|  |  |  |
| --- | --- | --- |
| Name | Class | Scenary |
| setupStage1 | BST | An empty BST type object is created |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** verify the correct addition of objects to the BST | | | | |
| **Class** | **Method** | **Scenary** | **Input** | **Output** |
| BST | insert | setupScenary  1 | new BST(node: “a” ," key: 3);  new BST(node: “b” ," key: 2);  new BST(node: “c” ," key: 8); | The node “a”,”b” and “c” were correctly added |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** verify the search of the BST | | | | |
| **Class** | **Method** | **Scenary** | **Input** | **Output** |
| BST | search | setupScenary1 | new BST(node: “a” ," key: 3);  new BST(node: “b” ," key: 2);  new BST(node: “c” ," key: 8);  search(key:3) | The value associated with key three is the value "a" |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the value of the successor given an object is correct | | | | |
| **Class** | **Method** | **Scenary** | **Input** | **Output** |
| BST | getSuccessor | setupScenary  1 | new BST(node: “a” ," key: 3);  new BST(node: “b” ," key: 2);  new BST(node: “c” ," key: 8);  getSuccessor(search(key:3)) | The value associated with the successor of key 3 is "c" |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify the minimum value of a BST | | | | |
| **Class** | **Method** | **Scenary** | **Input** | **Output** |
| BST | getMinimum | setupScenary  1 | new BST(node: “a” ," key: 3);  new BST(node: “b” ," key: 2);  new BST(node: “c” ," key: 8);  getMinimum(key: 3) | The key associated with the minimum of the n elements in BST is 2 |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify the maximum value of a BST | | | | |
| **Class** | **Method** | **Scenary** | **Input** | **Output** |
| BST | getMaximum | setupScenary  1 | new BST(node: “a” ," key: 3);  new BST(node: “b” ," key: 2);  new BST(node: “c” ," key: 8);  getMaximum (key: 3) | The key associated with the minimum of the n elements in BST is 8 |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify if an object is leaf or not | | | | |
| **Class** | **Method** | **Scenary** | **Input** | **Output** |
| BST | isLeaf | setupScenary  1 | new BST(node: “a” ," key: 3);  new BST(node: “b” ," key: 4);  new BST(node: “c” ," key: 5);  isLeaf (key: 3)  isLeaf (key: 4)  isLeaf (key: 5) | With key 3 the return value is false, with key 4 the return value is false and with key 5 the return value is true |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify if the BST are empty or not | | | | |
| **Class** | **Method** | **Scenary** | **Input** | **Output** |
| BST | isLeaf | setupScenary  1 | Tree.isEmpty(); | The return value is true |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify the height of the tree | | | | |
| **Class** | **Method** | **Scenary** | **Input** | **Output** |
| BST | bstHeight | setupScenary  1 | new BST(node: “a” ," key: 1);  new BST(node: “b” ," key: 4);  new BST(node: “c” ," key: 5);  tree.bstHeight  new BST(node:”a:”key:6)  tree.bstHeight | The return value is 3  And after of adding a bst object, the return value is 4 |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** verify that a leaf is properly removed | | | | |
| **Class** | **Method** | **Scenary** | **Input** | **Output** |
| BST | delete | setupScenary  1 | new BST(node: “a” ," key: 1);  new BST(node: “b” ," key: 4);  new BST(node: “c” ," key: 5);  new BST(node: “f” ," key: 3);  tree.delete(key:5);  tree.search(key:5);  tree.search(key:4).getRight(); | The return value is null  The return value is null |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify if an object with only one link to another object is correctly removed | | | | |
| **Class** | **Method** | **Scenary** | **Input** | **Output** |
| BST | delete | setupScenary  1 | new BST("a",5);  new BST ("b",7);  new BST ("c",6);  new BST ("d",2);  new BST ("e",4);  new BST ("f",1);  tree.delete(key:7);  tree.search(key:7);  tree.search(5).getRight().getValue(); | The return value is null  The return value is an object with key : 6 |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify if an object with only one link to another object is correctly removed | | | | |
| **Class** | **Method** | **Scenary** | **Input** | **Output** |
| BST | delete | setupScenary  1 | new BST("a",5);  new BST ("b",7);  new BST ("c",6);  new BST ("d",2);  new BST ("e",4);  new BST ("f",1);  new BST(“g”,8);  tree.delete(key:7);  tree.search(key:7);  tree.search(5).getRight(); | The return value is null  The return value is an object with key : 6 |
|  |  |  |  |  |
|  |  |  |  |  |