

## BÀI TẬP: XÂY DỰNG HỆ THỐNG QUẢN LÝ BÀI KIỂM TRA TRỰC TUYẾN

### YÊU CẦU BÀI TẬP

Xây dựng hệ thống **Online Exam Management System** cho phép tạo và làm bài kiểm tra với nhiều loại câu hỏi khác nhau. Hệ thống phải thể hiện đầy đủ 4 tính chất OOP: **Trừu tượng, Đóng gói, Kế thừa, Đa hình.**

---

### CHỨC NĂNG CHÍNH

#### 1. QUẢN LÝ CÂU HỎI (Áp dụng Tính trừu tượng + Kế thừa + Đa hình)

Hệ thống hỗ trợ **4 loại câu hỏi:**

##### A. Multiple Choice Question (Câu hỏi trắc nghiệm nhiều đáp án đúng)

- Có 4-6 lựa chọn (A, B, C, D, E, F)
- Có thể có 1 hoặc nhiều đáp án đúng
- Sinh viên chọn bằng cách nhập: "A,C" hoặc "B,D,E"
- Điểm: Chỉ được điểm khi chọn đúng **TẤT CẢ** và không chọn thừa

Ví dụ:

Câu hỏi: Ngôn ngữ nào sau đây là hướng đối tượng?

- A. C
- B. Java
- C. Python
- D. Assembly
- E. C#
- F. JavaScript

Đáp án đúng: B,C,E

Sinh viên trả lời: B,C,E → Đúng (10 điểm)

Sinh viên trả lời: B,C → Sai (0 điểm - thiếu)

Sinh viên trả lời: B,C,E,F → Sai (0 điểm - thừa)

##### B. True/False Question (Câu hỏi Đúng/Sai)

- Chỉ có 2 lựa chọn: True hoặc False
- Sinh viên nhập: "T" hoặc "F" (hoặc "True"/"False")
- Có thể có giải thích tại sao đúng/sai

Ví dụ:

Câu hỏi: Interface trong C# có thể chứa thuộc tính (properties). (T/F)

Đáp án: T

Giải thích: Từ C# 8.0, interface có thể có properties với default implementation.

### C. Short Answer Question (Câu hỏi trả lời ngắn)

- Câu trả lời ngắn (1-5 từ)
- Chấp nhận nhiều biến thể đáp án đúng
- Không phân biệt hoa thường (trừ khi yêu cầu)
- Có thể yêu cầu chính xác về số (ví dụ: toán học)

Ví dụ:

Câu hỏi: Từ khóa nào dùng để khai báo biến không thể thay đổi giá trị trong C#?

Đáp án chấp nhận: ["const", "readonly", "Const", "ReadOnly"]

Câu hỏi:  $2^10 = ?$

Đáp án chấp nhận: ["1024"]

### D. Essay Question (Câu hỏi tự luận - Chấm bằng AI)

- Câu trả lời dài (tối thiểu 100 từ)
- Sử dụng **Gemini AI** để chấm điểm
- AI đánh giá theo:
  - Độ chính xác nội dung
  - Các ý chính cần có
  - Cấu trúc, logic trình bày
  - Ngữ pháp, chính tả
- Trả về điểm số + nhận xét chi tiết

---

## 2. QUẢN LÝ BÀI KIỂM TRA (Áp dụng Đa hình)

Lớp Exam cần có:

- string Title - Tiêu đề bài thi
- string Subject - Môn học
- int Duration - Thời gian làm bài (phút)
- DateTime StartTime - Thời gian bắt đầu
- List<Question> Questions - Danh sách câu hỏi (ĐA HÌNH)
- int PassingScore - Điểm đạt (%)

Các phương thức:

- void AddQuestion(Question q) - Thêm câu hỏi

- void RemoveQuestion(int questionId) - Xóa câu hỏi
  - int CalculateTotalScore() - Tính tổng điểm
  - void Start() - Bắt đầu làm bài
  - ExamResult Submit() - Nộp bài và nhận kết quả
- 

### 3. HỆ THỐNG CHẤM ĐIỂM

Lớp ExamResult cần có:

- string StudentName - Tên sinh viên
- int TotalScore - Tổng điểm đạt được
- int MaxScore - Tổng điểm tối đa
- double Percentage - Phần trăm điểm
- bool IsPassed - Đạt/Không đạt
- TimeSpan TimeTaken - Thời gian làm bài
- List<QuestionResult> QuestionResults - Chi tiết từng câu

Lớp QuestionResult:

