**Test Strategy**

1. Test plan

In my Player Class, there are two constructors, two accessor methods, two mutator methods and five other methods for checking player name validation, in which some of those methods will accept parameters and set values for those variables. There are some restrictions on those variables, so that those methods will only execute their functions when the variables are valid. For example, player’s name can only contain alphabetical characters and at most one hyphen, the length of name must be minimum 2 alphabetical characters and with a sensible maximum length, here I set it as 20. Also, the name should not begin or end with a hyphen, and two players in a team cannot have the same name.

1. Test constructors
2. Test all the get methods
3. Test all the set methods
4. Test all the other methods
5. Class diagram

|  |
| --- |
| Player |
| -Name: String  -Goals: int |
| +Player()  +Player(String, int)  +getName(): String  +setName(String)  +getGoal(): int  +setGoal(int)  -isCharacter(String): boolean  -checkLength(String): boolean  -checkHyphen(String): boolean  -checkNameChar(String): boolean  -checkName(String): boolean |

1. Test plan

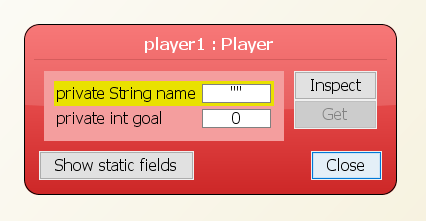
To test this class, I use valid parameters and invalid parameters to respectively observe the outcome, the data I used is below:

|  |  |  |
| --- | --- | --- |
| parameter | Valid data | Invalid data |
| input | “ab” or “abc” or “ab-cd” | “6a” or “^&” or “a” or “ ” or “-a” or “b-” or “--” or “ab-76” or “a--b” or “” |
| name | “abc” |  |
| goals |  |  |

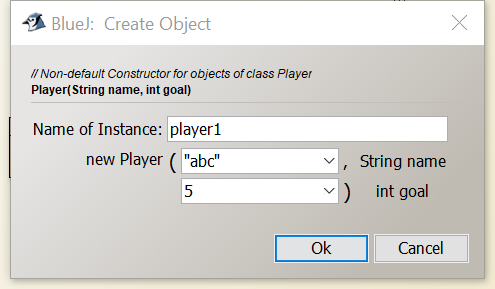
If those valid data are passed as parameters, the method should work correctly, and the values should set or modified as expected or return true, otherwise some error messages would show, and variables would not change as expected or return false.

1. Actual plan
2. Test constructors

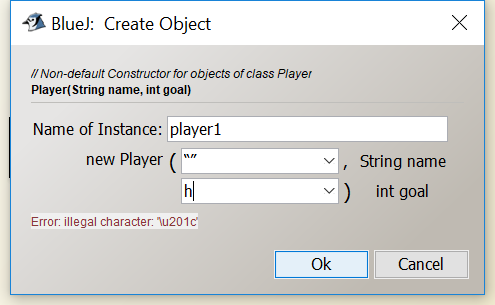
Firstly, use default constructor to initialize an object of Player.



Then, use valid data to initialize non-default constructor, it successfully create an object

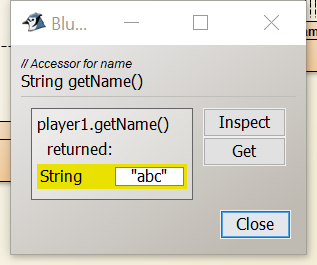


Use invalid data to initialize non-default constructor, it shows error message

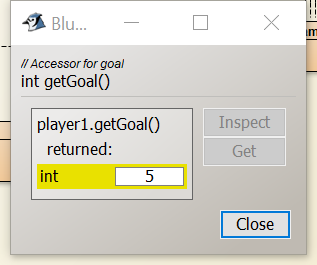


1. Test get methods, both outcomes show correctly

get player name:

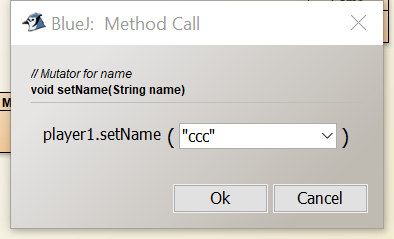
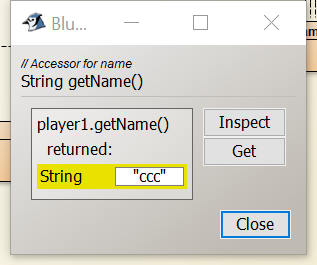


get player goals:

