**Test Cases**

**Hash Table**

**Scenario Configuration**

| **Name** | **Class** | **Scenario** |
| --- | --- | --- |
| setupStage1 | HashTableTest | The hashTable is em`pty |
| setupStage2 | HashTableTest | The hashTable is initialized with 1 element.  “1” - > “Juan” |
| setupStage3 | HashTableTest | The size of the has table is 1 and it has 2 element initialized  “1” -> “Juan”  “2” -> Rony |

**Tests:**

| **Test Goal:** Test the correct insertion of elements with the put method | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Result** |
| HashTable | put | setUpStage1 | “1” ‘Juan’ | The hash table should add the element “Juan” with key “1”. |
| HashTable | put | setUpStage1 | “1” “Juan”  “2” “Rony” | The hash Table should contain 2 elements, “Juan” with the key “1” and “Rony” with the key “2” |
| HashTable | put | setUpStage3 | null | Throws an exception |

| **Test Goal:** Test the correct behavior of the get method | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input Value** | **Result** |
| HashTable | Get | setUpStage1 | “1” | Null |
| HashTable | Get | setUpStage2 | “1” | “Juan” |
| HashTable | Get | setUpStage3 | Null | Throws an exception |

| **Test Goal:** Test the remove method | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input Values** | **Result** |
| HashTable | Remove | setupStage1 | Key "1" | The Hash Table remains unchanged, and should return null. |
| HashTable | Remove | setupStage2 | Key "1" | Hash Table turns empty |
| HashTable | Remove | setUpStage3 | Null | Should throw an exception |

| **Test Goal:** Test the correct use of containsKey | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input Values** | **Result** |
| HashTable | ContainsKey | setupStage1 | “1” | False |
| HashTable | ContainKey | setupStage2 | “1” | True |
| HashTable | ContainsKey | setupStage2 | “3” | False |

| **Test Goal:** Test the correct behavior of the is Empty method | | | | |
| --- | --- | --- | --- | --- |

| **Class** | **Method** | **Scenario** | **Result** |
| --- | --- | --- | --- |
| HashTable | isEmpty | setupStage1 | True |
| HashTable | isEmpty | setupStage2 | False |

| **Test Goal:** Test the clear method | | | | |
| --- | --- | --- | --- | --- |

| **Class** | **Method** | **Scenario** | **Result** |
| --- | --- | --- | --- |
| HashTable | Clear | setupStage1 | Hash Table remains empty |
| HashTable | Clear | setupStage2 | Hash Table becomes empty |

**Stack**

| **Name** | **Class** | **Scenario** |
| --- | --- | --- |
| setupStage1 | Stack | The Stack is empty |
| setupStage2 | Stack | Stack contains 1 element “1” |
| setupStage3 | Stack | Stack contains multiple elements.  “1”,”2”,”3” |

| **Test goal:** Test the correct use of the Push method. | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input Values** | **Result** |
| StackTest | Push | setupStage1 | “1” | The stack should contain the element “1” |
| StackTest | Push | setupStage2 | “2” | The stack should contain the element “1” and “2” being “2” the top |
| StackTest | Push | setupStage3 | “3” | The stack should contain the elements “1” , “2” “3”, “3”, being the last element inserted at the top |
| StackTest | Push | setupStage3 | Null | Should Throw and exception |

| **Test goal:**Check the correct popping of elements. | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input Values** | **Result** |
| StackTest | Pop | setupStage1 | “” | Null |
| StackTest | Pop | setupStage2 | “” | “1” and the stack should become empty |
| StackTest | Pop | setupStage3 | “” | “3” and the new stack should be “1” , “2” |

| **Test goal:**Checking the correct operations of peek | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input Values** | **Result** |
| StackTest | Peek | setupStage1 | “” | Null |
| StackTest | Peek | setupStage2 | “” | “1” and the stack should stay the same |
| StackTest | Peek | setupStage3 | “” | “3” and the stack should stay the same |

| **Test goal:** Test the current functioning of the clear function. | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input Values** | **Result** |
| StackTest | clear | setupStage1 | “” | Stack remains empty |
| StackTest | clear | setupStage2 | “” | Stack becomes empty |

| **Test goal:** Test the correct use of isEmpty | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input Values** | **Result** |
| StackTest | IsEmpty | setupStage1 | “” | True |
| StackTest | IsEmpty | setupStage2 | “” | False |

