Player.java

1. Create a Player object with the default constructor.
2. Create a Player object with the non-default constructor:
   * with valid field values
   * with invalid field values
3. Test all get methods:
   * Test getName()
   * Test getPosition()
   * Test getGoal()
   * Test getBehind()
   * Test getSuccessfulPass()
   * Test getKicks()
   * Test ifReported()
   * Test ifInjured()
4. Test all set methods:
   * Test setName()
     + with valid field values
     + with invalid field values
   * Test setPosition()
     + with valid field values
     + with invalid field values
   * Test setGoal()
     + with valid field values
     + with invalid field values
   * Test setBehind()
     + with valid field values
     + with invalid field values
   * Test setSuccessfulPass()
     + with valid field values
     + with invalid field values
   * Test setKicks()
     + with valid field values
     + with invalid field values
   * Test setReported()
     + with valid field values
     + with invalid field values
   * Test setInjured()
     + with valid field values
     + with invalid field values
   * Test checkInjured()
   * Test checkReported()
5. Test all function methods:
   * Test takeAction using different object of Player with different position
6. Test StarPlayer override method:
   * Test getResultBasedOnChance()
     + With different position

**Test 1**

Create an Player object with the default constructor.

Test data:

* name: ""
* position: 'X'
* goal: 0
* behind:0
* successfulPass:0
* kicks:0
* reported:false
* injured:false
* onceReserved:false

Expected results:

* name: ""
* position: 'X'
* goal: 0
* behind:0
* successfulPass:0
* kicks:0
* reported:false
* injured:false
* onceReserved:false

Actual results:

* name: ""
* position: 'X'
* goal: 0
* behind:0
* successfulPass:0
* kicks:0
* reported:false
* injured:false
* onceReserved:false

**Test 2**

Create an Player object with the Non-default constructor.

Test data:

* name: "test"
* position: “Forward”
* goal: 1
* behind:0
* successfulPass:0
* kicks:1
* reported:false
* injured:false
* onceReserved:false

Expected results:

* name: "test"
* position: “Forward”
* goal: 1
* behind:0
* successfulPass:0
* kicks:1
* reported:false
* injured:false
* onceReserved:false

Actual results:

* name: "test"
* position: “Forward”
* goal: 1
* behind:0
* successfulPass:0
* kicks:1
* reported:false
* injured:false
* onceReserved:false

Test3

Create an Player object and use toString.

Test data:

* name: "test"
* position: “Forward”
* goal: 1

Expected results:

test,Forward,1

Actual results:

test,Forward,1

Test4

Create an Player object and use display.

Test data:

* name: "test"
* position: “Forward”
* goal: 1

Expected results:

test,Forward,1

Actual results:

test,Forward,1

Test5

Create an Player object and use getPosition.

Test data:

* name: "test"
* position: “Forward”
* goal: 1

Expected results:

* position: “Forward”

Actual results:

* position: “Forward”

Test 6

Create an Player object and use setPosition with positive expectation.

Test data:

* position: “Midfielder”

Expected results:

* Midfielder

Actual results:

* Midfielder

Test 6

Create an Player object and use setPosition with negative expectation.

Test data:

* position: “invalid”

Expected results:

* exception

Actual results:

exception