Movie class test strategy

1. **Test Plan for Movie class:**

Test Plan

1. **Create a Movie object with the default constructor.**
2. **Create Movie objects with all non-default constructors.**
3. With valid field values
4. With invalid field values
5. **Test all the get methods.**
6. **Test all the set methods.**
7. With valid arguments
8. With invalid arguments
9. **Test the display method.**
10. **Actual Tests:**

Test 1:

Create a Movie object with the default constructor.

1. **Test Data:**

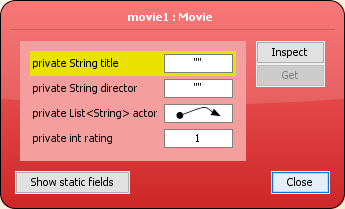
* No input

1. **Expected Results:**

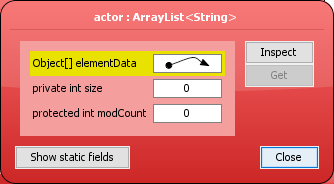
* title: “”
* director: “”
* actor: a null actor list;
* rating: 1

1. **Actual Results:**

Movie inspector:



Actor List:



Test 2(a):

Create a Movie object with the non-default constructor.

1. **Test Data:**

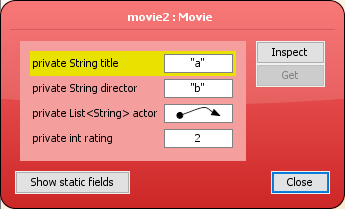
* title: “a”
* director: “b”
* actor 1: “c”
* actor 2: “d”
* actor 3: “e”
* rating: 2

1. **Expected Results:**

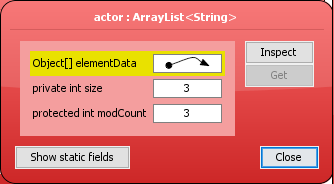
* title: “a”
* director: “b”
* actor 1: “c”
* actor 2: “d”
* actor 3: “e”
* rating: 2

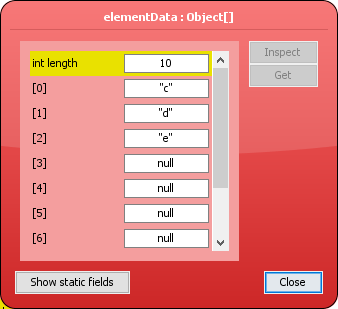
1. **Actual Results:**

Movie inspector:



Actor List:





Test 2(b):

Create a Movie object with invalid attributes(where possible).

1. **Test Data:**

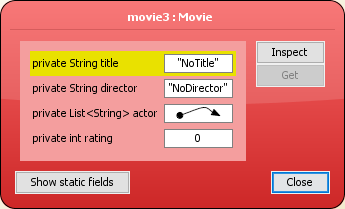
* title: “”
* director: “”
* actor: a null actor list;
* rating: 99

1. **Expected Results:**

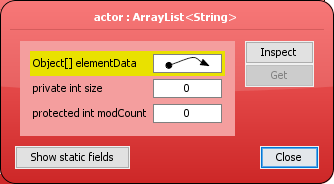
* title: “NoTitle”
* director: “NoDirector”
* actor: a null actor list
* rating: 0

1. **Actual Results:**

Movie inspector:



Actor List:



Test 3(a):

Test the String getTitle() method.

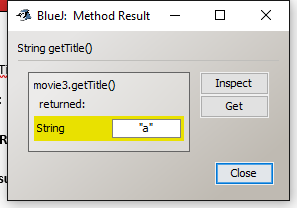
1. **Test Data:**

* title: “a”

1. **Expected Results:**

* title: “a”

1. **Actual Results:**



Test 3(b):

Test the String getDirector() method.

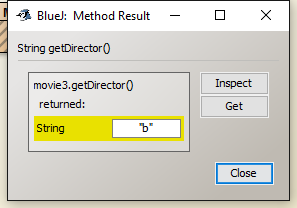
1. **Test Data:**

* director: “b”

1. **Expected Results:**

* director: “b”

1. **Actual Results:**



Test 3(c):

Use the List<String> getActor() method.

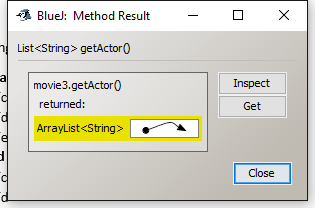
1. **Test Data:**

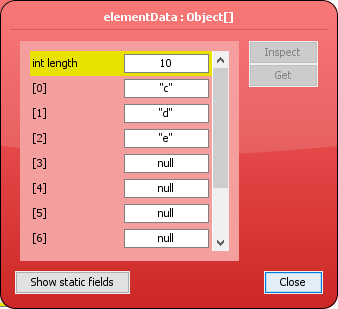
* actor1: “c”
* actor2: “d”
* actor3: “e”

1. **Expected Results:**

* actor1: “c”
* actor2: “d”
* actor3: “e”

1. **Actual Results:**





Test 3(d):

Test the int getRating() method.

