11.1.

Gefundene Fehler:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entdecker | Datum | Produkt | QS | Beschreibung | Fehlerklasse | Status |
| Thomas | 14.01.25 | Movie Manager | Code-Inspektion | Die Singleton-Instanz INSTANCE in der Klasse MovieManagerModel wird nicht initialisiert. | 1 | Unbehoben |
| Matthias | 14.01.25 | Movie Manager | Code-Inspektion | Die Methode removeMovie in der Klasse MovieManagerModel ruft movie.getPerformers() auf, was nicht existiert. | 2 | Unbehoben |
| Matthias | 14.01.25 | Movie Manager | Code-Inspektion | Die Methode removePerformer in der Klasse MovieManagerModel ruft performer.getMovies() auf, was nicht existiert. | 2 | Unbehoben |
| Thomas | 14.01.25 | Movie Manger | Code-Inspektion | Die Klasse Movie implementiert nicht das Interface Identifiable, das in getObjectById verwendet wird. | 3 | Unbehoben |

Begründung Fehlerklassen Zuordnung:

1. Da die Singleton-Instanz INSTANCE nicht initialisiert wird, kann die gesamte MovieManagerModel-Klasse nicht verwendet werden. Dies führt zu einem Systemabsturz, da die Methode getInstance() immer null zurückgibt.
2. Die Methode removeMovie versucht, eine nicht existierende Methode getPerformers() aufzurufen. Dies führt dazu, dass die wesentliche Funktion des Entfernens eines Films nicht korrekt ausgeführt wird.
3. Ähnlich wie bei removeMovie versucht die Methode removePerformer, eine nicht existierende Methode getMovies() aufzurufen. Dies beeinträchtigt die wesentliche Funktion des Entfernens eines Darstellers.
4. Die Klasse Movie implementiert nicht das Interface Identifiable, was dazu führt, dass die Methode getObjectById nicht korrekt funktioniert. Dies ist eine funktionale Abweichung, die die Nutzung der Methode einschränkt.

Fehlerbehebung (im Word Dok, da das Projekt nicht Compiliert wegen Gradle Fehler:

1. Singleton-Instanz INSTANCE initialisieren:

private static final MovieManagerModel INSTANCE = new MovieManagerModel();

1. removeMovie Methode korrigieren:

public void removeMovie(Movie movie) {

List<Performer> linkedPerformers = moviePerformerAssociations.get(movie).stream().toList();

linkedPerformers.forEach(performer -> performer.unlink(movie));

moviePerformerAssociations.removeAll(movie);

movies.remove(movie);

}

1. public void removePerformer(Performer performer) {

List<Movie> linkedMovies = moviePerformerAssociations.entries().stream()

.filter(entry -> entry.getValue().equals(performer))

.map(entry -> entry.getKey())

.toList();

linkedMovies.forEach(movie -> movie.unlink(performer));

moviePerformerAssociations.entries().removeIf(entry -> entry.getValue().equals(performer));

performers.remove(performer);

}

1. Movie Klasse Identifiable implementieren:

public class Movie implements Identifiable {

private int id;

// other attributes and methods

@Override

public int getId() {

return id;

}

}