- int QuestionId
  - string QuestionContent
  - string UserAnswer
  - string CorrectAnswer
  - int Score
  - int MaxScore
  - bool IsCorrect
  - string Feedback (Cho câu tự luận)
- 

### 4. TÍNH NĂNG BỔ SUNG

A. Timer (Đếm ngược thời gian)

- Hiển thị thời gian còn lại khi làm bài
- Tự động nộp bài khi hết thời gian
- Cảnh báo khi còn 5 phút, 1 phút

B. Review Mode (Xem lại bài làm)

- Xem lại toàn bộ câu hỏi
- Hiển thị đáp án đúng/sai của mình
- Xem giải thích cho từng câu (nếu có)

## C. Statistics (Thống kê)

- Số câu đúng/sai theo từng loại câu hỏi
- Điểm trung bình theo môn học
- Thời gian làm bài trung bình
- Tỷ lệ đạt/không đạt

## D. Export Report (Xuất báo cáo)

- Xuất kết quả ra file TXT hoặc CSV
- Format đẹp, dễ đọc
- Bao gồm tất cả thông tin chi tiết

---

## YÊU CẦU KỸ THUẬT

### 1. TÍNH TRÙU TƯỢNG (Abstraction)

csharp

// Tạo lớp trừu tượng Question với:

abstract class Question

{

// Properties chung

public int Id { get; set; }

public string Content { get; set; }

public int Points { get; set; }

public string Difficulty { get; set; } // Easy, Medium, Hard

// Phương thức trừu tượng BẮT BUỘC các lớp con implement

public abstract void Display();

public abstract bool CheckAnswer(string userAnswer);

public abstract void ShowCorrectAnswer();

public abstract string GetQuestionType();

}

### 2. TÍNH KẾ THỪA (Inheritance)

- Các lớp MultipleChoiceQuestion, TrueFalseQuestion, ShortAnswerQuestion, EssayQuestion kế thừa từ Question
- Mỗi lớp con override các phương thức trừu tượng theo logic riêng

### 3. TÍNH ĐA HÌNH (Polymorphism)

csharp

```
// Ví dụ sử dụng đa hình  
List<Question> examQuestions = new List<Question>();  
examQuestions.Add(new MultipleChoiceQuestion(...));  
examQuestions.Add(new TrueFalseQuestion(...));  
examQuestions.Add(new ShortAnswerQuestion(...));  
examQuestions.Add(new EssayQuestion(...));
```

// Đa hình: Gọi cùng phương thức nhưng hành vi khác nhau

```
foreach (var question in examQuestions)  
{  
    question.Display(); // Mỗi loại hiển thị khác nhau  
    string answer = Console.ReadLine();  
    question.CheckAnswer(answer); // Mỗi loại chấm khác nhau  
}
```

#### 4. TÍNH ĐÓNG GÓI (Encapsulation)

- Sử dụng private, protected, public hợp lý
- Dùng Properties với get, set
- Ẩn logic xử lý phức tạp bên trong class

---

## 🔧 TÍCH HỢP GEMINI AI

### Yêu cầu:

1. Tạo class GeminiGradingService
2. Chấm điểm câu tự luận với prompt chi tiết
3. Trả về điểm số (0-max) + feedback cụ thể
4. Xử lý lỗi khi API fail

