**Scenario and use case format**

# **Scenario configuration HashTable:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Class** | **Scenario** |
| setupStage1 | HashTableTest | A Node<Integer, String> class object with K=1 and V=”persona1”  A Node<Integer, String> class object with K=456 and V=”persona2”  A Node<Integer, String> class object with K=719 and V=”persona3”  A Node<Integer, String> class object with K=3 and V=”persona4” |
| setupStage2 | HashTableTest | A Node<Integer, String> class object with K=1 and V=”persona1”  A Node<Integer, String> class object with K=456 and V=”persona2”  A Node<Integer, String> class object with K=719 and V=”persona3”  A Node<Integer, String> class object with K=3 and V=”persona4”  A Node<Integer, String> class object with K=523 and V=”persona5”  A Node<Integer, String> class object with K=6455 and V=”persona6”  A Node<Integer, String> class object with K=71 and V=”persona7”  A Node<Integer, String> class object with K=309 and V=”persona8”  A Node<Integer, String> class object with K=864 and V=”persona9”  A Node<Integer, String> class object with K=69 and V=”persona10”  A Node<Integer, String> class object with K=4879 and V=”persona11”  A Node<Integer, String> class object with K=305 and V=”persona12”  A Node<Integer, String> class object with K=956 and V=”persona13”  A Node<Integer, String> class object with K=895 and V=”persona14”  A Node<Integer, String> class object with K=425 and V=”persona15”  A Node<Integer, String> class object with K=532 and V=”persona16” |
| setupStage3 | HashTableTest | A Node<Integer, String> class object with K=1 and V=”persona1”  A Node<Integer, String> class object with K=1 and V=”persona2”  A Node<Integer, String> class object with K=1 and V=”persona3”  A Node<Integer, String> class object with K=1 and V=”persona4” |

**Test cases design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that objects that are added and can be retrieved by their id when the table has no collisions, when it has collisions, and when it has objects with the same key | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | getValue & add | setupStage1 | K=1,  K=456,  K=719,  K=3. | V=”persona1”,  V=”persona2”,  V=”persona3”,  V=”persona4”. |
| HashTable | getValue & add | setupStage2 | K=1,  K=456,  K=719,  K=3,  K=523,  K=6455,  K=71,  K=309,  K=864,  K=69,  K=4879,  K=305,  K=956,  K=895,  K=425,  K=532. | V=”persona1”,  V=”persona2”,  V=”persona3”,  V=”persona4”,  V=”persona5”,  V=”persona6”,  V=”persona7”,  V=”persona8”,  V=”persona9”,  V=”persona10”,  V=”persona11”,  V=”persona12”,  V=”persona13”,  V=”persona14”,  V=”persona15”,  V=”persona16”. |
| HashTable | getValue & add | setupStage3 | K=1,  K=1,  K=1,  K=1. | V=”persona1”,  V=”persona1”,  V=”persona1”,  V”persona1”. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verify that objects that are added, can be removed from the table, by their id when the table has no collisions, when it has collisions, and when it has objects with the same key | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | remove | setupStage1 | K=1,  K=456,  K=719,  K=3. | “”,  “”,  “”,  “”, |
| HashTable | remove | setupStage2 | K=1,  K=456,  K=719,  K=1,  K=456,  K=719, | “”,  “”,  “”,  “Value not found to delete”,  “Value not found to delete”  “Value not found to delete”. |
| HashTable | remove y getValue (intercalados) | setupStage3 | K=1,  K=1,  K=1,  K=1.  K=1,  K=1,  K=1,  K=1,  K=1. | “”,  “persona1”  “”,  “persona1”  “”  “persona1”  “”  Null  “Value not found to delete” |

|  |  |  |
| --- | --- | --- |
| **Name** | **Class** | **Scenario** |
| setupStage1 | HashTableTest | A Node<Integer, String> class object with K=1 and V=”persona1”  A Node<Integer, String> class object with K=456 and V=”persona2”  A Node<Integer, String> class object with K=719 and V=”persona3”  A Node<Integer, String> class object with K=3 and V=”persona4” |
| setupStage2 | HashTableTest | A Node<Integer, String> class object with K=1 and V=”persona1”  A Node<Integer, String> class object with K=456 and V=”persona2”  A Node<Integer, String> class object with K=719 and V=”persona3”  A Node<Integer, String> class object with K=3 and V=”persona4”  A Node<Integer, String> class object with K=523 and V=”persona5”  A Node<Integer, String> class object with K=6455 and V=”persona6”  A Node<Integer, String> class object with K=71 and V=”persona7”  A Node<Integer, String> class object with K=309 and V=”persona8”  A Node<Integer, String> class object with K=864 and V=”persona9”  A Node<Integer, String> class object with K=69 and V=”persona10”  A Node<Integer, String> class object with K=4879 and V=”persona11”  A Node<Integer, String> class object with K=305 and V=”persona12”  A Node<Integer, String> class object with K=956 and V=”persona13”  A Node<Integer, String> class object with K=895 and V=”persona14”  A Node<Integer, String> class object with K=425 and V=”persona15”  A Node<Integer, String> class object with K=532 and V=”persona16” |
| setupStage3 | HashTableTest | A Node<Integer, String> class object with K=1 and V=”persona1”  A Node<Integer, String> class object with K=1 and V=”persona2”  A Node<Integer, String> class object with K=1 and V=”persona3”  A Node<Integer, String> class object with K=1 and V=”persona4” |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that objects that are added and can be retrieved by their id when the table has no collisions, when it has collisions, and when it has objects with the same key | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | getValue & add | setupStage1 | K=1,  K=456,  K=719,  K=3. | V=”persona1”,  V=”persona2”,  V=”persona3”,  V=”persona4”. |
| HashTable | getValue & add | setupStage2 | K=1,  K=456,  K=719,  K=3,  K=523,  K=6455,  K=71,  K=309,  K=864,  K=69,  K=4879,  K=305,  K=956,  K=895,  K=425,  K=532. | V=”persona1”,  V=”persona2”,  V=”persona3”,  V=”persona4”,  V=”persona5”,  V=”persona6”,  V=”persona7”,  V=”persona8”,  V=”persona9”,  V=”persona10”,  V=”persona11”,  V=”persona12”,  V=”persona13”,  V=”persona14”,  V=”persona15”,  V=”persona16”. |
| HashTable | getValue & add | setupStage3 | K=1,  K=1,  K=1,  K=1. | V=”persona1”,  V=”persona1”,  V=”persona1”,  V”persona1”. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Verify that objects that are added, can be removed from the table, by their id when the table has no collisions, when it has collisions, and when it has objects with the same key | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | remove | setupStage1 | K=1,  K=456,  K=719,  K=3. | “”,  “”,  “”,  “”, |
| HashTable | remove | setupStage2 | K=1,  K=456,  K=719,  K=1,  K=456,  K=719, | “”,  “”,  “”,  “Value not found to delete”,  “Value not found to delete”  “Value not found to delete”. |
| HashTable | remove y getValue (intercalados) | setupStage3 | K=1,  K=1,  K=1,  K=1.  K=1,  K=1,  K=1,  K=1,  K=1. | “”,  “persona1”  “”,  “persona1”  “”  “persona1”  “”  Null  “Value not found to delete” |

# **Scenario configuration PriorityQueue**

|  |  |  |
| --- | --- | --- |
| **Name** | **Class** | **Scenario** |
| setupStage1 | PriorityQueueTest | An object of the class Passenger with cedula= 1596745256  ticket=”A23”  first\_class=1  arrival\_time=” 08:07:40”  miles\_earned=500  special\_attention=1  elderly=0  number\_of\_suitcases=2  membership\_level=3  medical\_care=1  An object of the class Passenger with cedula= 4567854  ticket=”B23”  first\_class=1  arrival\_time=”07:12:25”  miles\_earned=0  special\_attention=1  elderly=1  number\_of\_suitcases=1  membership\_level=1  medical\_care=0  An object of the class Passenger with cedula= 23549845  ticket=”C23”  first\_class=1  arrival\_time=”06:07:40”  miles\_earned=200  special\_attention=1  elderly=1  number\_of\_suitcases=1  membership\_level=2  medical\_care=1 |
| setupStage2 | PriorityQueueTest | Empty prorityQueue |
| setupStage3 | PriorityQueueTest | An object of the class Passenger with cedula= 1596745256  ticket=”A23”  first\_class=1  arrival\_time=” 08:07:40”  miles\_earned=500  special\_attention=1  elderly=0  number\_of\_suitcases=2  membership\_level=3  medical\_care=1  An object of the class Passenger with cedula= 104285412  ticket=”A23”  first\_class=1  arrival\_time=” 08:07:40”  miles\_earned=500  special\_attention=1  elderly=0  number\_of\_suitcases=2  membership\_level=3  medical\_care=1  An object of the class Passenger with cedula= 21537832  ticket=”A23”  first\_class=1  arrival\_time=” 08:07:40”  miles\_earned=500  special\_attention=1  elderly=0  number\_of\_suitcases=2  membership\_level=3  medical\_care=1 |

**Test cases design**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the remove method can be retrieve objects in the correct order based on their priority, one with normal inputs, the other with an empty queue and last with exact same priority | | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| ProrityQueue | remove | setupStage1 | Remove,  Remove,  Remove. | Passenger with cedula= 1596745256.  Passenger with cedula= 4567854,  Passenger with cedula= 23549845. |
| ProrityQueue | remove | setupStage2 |  | null |
| ProrityQueue | remove | setupStage3 | Remove,  Remove,  Remove. | Passenger with cedula= 1596745256,  Passenger with cedula= 104285412,  Passenger with cedula= 21537832 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the isEmpty method works correctly, with one standard setup, one empty and one at first standard but then mkde it empty. | | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| ProrityQueue | isEmpty | setupStage1 |  | False |
| ProrityQueue | isEmpty | setupStage2 |  | True |
| ProrityQueue | isEmpty | setupStage3 | Remove(),  Remove(),  Remove() | False  True |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the Size method works correctly, with one standard setup, one with an empty queue, and one evaluated after each item removal. | | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| ProrityQueue | size | setupStage1 |  | 3 |
| ProrityQueue | size | setupStage2 |  | 0 |
| ProrityQueue | size | setupStage3 | Size(),  Remove(),  Size(),  Remove(),  Size(),  Remove()  Size(). | 3  2  1  0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Objective:** Verify that the peek method works correctly, with one standard setup, one with an empty queue, and one evaluated after each item removal. | | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| ProrityQueue | peek | setupStage1 |  | Passenger with cedula= 1596745256. |
| ProrityQueue | peek | setupStage2 |  | NoSuchElementException() |
| ProrityQueue | Peek | setupStage3 | peek(),  Remove(),  Size(),  Remove(),  Size(),  Remove()  Size(). | Passenger with cedula= 1596745256.  Passenger with cedula= 4567854,  Passenger with cedula= 23549845.  NoSuchElementException() |