**Tests Design – Dylan Bermudez Cardona | Victor Manuel Garzón Meneses | Luis Eduardo Charria Meneses**

* **HashtableTest**

|  |  |  |
| --- | --- | --- |
| **Name** | **Class** | **Stage** |
| setupStage1 | HashtableTest | A new hashtable with size=100000 |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the hashtable insert method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| Hashtable | insert() | setUpStage1 | node, node.getKey() | A new node has been inserted in the hashtable (The occupied size is 1) |
| Hashtable | insert() | setupStage1 | node2, node2.getKey()  node, node.getKey()  node3, node3.getKey()  node4, node4.getkey()  node5, node5.getKey() | Five new nodes have been inserted in the hashtable (The occupied size is 5) |
| Hashtable | insert() | setupStage1 | (null) | No nodes have been inserted in the hashtable (The occupied size is 0) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the hashtable search method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| Hashtable | search() | setUpStage1 | “1112388100” | The value found is 20. (The occupied size is 1) |
| Hashtable | search() | setupStage1 | “1112388104” | The value found is 24. (The occupied size is 5) |
| Hashtable | search() | setupStage1 | “1112388100” | No value found (null) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the hashtable delete method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| Hashtable | delete() | setUpStage1 | “1112388100” | The value with the key “1112388100” has been deleted (the occupied size was 1 and now is 0 and when I search that key there is no found value) |
| Hashtable | delete() | setupStage1 | “1112388101”  “1112388103” | The values with the key “1112388101” and “1112388103” have been deleted (the occupied size was 5 and now is 3 and when I search those keys there is no found values) |
| Hashtable | delete() | setupStage1 | “00000” | No value has been deleted. (The occupied size was 1 and now is 1) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the hashtable contains key method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| Hashtable | containsKey() | setUpStage1 | “1112388100” | The hashtable contains the key “1112388100” (the occupied size is 1) |
| Hashtable | containsKey() | setupStage1 | “1112388104” | The hashtable contains the key “1112388104” (the occupied size is 5) |
| Hashtable | containsKey() | setupStage1 | “1112388100” | The hashtable doesn’t contains the key “1112388100” (the occupied size is 0) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the hashtable contains value method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| Hashtable | containsValue() | setUpStage1 | 20 | The hashtable contains the value 20 (the occupied size is 1) |
| Hashtable | containsValue () | setupStage1 | 21 | The hashtable contains the value 21 (the occupied size is 5) |
| Hashtable | containsValue () | setupStage1 | 20 | The hashtable doesn’t contains the value 20 (the occupied size is 0) |

* **QueueTest**

|  |  |  |
| --- | --- | --- |
| **Name** | **Class** | **Stage** |
| setupStage1 | QueueTest | A new queue with size=4 |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the queue offer method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| Queue | offer() | setUpStage1 | node | A new node has been added to the queue (the occupied size is 1 and the node is in the zero position) |
| Queue | offer() | setupStage1 | node  node2  node3  node4 | Four new nodes have been added to the queue and when we try to add one more isn’t possible and is false because the queue has a size of 4. |
| Queue | offer() | setupStage1 | node3  node2  node | Three nodes have been added to the queue (the occupied size is 3) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the queue peek method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| Queue | peek() | setUpStage1 | (null) | The node “discretas” is peeked. (The occupied size was 1 and now is 1) |
| Queue | peek() | setupStage1 | (null) | The node “discretas” is peeked. (The occupied size was 4 and now is 4) |
| Queue | peek() | setupStage1 | (null) | No node is peeked (The occupied size was 0 and now is 0) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the queue poll method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| Queue | poll() | setUpStage1 | (null) | The node2 is extracted from the queue. (The occupied size was 4 and now is 3) |
| Queue | poll() | setupStage1 | (null) | The node is extracted from the queue (The occupied size was 1 and now is 0) |
| Queue | poll() | setupStage1 | (null) | No node is extracted from the queue (The occupied size was 0 and now is 0) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the queue is empty method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| Queue | isEmpty() | setUpStage1 | (null) | The queue isn’t empty (The occupied size was 4 and now is 4) (isEmpty method returns false) |
| Queue | isEmpty() | setupStage1 | (null) | The queue is empty (The occupied size was 0 and now is 0) (isEmpty method returns true) |
| Queue | isEmpty() | setupStage1 | (null) | First, the queue isn’t empty, then the queue is empty (The occupied size was 1 and now is 0) (isEmpty method first returns false and then returns true) |

