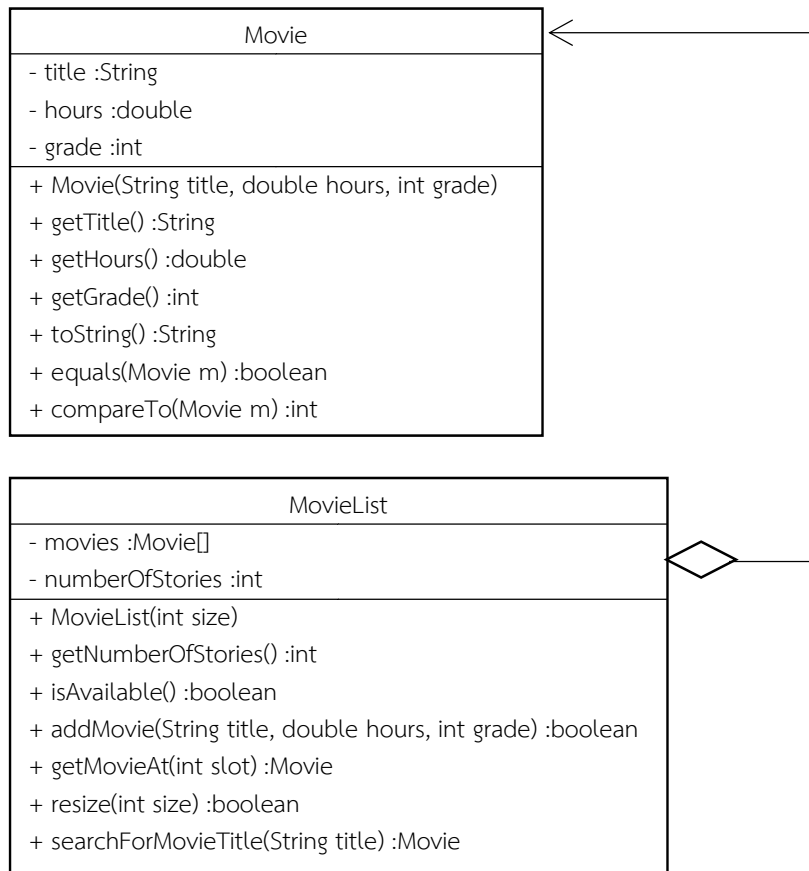


จงเขียนโปรแกรมตามข้อกำหนดใน class diagram โดยทำตามคำสั่งด้านล่างนี้



## 1) Movie Class

- 1.1) **Movie class** ประกอบด้วย (1) ชื่อภาพยนตร์ (**title**) เป็นข้อความ String (2) ความยาวภาพยนตร์เป็นชั่วโมง (**hours**) เป็นจำนวนทศนิยมที่ไม่มีค่าเป็นลบ และ (3) ระดับคุณภาพภาพยนตร์ (**grade**) เป็นจำนวนเต็มจาก 1 ถึง 5 โดย 1 หมายถึง ภาพยนตร์คุณภาพดีมาก และ 5 หมายถึง ภาพยนตร์คุณภาพแย่มาก
- 1.2) **Movie มี 1 constructor** ซึ่งรับชื่อ (title) ความยาว (hours) และระดับคุณภาพ (grade) ของภาพยนตร์เข้ามา เพื่อสร้างเป็น Movie object โดยมีเงื่อนไขการสร้างดังนี้ คือ ถ้า title มีค่าเป็น null จะกำหนดให้ title เป็น "No Title", ถ้า hours มีค่าน้อยกว่าหรือเท่ากับ 0.0 จะกำหนดให้ hours มีค่าเป็น 2.0, ถ้า grade ไม่ได้มีค่าอยู่ในช่วงของ 1 ถึง 5 จะกำหนดให้ grade มีค่าเป็น 3
- 1.3) **Movie มี 3 getters** ได้แก่ getTitle(), getHours(), และ getGrade() แต่ไม่มี setters นอกจากนี้แล้ว Movie ยังมีอีก 3 methods ได้แก่ toString(), equals(), และ compareTo() ซึ่งมีรายละเอียดดังนี้
