**Scenario configuration**

|  |  |  |
| --- | --- | --- |
| Name | Class | Scenario |
| setUpScenaryI () | QueueTest | tail |
| setUpScenary2() | QueueTest | tail |
| setUpScenary3() | QueueTest | tail |

Test Cases Design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Objective | Validate that the method enqueue is correctly working | | | |
| Class | Method | Scenario | Input Values | Result |
| QueueTest | enqueue | setUpScenaryI () | name: “Daniel”  currentFloor: 2  desiredOffice: 2  isOnOffice: false  futureFloor: 3 | The element of type Person is added to the queue, so, head and tail are equals to the node added.  One is added to the size. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Objective | Validate that the method enqueue is correctly working | | | |
| Class | Method | Scenario | Input Values | Result |
| QueueTest | enqueue | setUpScenary2() | name: “Camilo”  currentFloor: 2  desiredOffice: 3  isOnOffice: false  futureFloor: 2 | The new element of type Person is added to the queue, so now, head is pointed to the Person “Daniel” and tail is pointed to the new person added “Camilo”. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Objective | Validate that the method dequeue is correctly working | | | |
| Class | Method | Scenario | Input Values | Result |
| QueueTest | dequeue | setUpScenary3() |  | The element to which the head is pointing (“Daniel”) is taken from the queue, now head will point to the next element on the queue (“Camilo”).  One is subtracted from size. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Objective | Validate that the method dequeue is correctly working | | | |
| Class | Method | Scenario | Input Values | Result |
| QueueTest | dequeue | setUpScenary1() |  | The method will return null, be cause the queue is empty. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Objective | Validate that the method dequeue is correctly working | | | |
| Class | Method | Scenario | Input Values | Result |
| QueueTest | dequeue | setUpScenary2() |  | The element to which the head is pointing(“Daniel”) is taken from the queue, now head will point to null.  One is subtracted from size. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Objective | Validate that the method isEmpty is correctly working | | | |
| Class | Method | Scenario | Input Values | Result |
| QueueTest | isEmpty | setUpScenary1() |  | The method will return true, the queue is empty. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Objective | Validate that the method isEmpty is correctly working | | | |
| Class | Method | Scenario | Input Values | Result |
| QueueTest | isEmpty | setUpScenary2() |  | The method will return false. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Objective | Validate that the method getHead is correctly working | | | |
| Class | Method | Scenario | Input Values | Result |
| QueueTest | getHead | setUpScenary1() |  | The method will return null, the head is pointing to null be cause, the queue is empty. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Objective | Validate that the method getHead is correctly working | | | |
| Class | Method | Scenario | Input Values | Result |
| QueueTest | getHead | setUpScenary2() |  | The method will return the person which is pointed by the head (“Daniel”). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Objective | Validate that the method getHead is correctly working | | | |
| Class | Method | Scenario | Input Values | Result |
| QueueTest | getHead | setUpScenary3() |  | The method will return the person which is pointed by the head (“Daniel”). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Objective | Validate that the method getTail is correctly working | | | |
| Class | Method | Scenario | Input Values | Result |
| QueueTest | getTail | setUpScenary1() |  | The method will return null, the tail is pointing to null be cause, the queue is empty. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Objective | Validate that the method getTail is correctly working | | | |
| Class | Method | Scenario | Input Values | Result |
| QueueTest | getTail | setUpScenary2() |  | The method will return the person which is pointed by the tail and head (“Daniel”). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Objective | Validate that the method getTail is correctly working | | | |
| Class | Method | Scenario | Input Values | Result |
| QueueTest | getTail | setUpScenary3() |  | The method will return the person which is pointed by the tail (“Camilo”), the last person that was added. |