }

            s.Score = newScore;

            Console.WriteLine("Cập nhật điểm thành công!");

        }

        // Tính điểm trung bình

        public double GetAverageScore()

        {

            if (count == 0) return 0;

            double sum = 0;

            for (int i = 0; i < count; i++)

                sum += students[i].Score;

            return sum / count;

        }

        // Tìm điểm cao nhất

        public double GetMaxScore()

        {

            if (count == 0) return 0;

            double max = students[0].Score;

            for (int i = 1; i < count; i++)

                if (students[i].Score > max)

                    max = students[i].Score;

            return max;

        }

        // Tìm sinh viên theo ID

        public Student FindStudentById(string id)

        {

            for (int i = 0; i < count; i++)

                if (students[i].StudentId == id)

                    return students[i];

            return null;

        }

        // In danh sách sinh viên

        public void DisplayAllStudents()

        {

            if (count == 0)

            {

                Console.WriteLine("Danh sách trống!");

                return;

            }

            for (int i = 0; i < count; i++)

                students[i].Display();

}

**}**

**TODO 6.3: [MENU CHÍNH - MAIN PROGRAM]**

using System;

namespace StudentManagementSystem

{

class Program

{

static void Main(string[] args)

{

StudentManager manager = new StudentManager();

bool running = true;

while (running)

{

Console.WriteLine("\n========== MENU ==========");

                Console.WriteLine("1. Thêm sinh viên");

                Console.WriteLine("2. Xóa sinh viên");

                Console.WriteLine("3. Cập nhật điểm");

                Console.WriteLine("4. In danh sách");

                Console.WriteLine("5. Tính điểm trung bình");

                Console.WriteLine("6. Tìm điểm cao nhất");

                Console.WriteLine("7. Tìm sinh viên");

                Console.WriteLine("0. Thoát");

                Console.Write("Chọn: ");

                try

                {

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

                    switch (choice)

                    {

                        case 1:

                            Console.Write("ID: ");

                            string id = Console.ReadLine();

                            Console.Write("Tên: ");

                            string name = Console.ReadLine();

                            Console.Write("Điểm: ");

                            double score = double.Parse(Console.ReadLine());

                            manager.AddStudent(id, name, score);

                            break;

                        case 2:

                            Console.Write("Nhập ID cần xóa: ");

                            manager.RemoveStudent(Console.ReadLine());

                            break;

                        case 3:

                            Console.Write("ID: ");

                            string uid = Console.ReadLine();

                            Console.Write("Điểm mới: ");

                            double newScore = double.Parse(Console.ReadLine());

                            manager.UpdateScore(uid, newScore);

                            break;

                        case 4:

                            manager.DisplayAllStudents();

                            break;

                        case 5:

                            Console.WriteLine("Điểm trung bình: " + manager.GetAverageScore());

                            break;

                        case 6:

                            Console.WriteLine("Điểm cao nhất: " + manager.GetMaxScore());

                            break;

                        case 7:

                            Console.Write("Nhập ID: ");

                            Student s = manager.FindStudentById(Console.ReadLine());

                            if (s != null) s.Display();

                            else Console.WriteLine("Không tìm thấy sinh viên!");

                            break;

                        case 0:

                            running = false;

                            break;

                        default:

                            Console.WriteLine("Lựa chọn không hợp lệ!");

                            break;

                    }

                }

                catch (Exception ex)

                {

                    Console.WriteLine("Lỗi: " + ex.Message);

                }

            }

        }

}

}

}

}

**📸 CHỨNG THỰC DỰ ÁN:**

* Chụp screenshot menu



A screenshot of a computer

Description automatically generated

Cập nhật:

A screen shot of a computer

Description automatically generated

* Chụp screenshot khi có lỗi (input sai), kiểm tra xử lý lỗi
* File source code hoàn chỉnh

### 🤔 CÂU HỎI PHẢN BIỆN**Nếu người dùng nhập dữ liệu sai (ví dụ: điểm là 15), chương trình nên xử lý như thế nào?**

Sẽ báo lỗi

1. \*\*Có thể dùng Listthay vì array không? Ưu điểm gì?\*\*

Có thể dùng List. Ưu điểm: Tự động tăng, an toàn, code gọn hơn

1. **Nếu muốn lưu dữ liệu xuống file sau khi chương trình thoát, làm sao?**

Ghi file text (.txt)

### 💫 KẾT NỐI ĐÁNH GIÁ

**Bước tiếp theo:**

* Nếu hoàn thành được dự án này, bạn đã sẵn sàng cho **ASP.NET Core** Backend
* Phiếu tiếp theo sẽ dạy **advanced OOP** (interfaces, abstract class, LINQ)
* Tiếp theo là **xây dựng Web API** thực tế