|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Clase* | *Método* | *Escenario* | *Valores de entrada* | *Resultado* |
| IStack | pop():T | [10, 20, 5, 40] | Este Método no tiene Entradas | 40.  Se retorna el 40. Y la lista queda con los siguientes valores.  [10, 20, 5] |
| IStack | last():T | [10, 20, 5, 40, 100, 541] | Este Método no tiene entradas. | 541.  El valor esperado es 541. La lista no cambia. |
| IStack | removeLastElement():void | [10, 20, 5, 40, 100, 541] | Este Método no tiene entradas. | Se elimina 541 y la lista debe quedar de la siguiente manera.  [10, 20, 5, 40, 100] |
| IStack | push(T Value):void | [10, 20, 5, 40, 100, 541] | 99 | Se agrega el 99 en la última posición y la lista queda de la siguiente forma.  [10, 20, 5, 40, 100, 541, 99] |
| IStack | isEmpty():boolean | 1. [10, 20, 5, 40, 100, 541] 2. [] | Este método no tiene entradas. | * Retorna false, ya que el tamaño de la lista es mayor a 0 * Retorna True, ya que el tamaño de la lista es 0 y eso implica que está vacía |
| IStack | size():int | 1. [10, 20, 5, 40, 100, 541] 2. [] | Este método no tiene entradas. | 1. 6 2. 0 |
| IQueue | offer(T e):void | 1. [Hola, Joe, 10, 10.22] 2. [asd, qwe, ew] | 1. Hi 2. Kwe | 1. Se agrega “Hi” a la cola la lista queda de la siguiente manera: [Hi, Hola, Joe, 10, 10.22,] 2. En la segunda entrada queda de la siguiente manera: [kwe, asd, qwe, ew] |
| IQueue | removeFirstElement():void | 1. [Hola, Joe, 10, 10.22] 2. [asd, qwe, ew] | Este método no tiene entradas. | 1. Se elimina el “Hola” de la lista, queda de la siguiente manera: [Joe, 10, 10.22] 2. Se elimina el “asd” de la lista, queda de la siguiente manera [qwe, ew] |
| IQueue | poll():T | [asd, qwe, ew] | Este método no tiene entradas | Asd  Retorna “asd” y la lista queda de la siguiente manera [qwe, ew] |
| IQueue | firts():T | [lqew, duew, qweq] | Este método no tiene entradas | Lque  Este método te retonar el primer elemento de una lista |
| IQueue | size():T | 1. [10, 20, 5, 40, 100, 541] 2. [] | Este método no tiene entradas | 1. u |
| IQueue | isEmpty():boolean | 1. [10, 20, 5, 40, 100, 541] 2. [] | Este método no tiene entradas | 1. Retorna false ya que el tamaño es mayor que 0 2. Retorna true ya que el tamaño es igual a 0 |
| ITable | get(Object key):V | {[“Test1”,”1”],[“Test2”,”2”],[“Test3”,”3”]} | “Test2” | “2”  Devuelve el valor que almacena la llave “Test2” el cual es “2” |
| ITable | hashCode(K key): int | {[1,”1”],[2,”2”],[6,6]} | 4 | 4  Devuelve el código hash que se le asigna según la función hash para clave igual a 4 que es 4 |
| ITable | isEmpty(): boolean | 1. {} 2. {[“Test1”,1]} | Este Método no tiene entradas. | 1. Devuelve true porque la cantidad de elementos almacenados en la tabla es 0 2. Devuelve false porque la tabla no está vacía, hay un elemento. |
| ITable | put(K key,V value): boolean | {} | “Test1”,”1” | True  Devuelve true porque fue posible insertar el elemento con clave valor de entrada en la tabla. La tabla quedo con un elemento adicional. |
| ITable | remove(K key): boolean | 1. {[“Test1”,”1”], [“Test2”,”2”]} 2. {[“Test1”,”1”],[“Number”,”2”]} | “Number” | 1. Devuelve false porque no encontró coincidencias con la llave. 2. Devuelve true porque encontró la llave “Number” y la elimino |
| ITable | size(): int | 1. {[“Number”,”3”]} 2. {[1,”3”],10,”3”]} | Este Método no tiene entradas. | 1. Devuelve la cantidad de llaves distintas almacenadas, 1 porque solo hay “Number” 2. Devuelve 1 porque hay dos llaves con el mismo valor hash |
| ITable | clear(): boolean | {[“Test1”,”2”]} | Este Método no tiene entradas. | True  Devuelve true porque no hay elementos almacenados |
|  |  |  |  |  |