Test Documentation

Diego C.

Table of Contents

[Test #1: Opening an invalid CSV file (list of players) raises OSError 3](#_Toc49416712)

[Test #2: By opening a valid CSV files, the list of players is correctly parsed 4](#_Toc49416713)

[Test #3: Searching by a last name that is not present in the list returns None 5](#_Toc49416714)

[Test #4: Searching by an existing last\_name returns the expected Player 6](#_Toc49416715)

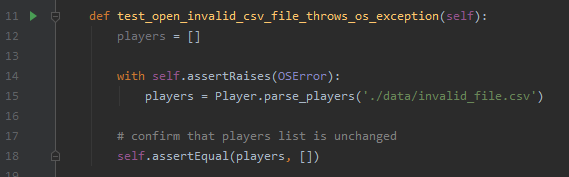
[Test #5: Searching by an existing last\_name from a list of players parsed from a CSV file returns the expected Player 7](#_Toc49416716)

[Test #6: Searching by an existing last\_name that corresponds to multiple players returns all the expected Players 8](#_Toc49416717)

[Test results 9](#_Toc49416718)

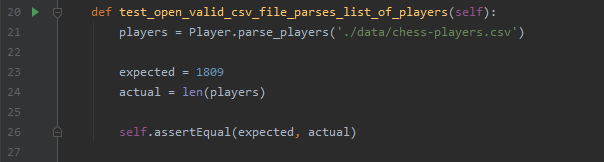
# Test #1: Opening an invalid CSV file (list of players) raises OSError

This test asserts that opening an invalid file (which either does not exist or cannot be accessed) raises an OSError exception.



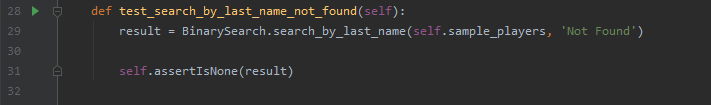
# Test #2: By opening a valid CSV files, the list of players is correctly parsed

In the same vein as the previous unit test, this one asserts whether the full list of players from a valid CSV files is correctly parsed.



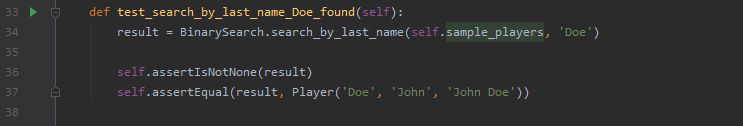
# Test #3: Searching by a last name that is not present in the list returns None

This test asserts whether searching by a last name that does not exist in the list of players via the BinarySearch.search\_by\_last\_name() function returns None.



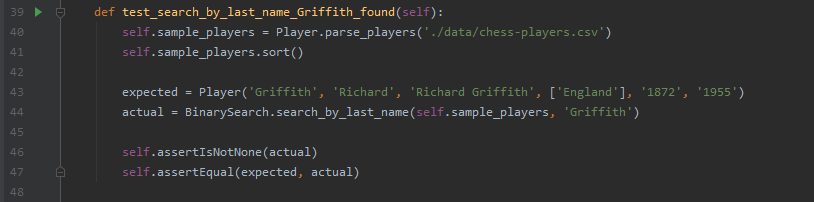
# Test #4: Searching by an existing last\_name returns the expected Player

This test asserts whether searching by a last name that is present in the list of players (taken from a sample list of players – see the setUp() function) via the BinarySearch.search\_by\_last\_name() function returns the respective Player object.



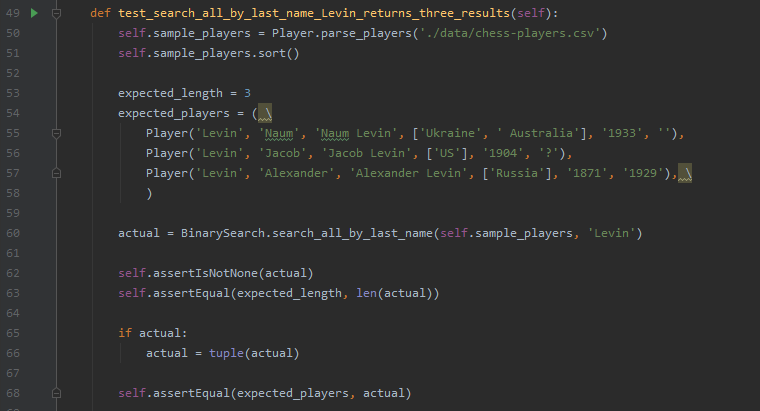
# Test #5: Searching by an existing last\_name from a list of players parsed from a CSV file returns the expected Player

Similarly to the previous test, this one uses the BinarySearch.search\_by\_last\_name to assert whether a Player present in a list of players which is parsed from a CSV file (instead of a sample list of players from the setUp() function) returns the expected Player.



# Test #6: Searching by an existing last\_name that corresponds to multiple players returns all the expected Players

This unit test uses the BinarySearch.search\_all\_by\_last\_name() to assert whether an existing last\_name which points to multiple results returns all the expected Players from the list.



# Test results