* **PriorityQueueTest**

|  |  |  |
| --- | --- | --- |
| **Name** | **Class** | **Stage** |
| setupStage1 | PriorityQueueTest | A new priority queue with size=5  A new IPSController variable  A new queue with size=5 |
| setupStage2 | PriorityQueueTest | A new priority queue with size=0  A new IPS controller variable |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the priority queue insert method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| PriorityQueue | insert() | setUpStage1 | new PriorityQueueNode<Patient, Integer>(patient1, partient1.getPriority())  new PriorityQueueNode<Patient, Integer>(patient2, partient2.getPriority())  new PriorityQueueNode<Patient, Integer>(patient3, partient3.getPriority())  new PriorityQueueNode<Patient, Integer>(patient4, partient4.getPriority())  new PriorityQueueNode<Patient, Integer>(patient5, partient5.getPriority()) | Five new nodes have been added to the priority queue (the occupied size is 5 and the maximum is the fifth node) |
| PriorityQueue | insert() | setupStage1  setupStage2 | new PriorityQueueNode<Patient, Integer>(patient1, partient1.getPriority())  new PriorityQueueNode<Patient, Integer>(patient2, partient2.getPriority()) | Two new nodes have been inserted to the priority queue (the occupied size is 2 and the maximum is the third node). Also when we execute the stage 2 we can’t insert a node. |
| PriorityQueue | insert() | setupStage1 | new PriorityQueueNode<Patient, Integer>(patient1, partient1.getPriority())  new PriorityQueueNode<Patient, Integer>(patient2, partient2.getPriority())  new PriorityQueueNode<Patient, Integer>(patient3, partient3.getPriority())  new PriorityQueueNode<Patient, Integer>(patient4, partient4.getPriority())  new PriorityQueueNode<Patient, Integer>(patient5, partient5.getPriority()) | Five new nodes have been added to the priority queue (the occupied size is 5 and the maximum is the fifth node) we delete one and then the occupied size is 4. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the priority queue maximum and minimum methods are running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| PriorityQueue | mínimum()  maximum() | setUpStage2 | (null) | There is no maximum and minimum. |
| PriorityQueue | mínimum()  maximum() | setupStage1 | (null) | Five new nodes have been inserted to the priority queue. The minimum is the node with the priority 1 and the maximum is the node with the priority 24. |
| PriorityQueue | mínimum()  maximum() | setupStage1 | (null) | Five new nodes have been added to the priority queue (the occupied size is 5), the minimum is the node with priority 1 and the maximum is the node with the priority 24. We delete 2 nodes (the occupied size is now 3), the minimum is the node with priority 2 and the maximum is the node with priority 4. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the priority queue extract maximum and extract minimum methods are running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| PriorityQueue | extractMin()  extractMax() | setUpStage1 | (null) | Five new nodes have been inserted to the priority queue. The method extract the maximum (with the id 4) and the last position of the priority queue is now null. Also, the method extract the minimum (with the id 5) and the last position of the priority queue is now null. |
| PriorityQueue | extractMin()  extractMax() | setupStage2 | (null) | There is no maximum and minimum extracted because the priority queue is void. |
| PriorityQueue | extractMin()  extractMax() | setupStage1 | (null) | Five new nodes have been inserted to the priority queue. The method extract the maximum (with the id 3) and the last position of the priority queue is now null. Also, the method extract the minimum (with the id 5) and the last position of the priority queue is now null. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the priority queue increase key and decrease key methods are running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| PriorityQueue | increaseKey()  decreaseKey() | setUpStage1 | patient5, 2  patient3, 1 | Five new nodes have been inserted to the priority queue. The key of the patient5 has been increased to 2 and the key of the patient3 has been decreased to 1. |
| PriorityQueue | increaseKey()  decreaseKey() | setupStage2 | “1111”, 5  “98765”, 1 | There is no increased and decreased key because the priority queue is void. |
| PriorityQueue | increaseKey()  decreaseKey() | setupStage1 | patient5, 0  patient3, 6 | Five new nodes have been inserted to the priority queue. The key of the patient5 can’t be increased to 0 and the key of the patient3 has been increased to 6 (now the maximum) |

* **StackTest**

|  |  |  |
| --- | --- | --- |
| **Name** | **Class** | **Stage** |
| setupStage1 | StackTest | A new stack with dynamic size. |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the stack push method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| Stack | push() | setUpStage1 | action1  action2  action3 | Three new actions have been pushed into the stack. (The size is 3) |
| Stack | push() | setupStage1 | action1  action2  action3 | First, two new actions have been pushed into the stack (The size is 2). After that, one more new action has been pushed into the stack (Now the size is 3). |
| Stack | push() | setupStage1 | action1  action2 | Two new actions have been pushed into the stack (the size is 2) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the stack pop method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| Stack | pop() | setUpStage1 | (null) | Three new actions have been pushed into the stack. (The size is 3), then, we pop and the action3 has been extracted, now the size is 2. |
| Stack | pop() | setupStage1 | (null) | Three new actions have been pushed into the stack (The size is 3), then we pop 3 times, getting the action3, action2 and action1. Now the size is 0 and we pop and we doesn’t get nothing (is null). |
| Stack | pop() | setupStage1 | (null) | Three new actions have been pushed into the stack (The size is 3). We pop 2 times and get action3 and action2 (Now the size is 2) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the stack top method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| Stack | top() | setUpStage1 | (null) | Three new actions have been pushed into the stack. (The size is 3), then, we pop and we get the action3. |
| Stack | top() | setupStage1 | (null) | No actions have been pushed, we pop and get null. |
| Stack | top() | setupStage1 | (null) | Three new actions have been pushed into the stack (The size is 3). We pop 2 times and extract 2 actions, now, we get the top and get action1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test objective:** To know if the stack is empty method is running correctly. | | | | |
| **Class** | **Method** | **Stage** | **Inputs** | **Result** |
| Stack | isEmpty() | setUpStage1 | (null) | Three new actions have been pushed into the stack. (The size is 3), then the stack isn’t empty (isEmpty method returns false) |
| Stack | isEmpty() | setupStage1 | (null) | Three new actions have been pushed into the stack. (The size is 3), we pop 3 times, and the stack now is empt (isEmpty method returns true) |
| Stack | isEmpty() | setupStage1 | (null) | Three new actions have been pushed into the stack (The size is 3). We pop 2 times and the stack isn’t empty (isEmpty method returns false) |