**Queue**

| **Name** | **Class** | **Scenario** |
| --- | --- | --- |
| setupStage1 | Queue | The queue is empty |
| setupStage2 | Queue | The queue is initialized with 3 elements.  “node1”, “node22 , node3” |
| setupStage3 | Queue | The queue is initialized with 15 elements.  "node1", "node2", "node 3", "node4", "node5", "node6", "node7", "node8", "node9", "node10", "node11", "node12", "node13", "node14", "node15" |

| **Test goal:**.Test the enqueue method | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input Values** | **Result** |
| QueueTest | Enqueue | setupStage1 | n | The top of the queue becomes n |
| QueueTest | Enqueue | setupStage2 | n | The top of the queue should be item1 but the queue should also contain the item2 |
| QueueTest | Enqueue | setupStage3 | Null | Throws an exception |

| **Test goal:**.Test the dequeue method | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input Values** | **Result** |
| QueueTest | Dequeue | setupStage1 | “” | node 2 |
| QueueTest | Dequeue | setupStage2 | Using the method dequeue 3 times. | "” |
| QueueTest | Dequeue | setupStage2 | “” | "Node 2" |

| **Test goal:**.Test the front method | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input Values** | **Result** |
| QueueTest | Front | setupStage2 | “” | Node 1 |
| QueueTest | Front | setupStage1 | “” | Should throw a null pointer exception5 |
| QueueTest | Front | setupStage3 | “” | Node 1 |

| **Test goal:**.Test the isEmpty method | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input Values** | **Result** |
| QueueTest | IsEmpty | setupStage1 | “” | True |
| QueueTest | IsEmpty | setupStage2 | “” | False |

**Heap**

**Scenario Configuration**

| **Name** | **Class** | **Scenario** |
| --- | --- | --- |
| setupStage1 | HeapTests | The heap is empty |
| setupStage2 | HeapTests | The heap is initialized with the numbers 1 to 5 in ascending order |
| setupStage3 | HeapTests | The heap is initialized with the numbers 1 to 5 in descending order |
| setupStage4 | HeapTests | The heap is initialized with the numbers 1 to 5 in disorder |

**Tests:**

| **Test Goal:** Test the correct insertion of elements with the add method | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Result** |
| Heap | add | setUpStage1 | “1” | The hash table should add the element “1”. |
| Heap | add | setUpStag21 | “6” | The hash table should add the element “6”. |
| Heap | add | setUpStage3 | “6” | The hash table should add the element “6”. |

| **Test Goal:** Test the correct disposal of elements with the poll method | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Result** |
| Heap | poll | setUpStage2 | “” | The method must return “1” and the heap must not have the “1” |
| Heap | poll | setUpStage3 | “” | The method must return “1” and the heap must not have the “1” |
| Heap | poll | setUpStage4 | “” | The method must return “1” and the heap must not have the “1” |

| **Test Goal:** .Test the isEmpty method | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Result** |
| Heap | isEmpty | setUpStage1 | “” | True |
| Heap | isEmpty | setUpStage2 | “” | True |
| Heap | isEmpty | setUpStage3 | “” | True |

| **Test Goal:** .Test the size method | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Result** |
| Heap | size | setUpStage1 | “” | “0” |
| Heap | size | setUpStage2 | “” | “5” |
| Heap | size | setUpStage3 | “” | “5” |

| **Test Goal:** .Test the clear method | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Result** |
| Heap | clear | setUpStage1 | “” | The heap must be empty |
| Heap | clear | setUpStage2 | “” | The heap must be empty |
| Heap | clear | setUpStage3 | “” | The heap must be empty |

| **Test Goal:** .Test the remove method | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Result** |
| Heap | remove | setUpStage2 | “1” | The heap must not have the “1” and its head should be 2 |
| Heap | remove | setUpStage3 | “5” | The heap must not have the “5” and its head should be 1 |
| Heap | remove | setUpStage4 | “4” | The heap must not have the “4” and its head should be 1 |

| **Test Goal:** .Test the HeapifyUp method, testing that when I add the 0 it remains as the head with differents sceneries | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Result** |
| Heap | HeapifyUp | setUpStage2 | “0” | The heap must be reordered and its head must be 0 |
| Heap | HeapifyUp | setUpStage3 | “0” | The heap must be reordered and its head must be 0 |
| Heap | HeapifyUp | setUpStage4 | “0” | The heap must be reordered and its head must be 0 |

| **Test Goal:T**test the HeapifyDown method, checking that when pulled it still maintains the min heap property. | | | | |
| --- | --- | --- | --- | --- |
| **Class** | **Method** | **Scenario** | **Input values** | **Result** |
| Heap | HeapifyDown | setUpStage2 | “” | The heap must be reordered and its head must be 2 |
| Heap | HeapifyDown | setUpStage3 | “” | The heap must be reordered and its head must be 2 |
| Heap | HeapifyDown | setUpStage4 | “” | The heap must be reordered and its head must be 2 |