



BÁO CÁO BÀI TẬP LẬP TRÌNH JAVA

Môn: Lập trình hướng đối tượng

Đề tài: Quản lý Phim chiếu rạp (Cinema Management)

Thành viên nhóm 8: Nguyễn Công Vinh – B24DCCC289

Trương Công Chiến – B24DCCC040

Lê Minh Đạo – B24DCCC052

Phùng Khắc Phúc – B24DCCC228

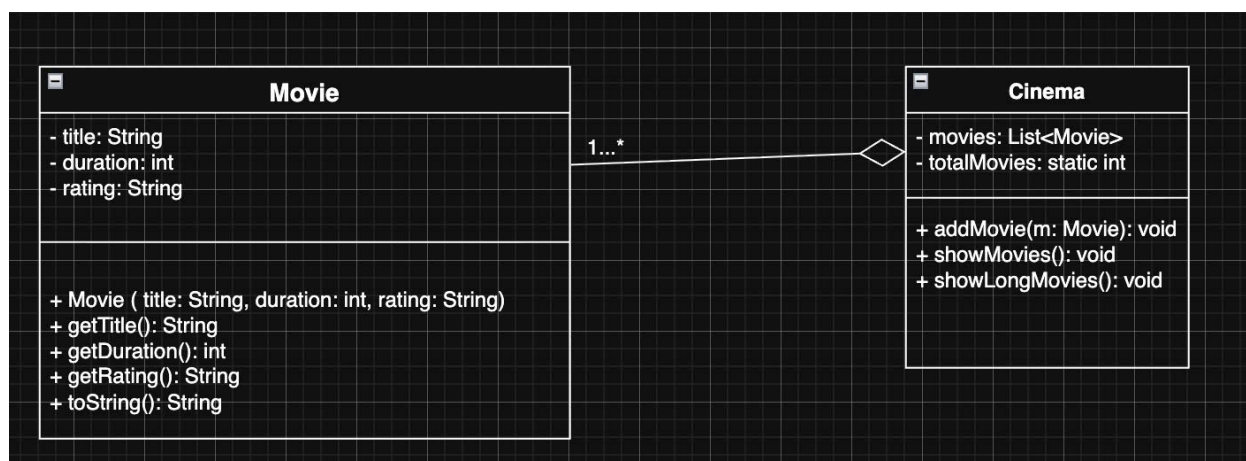
Mục lục

1. Giới thiệu
2. Mô tả các file và lớp
 - Movie.java
 - Cinema.java
 - Main.java
3. Luồng hoạt động của chương trình
4. Kết quả chạy mẫu
5. Đánh giá — Ưu điểm & Hạn chế
6. Gợi ý nâng cấp / Tối ưu
7. Hướng dẫn biên dịch & chạy
8. Tài liệu tham khảo & Ghi chú về mã nguồn

1. Giới thiệu

Dự án **CinemaProject** là một ứng dụng Java nhỏ dùng để quản lý danh sách phim trong một rạp/chuỗi rạp. Mục tiêu chính của chương trình: lưu trữ thông tin phim (title, duration, rating), thêm phim vào rạp, hiển thị danh sách phim và lọc các phim có thời lượng lớn hơn một ngưỡng (ví dụ > 120 phút).

2. Vẽ UML classdiagram



3. Tạo các class với các thuộc tính

3.1. Movie.java

Mục đích: Lớp dữ liệu đại diện cho một bộ phim.

Thuộc tính (fields)

- private String title; — tiêu đề phim.
- private int duration; — thời lượng (phút).
- private String rating; — phân loại (ví dụ: “PG-13”, “All”).

Phương thức

- Getter cho từng thuộc tính.

- toString() — trả về mô tả phim theo định dạng:

Movie [Title: <title>, Duration: <duration> phút, Rating: <rating>].

```
1 package LapTrinhCoBan.CinemaProject;
2
3 public class Movie {
4     private String title;
5     private int duration; // phút
6     private String rating;
7
8     public Movie(String title, int duration, String rating) {
9         this.title = title;
10        this.duration = duration;
11        this.rating = rating;
12    }
13
14    public String getTitle() {
15        return title;
16    }
17
18    public int getDuration() {
19        return duration;
20    }
21
22    public String getRating() {
23        return rating;
24    }
25
26    @Override
27    public String toString() {
28        return "Movie [Title: " + title +
29            ", Duration: " + duration + " phút" +
30            ", Rating: " + rating + "]\n";
31    }
32 }
33
```

3.2. Cinema.java

Mục đích: Quản lý danh sách phim trong rạp.