- 1.4) **toString() method** จะ return ค่าเป็น String ซึ่งประกอบด้วย ชื่อ (title) ความยาว (hours) และระดับคุณภาพ (grade) ของภาพยนตร์ โดย return ในรูปแบบดังนี้  
**Movie: \$ (#.00 hours, grade:#)** เช่น **Movie: The Avengers (2.25 hours, grade:1)**
- 1.5) **equals(Movie m) method** รับอีก movie หนึ่งเข้ามาเปรียบเทียบ โดย (1) จะ return **true** ถ้าภาพยนตร์เรื่องนี้ (this) และภาพยนตร์ที่รับเข้ามา มีชื่อเหมือนกัน มีความยาวเท่ากัน และมีระดับคุณภาพเท่ากัน (เหมือนกันทุกประการ) หรือ (2) method นี้จะ return ค่าเป็น **false** ถ้าภาพยนตร์เรื่องนี้ (this) และภาพยนตร์ที่รับเข้ามา มีชื่อต่างกัน หรือมีความยาวต่างกัน หรือมีระดับคุณภาพต่างกัน หรือภาพยนตร์ที่รับเข้ามา มีค่าเป็น null
- 1.6) **compareTo(Movie m) method** รับอีก movie หนึ่งเข้ามาเปรียบเทียบ โดย (1) จะ return ค่าเป็น **1** ถ้าภาพยนตร์เรื่องนี้ (this) มีระดับคุณภาพ (grade) ดีกว่า (เลขน้อยกว่า) ภาพยนตร์ที่รับเข้ามา หรือภาพยนตร์ที่รับเข้ามา มีค่าเป็น null หรือ (2) จะ return ค่าเป็น **-1** ถ้าภาพยนตร์เรื่องนี้ (this) มีระดับคุณภาพ (grade) แย่กว่า (เลขมากกว่า) ภาพยนตร์ที่รับเข้ามา แต่ (3) ถ้าภาพยนตร์ทั้งสองเรื่องมีระดับคุณภาพ (grade) เท่ากัน จะ return ค่าเป็น **0**