### Tiêu chí chấm của AI:

csharp

```
- Độ chính xác nội dung: 40%  
- Các ý chính cần có: 30%  
- Cấu trúc, logic: 20%  
- Ngôn ngữ, trình bày: 10%  
```
```

---

## ##💡 GỢI Ý THIẾT KẾ

### ### Cấu trúc thư mục:

---

ExamSystem/

  |—— Models/

  |  |—— Questions/

  |  |  |—— Question.cs (abstract)

  |  |  |—— MultipleChoiceQuestion.cs

  |  |  |—— TrueFalseQuestion.cs

  |  |  |—— ShortAnswerQuestion.cs

  |  |  \—— EssayQuestion.cs

  |  |—— Exam.cs

  |  |—— ExamResult.cs

  |  \—— QuestionResult.cs

  |—— Services/

  |  |—— GeminiGradingService.cs

  |  |—— TimerService.cs

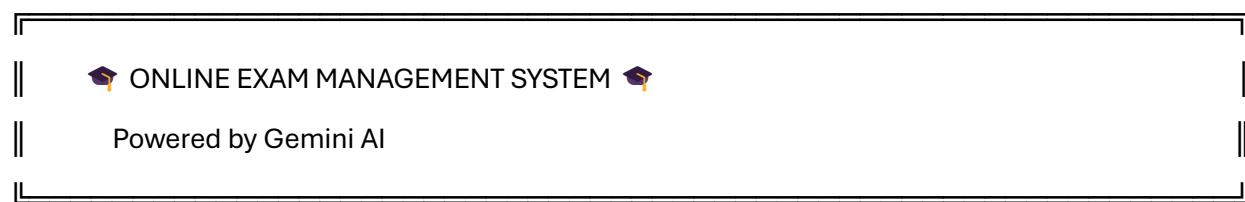
  |  \—— ReportService.cs

  \—— Program.cs

---

### ### Giao diện Console mẫu:

---



Môn: Lập trình hướng đối tượng

Thời gian: 60 phút

Số câu hỏi: 20 câu

Điểm đạt: 70%

 Thời gian còn lại: 58:30

---

Câu 1: [TRẮC NGHIỆM NHIỀU ĐÁP ÁN] - 10 điểm -  HARD

---

Ngôn ngữ nào sau đây hỗ trợ garbage collection?

- A. C
- B. Java
- C. C++
- D. C#
- E. Python
- F. Assembly

Nhập đáp án (VD: A,B,C): \_

---

 **CHỨC NĂNG DEMO PHẢI CÓ**

Trong hàm Main(), tạo một bài kiểm tra mẫu gồm:

- **3 câu Multiple Choice** (1 easy, 1 medium, 1 hard)
- **2 câu True/False**
- **3 câu Short Answer**
- **2 câu Essay** (chấm bằng AI)

Sau khi hoàn thành:

1. Hiển thị kết quả tổng quan
2. Hiển thị chi tiết từng câu
3. Xuất báo cáo ra file
4. Hiển thị thống kê

---

## TIÊU CHÍ ĐÁNH GIÁ

| Tiêu chí | Điểm |
|----------|------|
|----------|------|

|                              |    |
|------------------------------|----|
| Áp dụng đúng 4 tính chất OOP | 25 |
|------------------------------|----|

|                        |    |
|------------------------|----|
| Code sạch, có comments | 10 |
|------------------------|----|

|                               |    |
|-------------------------------|----|
| Tích hợp Gemini AI thành công | 20 |
|-------------------------------|----|

|                       |    |
|-----------------------|----|
| UI/UX đẹp, dễ sử dụng | 10 |
|-----------------------|----|

|               |    |
|---------------|----|
| Xử lý lỗi tốt | 10 |
|---------------|----|

|                           |    |
|---------------------------|----|
| Tính năng Timer hoạt động | 10 |
|---------------------------|----|

|               |   |
|---------------|---|
| Export Report | 5 |
|---------------|---|

|                      |    |
|----------------------|----|
| Tính năng Statistics | 10 |
|----------------------|----|

|             |            |
|-------------|------------|
| <b>TỔNG</b> | <b>100</b> |
|-------------|------------|

---

## THÁCH THỨC NÂNG CAO (Bonus)

Nếu làm được các tính năng sau, được **điểm cộng**:

1. **Save/Load Exam** (+5 điểm)

- Lưu đề thi ra JSON file
- Load đề thi từ file

2. **Shuffle Questions** (+5 điểm)

- Xáo trộn thứ tự câu hỏi
- Xáo trộn thứ tự đáp án (với Multiple Choice)

3. **Question Bank** (+10 điểm)

- Quản lý ngân hàng câu hỏi
- Tạo đề thi ngẫu nhiên từ question bank theo difficulty

4. **Multi-User Support** (+10 điểm)

- Nhiều sinh viên có thể làm bài
- Lưu kết quả từng sinh viên
- So sánh kết quả

5. **Web API Integration** (+15 điểm)

- Tạo REST API cho hệ thống
- Frontend có thể gọi API để làm bài

---

## GỢI Ý HỌC TẬP

Sinh viên nên tìm hiểu:

- Abstract class vs Interface
  - Override vs Overload
  - Polymorphism (compile-time vs runtime)
  - SOLID principles
  - Design patterns: Factory, Strategy
  - Async/Await cho Gemini API
  - Exception handling
  - File I/O trong C#
  - JSON serialization
- 

## THỜI GIAN HOÀN THÀNH

- **Cơ bản (60 điểm):** 3-4 ngày
  - **Đầy đủ (100 điểm):** 1 tuần
  - **Nâng cao (bonus):** 2 tuần
- 

## NỘP BÀI

Sinh viên nộp:

1. Source code (C# project)
2. File README.md hướng dẫn chạy
3. Video demo (3-5 phút)
4. Báo cáo giải thích thiết kế OOP