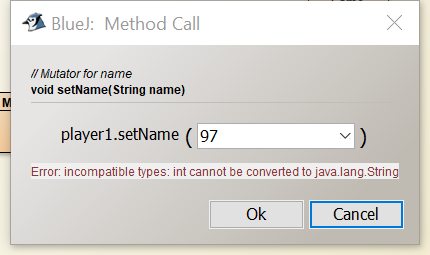
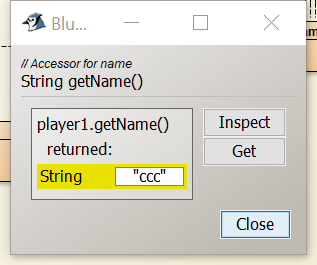


1. Test set methods

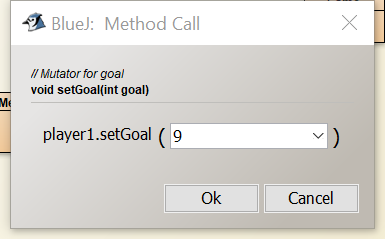
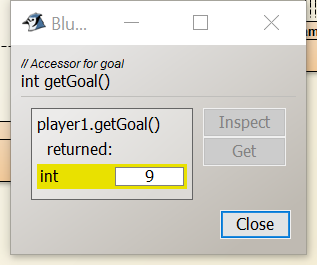
Use valid data to set player name, it is correct

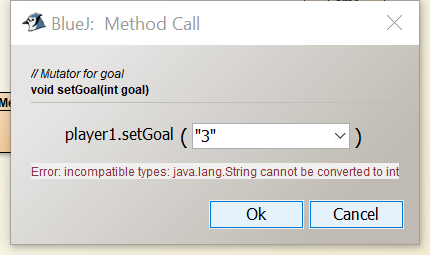
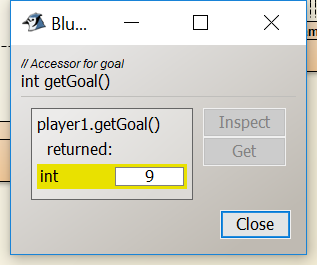
Use invalid data to set player name, shows error and no changes made for name

Use valid data to set player goal, it is correct

Use invalid data to set player goal, it shows error and no changes made for goals

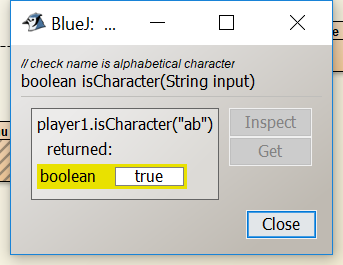
 

1. Test isCharacter(String) method

Use valid data: “ab”

expected result is true

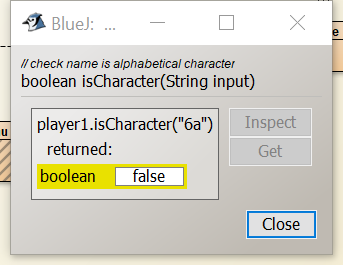
actual result:



Use invalid data: “6a”

Expected result is false

Actual result:

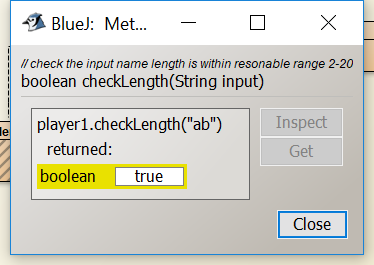


1. Test checkLength(String) method

Use valid data: “ab”