## 2) MovieList Class

- 2.1) **MovieList class** ประกอบด้วย (1) **movies** ซึ่งเป็น array ของ Movie สำหรับเก็บข้อมูลเกี่ยวกับภาพยนตร์ และ (2) **numberOfStories** เป็นจำนวนภาพยนตร์ที่มีอยู่ใน array
- 2.2) **MovieList** มี **1 constructor** ซึ่งรับขนาดความจุ (capacity) เข้ามา เพื่อสร้าง Movie array ที่มีขนาดความจุเท่ากับ capacity ให้กับ movies แต่ถ้า capacity ที่รับเข้ามา มีค่าน้อยกว่าหรือเท่ากับ 0 จะสร้าง Movie array ที่มีขนาดความจุเท่ากับ 10 ให้กับ movies ขนาดความจุของ array นี้บ่งบอกถึงความจุของ MovieList ว่าสามารถใส่ Movie ลงไปได้กี่ครั้งจึงจะเต็ม
- 2.3) **MovieList** มี **1 getter** คือ **getNumberOfStories()** และไม่มี setter ใด ๆ นอกจากนี้แล้ว MovieList ยังมีอีก 5 methods ได้แก่ **isAvailable()**, **addMovie()**, **getMovieAt()**, **resize()**, และ **searchForMovieTitle()** ซึ่งมีรายละเอียดดังนี้
- 2.4) **isAvailable() method** ตรวจสอบว่า MovieList นี้ ยังมีที่ว่างสำหรับบรรจุ Movie อีกหรือไม่ (1) method นี้ return **true** ถ้า MovieList นี้ยังไม่เต็ม หรือ (2) return **false** ถ้า MovieList นี้มี Movie บรรจุอยู่เต็มตามความจุที่รองรับได้แล้ว
- 2.5) **addMovie(String title, double hours, int grade) method** จะ (1) เพิ่ม Movie ที่มีชื่อเป็น **title** ความยาวมีค่าเป็น **hours** และระดับคุณภาพมีค่าเป็น **grade** เข้าไปใน MovieList และนับจำนวน Movie ใน MovieList เพิ่มขึ้นอีก 1 ก่อนจะ return ค่าเป็น **true** แต่ (2) ถ้า MovieList นี้เต็มแล้ว หรือ **hours** ที่รับเข้ามา มีค่าน้อยกว่าหรือเท่ากับ 0 หรือ **grade** ที่รับเข้ามา ไม่ได้มีค่าอยู่ในช่วงของ 1 ถึง 5 จะ return เป็น **false** และไม่เพิ่มภาพยนตร์นี้เข้าไปใน MovieList
- 2.6) **getMovieAt(int slot) method** จะ (1) return **Movie** ที่อยู่ในช่องที่ **slot** ของ MovieList ถ้าช่อง (**slot**) ที่รับเข้ามา มีค่าตั้งแต่ 0 ขึ้นไป และไม่เกินช่องที่มีภาพยนตร์อยู่ มิฉะนั้น (2) method นี้จะ return ค่าเป็น **null**
- 2.7) **resize(int newSize) method** จะ (1) เปลี่ยนขนาดความจุของ MovieList เป็น **newSize** ช่องและ return ค่าเป็น **true** แต่ (2) ถ้าค่า **newSize** ที่รับเข้ามา มีค่าน้อยกว่าหรือเท่ากับจำนวนภาพยนตร์ที่บรรจุอยู่แล้วใน MovieList จะไม่เปลี่ยนแปลงขนาดของ MovieList และจะ return ค่าเป็น **false** (หมายเหตุ: การเปลี่ยนแปลงขนาดความจุของ MovieList สามารถทำได้โดยการสร้าง Movie array ขึ้นใหม่ตามขนาดความจุเท่ากับ newSize ตามที่ต้องการ แล้วย้าย Movie ที่มีอยู่ใน array เดิม ไปใส่ไว้ใน array ใหม่ จากนั้นจึงนำ array ใหม่ไปแทนที่ array เดิมใน MovieList)
- 2.8) **searchForMovieTitle(String title) method** จะ (1) return **Movie** ใน MovieList ที่มีชื่อ **title** ตามที่รับเข้ามา แต่ถ้ามี Movie ที่มีชื่อ **title** ซ้ำกันหลายช่อง ให้ return **Movie** ในช่องแรกสุดที่พบ โดยนับจากตำแหน่ง 0 เป็นช่องแรก แต่ (2) หากว่าไม่มี Movie ในชื่อ **title** หรือ **title** ที่รับเข้ามา มีค่าเป็น **null** ให้ return ค่า **null**

### ตัวอย่างของการเรียกใช้งาน class มีดังนี้

```
Movie a = new Movie("Ant-Man", 1.99, 3);
Movie b = new Movie("Black Panther", 2.25, 2);
System.out.println(a);
System.out.println(a.equals(b));
System.out.println(a.compareTo(b));

MovieList ml = new MovieList(2);
ml.addMovie("Doctor Strange", 1.92, 3);
ml.addMovie("Iron Man", 2.1, 1);
ml.resize(5);
ml.addMovie("The Avengers", 2.33, 1);
Movie m = ml.searchForMovieTitle("Doctor Strange");
if (m != null) {
    System.out.println(m);
}
```

หมายเหตุ: การเปรียบเทียบ String ว่ามี content เหมือนกันหรือไม่นั้น ต้องใช้ **equals** เช่น

```
String a = new String("123");
String b = new String("123");
a.equals(b);
```