Thuộc tính

- private List<Movie> movies = new ArrayList<>();
- private static int totalMovies = 0;

Phương thức

- addMovie(Movie m) — thêm phim và tăng biến đếm.
- showMovies() — hiển thị tất cả phim và tổng số phim.
- showLongMovies() — hiển thị phim có thời lượng > 120 phút.

```

1  package LapTrinhCoBan.CinemaProject;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  public class Cinema {
7      private List<Movie> movies = new ArrayList<>();
8      private static int totalMovies = 0; // đếm tổng số phim
9
10     public void addMovie(Movie m) {
11         movies.add(m);
12         totalMovies++;
13     }
14
15     public void showMovies() {
16         System.out.println(x:"Danh sách phim trong rạp:");
17         for (Movie m : movies) {
18             System.out.println(m);
19         }
20         System.out.println("Tổng số phim: " + totalMovies);
21     }
22
23     public void showLongMovies() {
24         System.out.println(x:"Các phim có thời lượng > 120 phút:");
25         for (Movie m : movies) {
26             if (m.getDuration() > 120) {
27                 System.out.println(m);
28             }
29         }
30     }
31 }
32

```

3.3. Main.java

Mục đích: Điểm vào của chương trình.

Tạo Cinema, thêm 3 phim, gọi showMovies() và showLongMovies().

```

1 package LapTrinhCoBan.CinemaProject;
2
3
4 public class Main {
5     public static void main(String[] args) {
6         Cinema cinema = new Cinema();
7
8         Movie m1 = new Movie(title:"Avengers: Endgame", duration:181, rating:"PG-13");
9         Movie m2 = new Movie(title:"Spider-Man: Homecoming", duration:133, rating:"PG-13");
10        Movie m3 = new Movie(title:"Doraemon: Nobita và Mặt trăng phiêu lưu kỳ", duration:111, rating:"All");
11
12        cinema.addMovie(m1);
13        cinema.addMovie(m2);
14        cinema.addMovie(m3);
15
16        cinema.showMovies();
17        System.out.println();
18        cinema.showLongMovies();
19    }
20 }
21

```

4. Luồng hoạt động

1. Tạo Cinema.
2. Tạo các Movie.
3. Thêm phim vào Cinema.
4. Hiện thị tất cả phim.
5. Hiện thị phim dài > 120 phút.

5. Kết quả chạy mẫu

Danh sách phim trong rạp:

Movie [Title: Avengers: Endgame, Duration: 181 phút,
Rating: PG-13]

Movie [Title: Spider-Man: Homecoming, Duration: 133
phút, Rating: PG-13]

Movie [Title: Doraemon: Nobita và Mặt trăng phiêu lưu ký, Duration: 111 phút, Rating: All]

Tổng số phim: 3

Các phim có thời lượng > 120 phút:

Movie [Title: Avengers: Endgame, Duration: 181 phút, Rating: PG-13]

Movie [Title: Spider-Man: Homecoming, Duration: 133 phút, Rating: PG-13]

```
Danh sách phim trong r?p:  
Movie [Title: Avengers: Endgame, Duration: 181 phút, Rating: PG-13]  
Movie [Title: Spider-Man: Homecoming, Duration: 133 phút, Rating: PG-13]  
Movie [Title: Doraemon: Nobita và Mặt trăng phiêu lưu ký, Duration: 111 phút, Rating: All]  
Tổng số phim: 3
```

```
Các phim có thời lượng > 120 phút:  
Movie [Title: Avengers: Endgame, Duration: 181 phút, Rating: PG-13]  
Movie [Title: Spider-Man: Homecoming, Duration: 133 phút, Rating: PG-13]
```

6. Đánh giá

- **Ưu điểm:** đơn giản, dễ hiểu, phù hợp học tập.
- **Hạn chế:** thiếu phương thức xóa/tìm kiếm; totalMovies có thể gây sai lệch; hard-code ngưỡng 120.

7. Cách chạy chương trình

Chạy File Main.java

8. Kết luận

Cấu trúc chương trình tốt cho mục đích học tập; cần bổ sung tính năng và xử lý lỗi nếu triển khai thực tế.