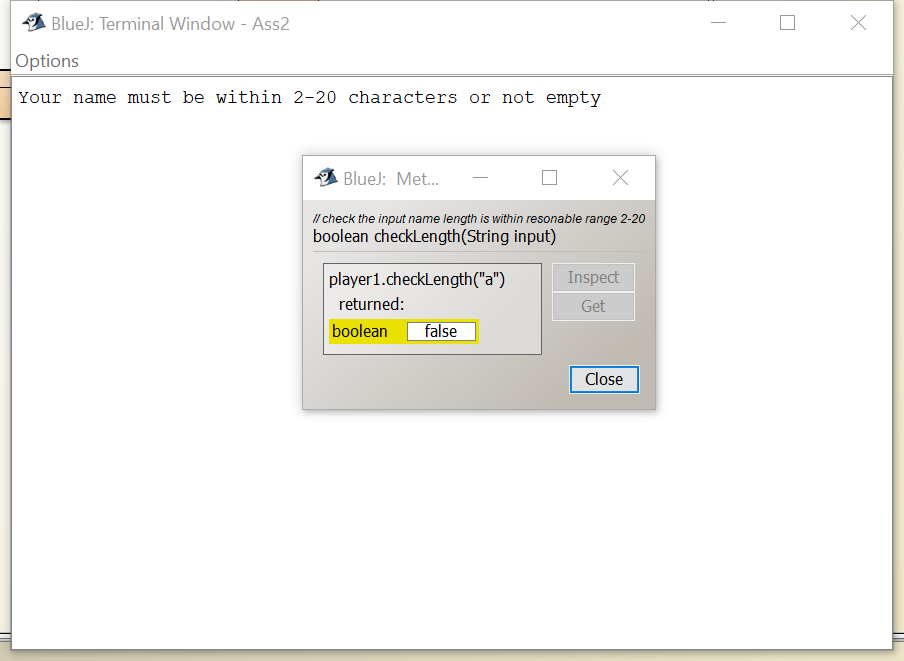
Expected result: true

Actual result:



Use invalid data: “a”

Expected result: false and error message

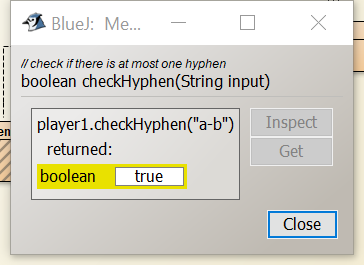
Actual result: 

1. checkHyphen(String) method

use valid data: “a-b”

expected result: true

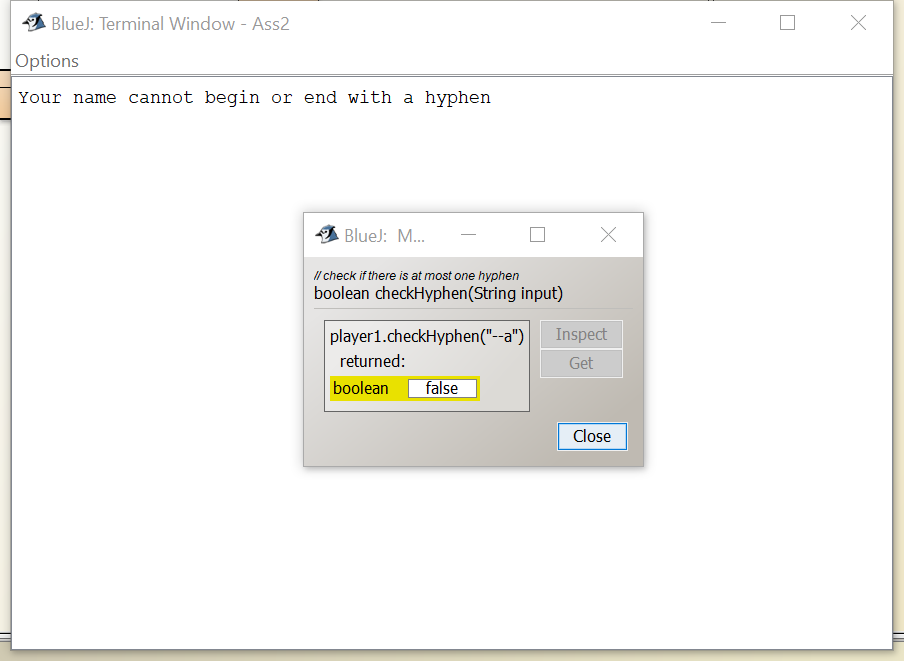
actual result:



use invalid data: “--a”

expected result: false and error message

actual result:

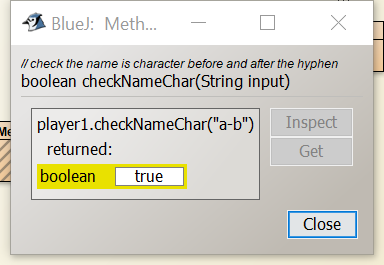


1. checkNameChar(String)

use valid data: “a-b”

expected result: true

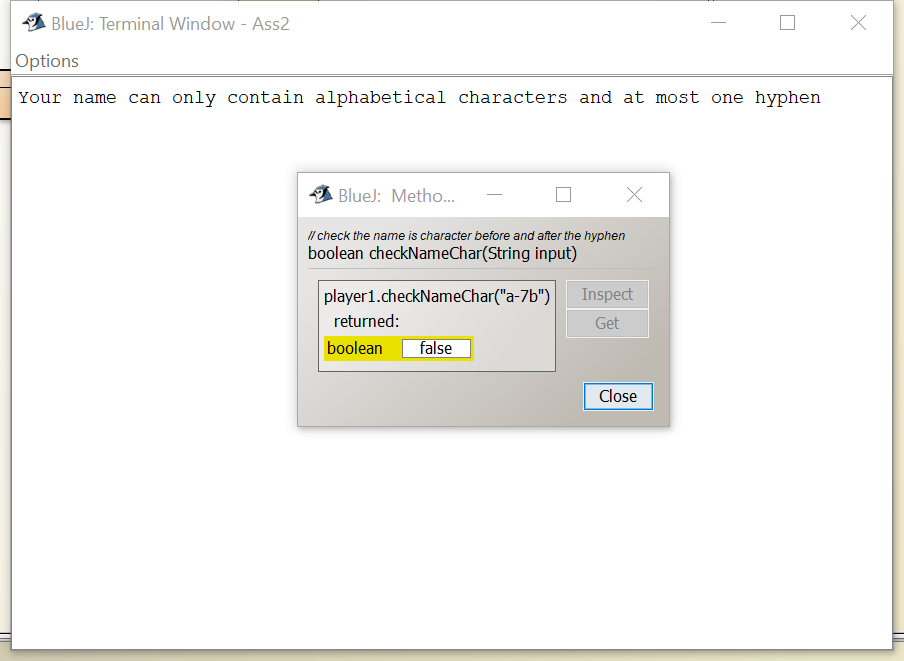
actual result:



use invalid data: “a-7b”

expected result: false and error message

actual result:

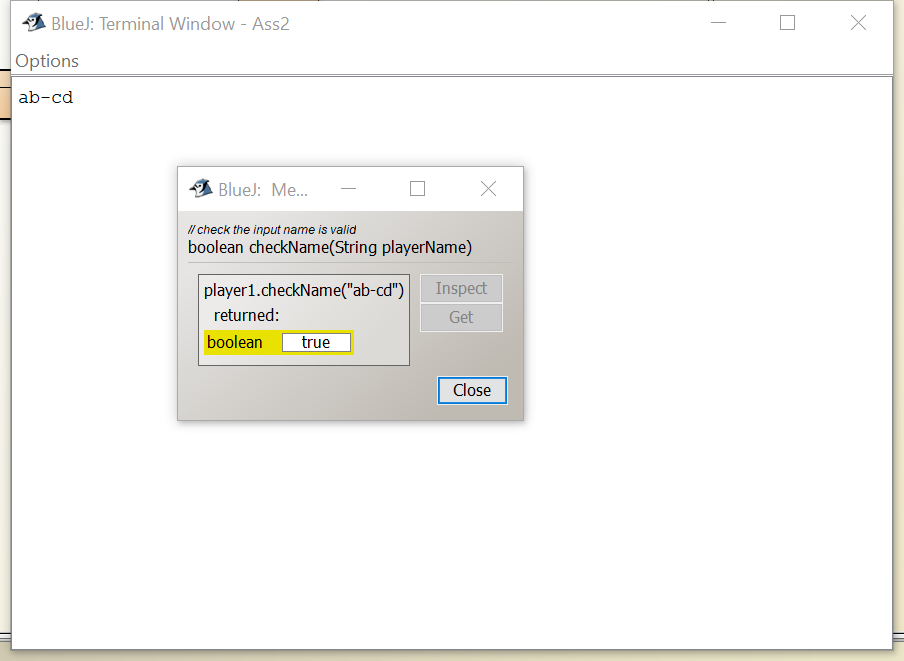


1. checkName(String)

use valid data: “ab-cd”

expected result: true

actual result:



use valid data: “--7b--a” and “89n”

expected result: false and error message

actual result:

