# **Hashtable Test Cases**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpEmptyHashTable | HashTableTest | An empty hashtable. |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpHashTable | HashTableTest | (“Daniel”,1)  (“Andrew”,2)  (“Carlos”,3) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** putTest, probar si el método put agrega adecuadamente objetos dentro de una vacia HashTable. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| HashTableTest | put | setUpEmptyHashTable | (“Capitán Teemo”,666)  (“Capitán Gareb”,667)  (“Capitán Ashe”,668) | El resultado esperado es que el tamaño de la hashtable vaya aumentando a medida que insertamos objetos en ella |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** putTest, probar si el método put agrega adecuadamente objetos dentro del HashTable aun si ya hay objetos dentro de la hashtable. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| HashTableTest | put | setUpHashTable | (“Capitán Teemo”,666)  (“Capitán Gareb”,667)  (“Capitán Ashe”,668) | El resultado esperado es que el tamaño de la hashtable vaya aumentando a medida que insertamos objetos en ella |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** getTest, probar si el método get lidia correctamente con las colisiones en la HashTable. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| HashTableTest | get | setUpHashTable | (“Capitán Teemo”,666)  (“Capitán Gareb”, 667)  (“Capitán Ashe”, 6681) | El resultado esperado es que el método get nos devuelva correctamente los objetos esperados. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** getTest, comprobar el comportamiento adecuado de el método cuando se busca objetos en una Hashtable vacia. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| HashTableTest | get | setUpEmptyHashTable |  | El resultado esperado es que el método get nos devuelva un aviso de que la hashtable esta vacia. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** removeTest, probar si el método remove borra correctamente objetos del HashTable. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| HashTableTest | remove | setUpHashTable | 1,2,3 | El resultado esperado es que el método remove borre correctamente los objetos en la hashtable. |

# **Queue Test Cases**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpEmptyQueue | QueueTest | An empty Queue. |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpQueue | QueueTest | (“Pablo”)  (“Daniel”)  (“Andrea”) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** enqueueTest, probar que le método enqueue coloca objetos en la Queue adecuadamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| QueueTest | Enqueue | setUpEmptyQueue | (“Pablo”)  (“Daniel”)  (“Andrea”) | El resultado esperado debe ser un incremento en le tamaño de la Queue. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** enqueueTest, probar que le método enqueue coloca objetos en la Queue adecuadamente aun si ya hay objetos en la cola. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| QueueTest | Enqueue | setUpQueue | (“Pablo”)  (“Daniel”)  (“Andrea”) | El resultado esperado debe ser un incremento en le tamaño de la Queue. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** dequeueTest, probar que le método dequeue remueve objetos de la queue adecuedamente | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| QueueTest | dequeue | setUpQueue |  | El resultado esperado debe ser un decremento en el tamaño de la Queue. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** isEmptyTest, prueba que el método isEmpty funciona correctamente y retorna el valor booleano esperado | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| QueueTest | isEmpty | setUpEmptyQueue |  | El resultado esperado es que el método debe retornar una booleano indicando que la estructura se encuentra vacia. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** isEmptyTest, prueba que el método isEmpty funciona correctamente y retorna el valor booleano esperado | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| QueueTest | isEmpty | setUpQueue |  | El resultado esperado es que el método debe retornar una booleano indicando que la estructura no se encuentra vacia. |

# **Stack Test Cases**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpEmptyStack | StackTest | An empty Stack. |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpStack | StackTest | (“Thing”)  (“Magical”)  (“a”)  (“not”) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** pushTest, probar que le método push coloca objetos en el stack de la forma esperada. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| StackTest | Push | setUpEmptyStack | (“Pablo”)  (“Daniel”)  (“Andrea”) | El resultado esperado debe ser un incremento en le tamaño del stack. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** peekTest, probar que le método peek retorna el objeto al tope del stack sin eliminarlo. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| StackTest | Push | setUpEmptyStack | (“Pablo”)  (“Daniel”)  (“Andrea”) | El resultado esperado es que el método debe retornar el ultimo objeto del stack sin alterar su tamaño. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** popTest, probar que le método pop retorna el objeto al tope del stack y lo elimina de este. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| StackTest | Push | setUpStack |  | El resultado esperado es que el método debe retornar el ultimo objeto del stack y su tamaño debería decrecer en 1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** isEmptyTest, prueba que el método isEmpty funciona correctamente y retorna el valor booleano esperado | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| StackTest | isEmpty | setUpEmptyStack |  | El resultado esperado es que el método debe retornar una booleano indicando que la estructura se encuentra vacia. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** isEmptyTest, prueba que el método isEmpty funciona correctamente y retorna el valor booleano esperado | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| StackTest | isEmpty | setUpStack |  | El resultado esperado es que el método debe retornar una booleano indicando que la estructura no se encuentra vacia. |

# **LinkedList Test Case**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpEmptyLinkedList | LinkedListTest | An empty LinkedList. |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpLinkedList | LinkedListTest | { 1,2,3,4,5,6,7,8,9} |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** pushTest, probar que le método push coloca objetos en la LinkedLisT de la forma esperada. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| LinkedLisT Test | Push | setUpEmpty LinkedLisT | (“Pablo”)  (“Daniel”)  (“Andrea”) | El resultado esperado debe ser un incremento en le tamaño de la LinkedLisT. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** peekTest, probar que le método peek retorna el objeto al tope de la LinkedList sin eliminarlo. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| LinkedLisT Test | Push | setUpEmpty LinkedLisT | (“Pablo”)  (“Daniel”)  (“Andrea”) | El resultado esperado es que el método debe retornar el ultimo objeto de la LinkedLisT sin alterar su tamaño. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** popTest, probar que le método pop retorna el objeto al tope la LinkedLisT y lo elimina de esta. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| LinkedLisT Test | Push | setUp LinkedLisT |  | El resultado esperado es que el método debe retornar el ultimo objeto del LinkedLisT y su tamaño debería decrecer en 1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** isEmptyTest, prueba que el método isEmpty funciona correctamente y retorna el valor booleano esperado | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| LinkedListTest | isEmpty | setUpEmptyLinkedList |  | El resultado esperado es que el método debe retornar una booleano indicando que la estructura se encuentra vacia. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** isEmptyTest, prueba que el método isEmpty funciona correctamente y retorna el valor booleano esperado | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| LinkedListTest | isEmpty | setUpLinkedList |  | El resultado esperado es que el método debe retornar una booleano indicando que la estructura no se encuentra vacia. |

# **PriorityQueue Test Case**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpEmptyPriorityQueue | PriorityQueueTest | An empty PrirorityQueue. |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUp PriorityQueueTest | PriorityQueueTest | { (“Camile”,1) ,(“Carlos”,2) , (“Daniel”,3) } |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** addTest, probar si el método add agrega adecuadamente objetos dentro del PriorityQueue aun si ya hay objetos dentro de la PriorityQueue. | | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | | **Resultado** |
| PriorityQueueTest | Add | setUp PriorityQueue | (“Capitán Teemo”,666)  (“Capitán Gareb”,667)  (“Capitán Ashe”,668) | | El resultado esperado es que el tamaño de la PriorityQueue vaya aumentando a medida que insertamos objetos en ella |
| **Objetivo de la Prueba:** isEmpty, prueba que el método isEmpty funciona correctamente y retorna el valor booleano esperado | | | | | |
| **Clase** | **Método** | **Escenario** | | **Valores de Entrada** | **Resultado** |
| PirorityQueueTest | isEmpty | setUpEmptyPriorityQueue | |  | El resultado esperado es que el método debe retornar una booleano indicando que la estructura se encuentra vacia. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** popTest, probar que le método pop retorna el objeto al tope la LinkedLisT y lo elimina de esta. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| PriorityQueueTest | pop | setUp PriorityQueue |  | El resultado esperado es que el método debe retornar el ultimo objeto del PriorityQueue y su tamaño debería decrecer en 1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** isEmpty, prueba que el método isEmpty funciona correctamente y retorna el valor booleano esperado | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| PriorityQueueTest | isEmpty | setUpPriorityQueue |  | El resultado esperado es que el método debe retornar una booleano indicando que la estructura no se encuentra vacia. |

# **BST Test Case**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUpEmptyBST | BSTTest | An empty BST. |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUp BSTTest | BSTTest | { 23,5,8,32,12,4,7 } |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** addTest, probar si el método add agrega adecuadamente objetos dentro del BST aun si ya hay objetos dentro del BST. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| BSTTest | Add | setUpBST | (“Capitán Teemo”)  (“Capitán Gareb”)  (“Capitán Ashe”) | El resultado esperado es que el tamaño del BST vaya aumentando a medida que insertamos objetos en la estructura |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** removeTest, probar si el método remove borra adecuadamente objetos dentro del BST. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| BSTTest | remove | setUpBST |  | El resultado esperado es que el elemento sea borrado por lo cual nos retorna el objeto a borrar. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** searchTest, probar si el método add agrega adecuadamente objetos dentro del BST aun si ya hay objetos dentro del BST. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| BSTTest | search | setUpBST |  | El resultado esperado es que el método retorne el elemento buscado dentro del árbol binario de búsqueda. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** isEmpty, prueba que el método isEmpty funciona correctamente y retorna el valor booleano esperado | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| BSTTest | isEmpty | setUpEmptyBST |  | El resultado esperado es que el método debe retornar una booleano indicando que la estructura se encuentra vacia. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** isEmpty, prueba que el método isEmpty funciona correctamente y retorna el valor booleano esperado | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| BSTTest | isEmpty | setUpBST |  | El resultado esperado es que el método debe retornar una booleano indicando que la estructura no se encuentra vacia. |