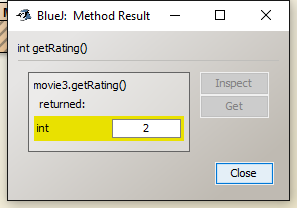
1. **Test Data:**

* rating: 2

1. **Expected Results:**

* rating: 2

1. **Actual Results:**



Test 4(a1):

Test the void setTitle() method with valid argument.

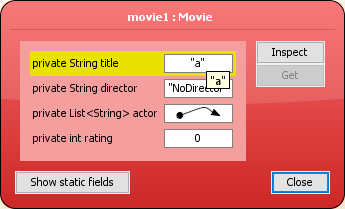
1. **Test Data:**

* title: “a”

1. **Expected Results:**

* title: “a”

1. **Actual Results:**



Test 4(a2):

Test the void setTitle() method with invalid argument.

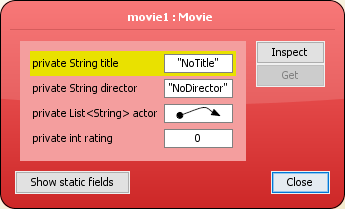
1. **Test Data:**

* title: “”

1. **Expected Results:**

* title: “NoTitle”

1. **Actual Results:**



Test 4(b1):

Test the void setDirector() method with valid argument.

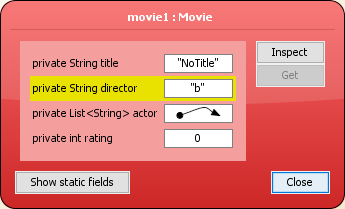
1. **Test Data:**

* director: “b”

1. **Expected Results:**

* director: “b”

1. **Actual Results:**



Test 4(b2):

Test the void setDirector() method with invalid argument.

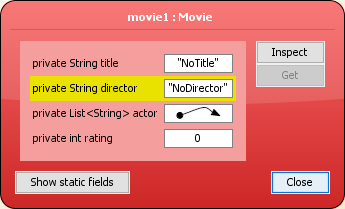
1. **Test Data:**

* director: “”

1. **Expected Results:**

* director: “NoDirector”

1. **Actual Results:**



Test 4(c):

Test the void setActor() method with valid argument.

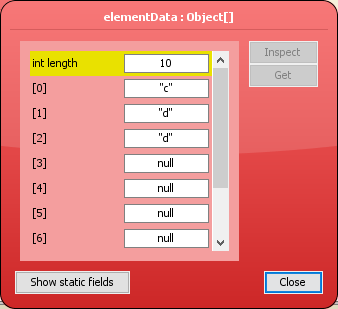
1. **Test Data:**

* actor1: “c”
* actor2: “d”
* actor3: “e”

1. **Expected Results:**

* actor1: “c”
* actor2: “d”
* actor3: “e”

1. **Actual Results:**



Test 4(d1):

Test the void setRatting() method with valid argument.

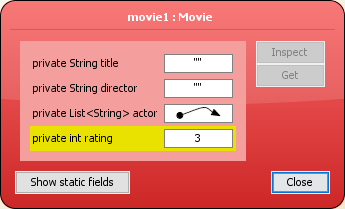
1. **Test Data:**

* rating: 3

1. **Expected Results:**

* rating: 3

1. **Actual Results:**



Test 4(d2):

Test the void setRatting() method with invalid argument.

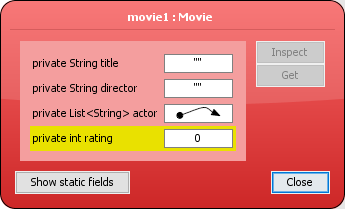
1. **Test Data:**

* rating: 99

1. **Expected Results:**

* rating: 0

1. **Actual Results:**



Test 5:

Test the display() method with valid argument.

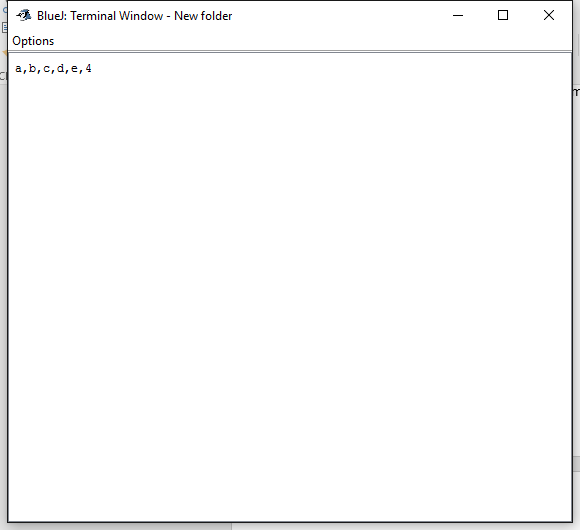
1. **Test Data:**

* title: “a”
* director: “b”
* actor 1: “c”
* actor 2: “d”
* actor 3: “e”
* rating: 4

1. **Expected Results:**

* Display this value: a,b,c,d,e,4

1. **Actual Results:**



All in all, The Movie class works fine.