

ĐÁNH GIÁ CHẤT LƯỢNG KỸ NĂNG NÂNG CAO

MODULE 03 – JAVA CORE – HACKATHON 04

THỜI GIAN: 240 phút

Yêu cầu:

- ☐ Đảm bảo clean code, naming convention trong việc đặt tên các biến, phương thức, lớp
- ☐ Tạo project folder theo hướng dẫn sau:
 - Tạo folder `HN_JV230313_AD_[StudentName]` chứa toàn bộ file mã nguồn
 - Ví dụ: `HN_JV230313_AD_NguyenVanA`
- ☐ Sau khi hoàn thành project, đẩy code lên github và nộp link cho người phụ trách
- ☐ Công nghệ sử dụng: Java
- ☐ IDE: IntelliJ

Thực hành:

Xây dựng ứng dụng Quản lý bài hát có tên Exam_Advance:

1. Xây dựng lớp Singer trong package ra.model gồm các thông tin: **[5 điểm]**

- ☐ Các thuộc tính private :
 - singerId – mã ca sĩ – int (Tự động tăng)
 - singerName – tên ca sĩ – String (Không được để trống)
 - age – tuổi – int (Phải lớn hơn 0)
 - nationality – quốc tịch – String (không được để trống)
 - gender - giới tính - boolean
 - genre - thể loại - String (Không được để trống)
- ☐ Các constructor
- ☐ Các phương thức:
 - Các phương thức getter/setter
 - Triển khai phương thức inputData(): Nhập thông tin của ca sĩ từ bàn phím.

- Triển khai phương thức displayData() cho phép hiển thị tất cả thông tin của ca sĩ đó.

2. Xây dựng lớp Song trong package ra.model gồm các thông tin sau: [5 điểm]

□ Các thuộc tính private :

- songId – mã bài hát – String (Có 4 ký tự và bắt đầu bằng kí tự S, không trùng lặp)
- songName – tên bài hát – String (Không được để trống)
- descriptions – mô tả bài hát – String
- singer - ca sĩ - Singer (không được null)
- songWriter - người sáng tác- String (không được để trống)
- createDate- ngày tạo - Date (mặc định là thời gian hệ thống)
- songStatus- trạng thái - boolean

□ Các constructor

□ Các phương thức:

- Các phương thức getter/setter
- Triển khai phương thức inputData():
 - Nhập thông tin sản phẩm từ bàn phím (trừ createDate mặc định là thời gian hệ thống)
 - Thực hiện thị danh sách ca sĩ cho người dùng chọn theo id nhập vào, nếu không có ca sĩ nào thì bắt buộc người dùng thêm ca sĩ trước.
- Triển khai phương thức displayData() cho phép hiển thị tất cả thông tin sản phẩm (trừ đối tượng ca sĩ , thay vào đó chỉ hiển thị tên ca sĩ trình bày)

3. Xây dựng các lớp SingerService và SongService trong package ra.service thực hiện các chức năng thêm sửa xóa,tìm kiếm và validate dữ liệu theo yêu cầu chương trình.[10 điểm]

4. Xây dựng các lớp SongController và SingerController trong package ra.controller để điều hướng chức năng sang service phù hợp. [10 điểm]

5. Xây dựng lớp MusicManagement trong package ra.run có chứa phương thức main() để chạy chương trình bao gồm các menu như sau:

*****MUSIC-MANAGEMENT*****

1. Quản lý ca sĩ [20 điểm]
2. Quản lý bài hát [20 điểm]
3. Tìm kiếm bài hát [25 điểm]
4. Thoát [5 điểm]

Chọn 1: Điều hướng hiển thị menu quản lý ca sĩ , điều hướng các thao tác tương tự như trên :

*****SINGER-MANAGEMENT*****

1. Nhập vào số lượng ca sĩ cần thêm và nhập thông tin cần thêm mới (có validate dữ liệu nhập vào) [10 điểm]
2. Hiện thị danh sách tất cả ca sĩ đã lưu trữ [5 điểm]
3. Thay đổi thông tin ca sĩ theo mã id [10 điểm]
4. Xóa ca sĩ theo mã id (kiểm tra xem nếu ca sĩ có bài hát thì không xóa được) [5 điểm]
5. Thoát

Chọn 2: Điều hướng hiển thị menu quản lý bài hát , điều hướng các thao tác tương tự như trên :

*****SONG-MANAGEMENT*****

1. Nhập vào số lượng bài hát cần thêm và nhập thông tin cần thêm mới (có validate dữ liệu nhập vào) [10 điểm]
2. Hiện thị danh sách tất cả bài hát đã lưu trữ [5 điểm]
3. Thay đổi thông tin bài hát theo mã id [10 điểm]
4. Xóa bài hát theo mã id [5 điểm]
5. Thoát

Chọn 3: Điều hướng hiển thị menu tìm kiếm (tìm kiếm tương đối), điều hướng các thao tác tương tự như trên :

*****SEARCH-MANAGEMENT*****

1. Tìm kiếm bài hát theo tên ca sĩ hoặc thể loại . [5 điểm]
2. Tìm kiếm ca sĩ theo tên hoặc thể loại [5 điểm]
3. Hiện thị danh sách bài hát theo thứ tự tên tăng dần [5 điểm]
4. Hiện thị 10 bài hát được đăng mới nhất [10 điểm]
5. Thoát

Chọn 4: Thoát khỏi chương trình

Lưu ý: Chỉ tính điểm khi chức năng phải thực hiện đúng theo yêu cầu