|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Queue | Peek(); | El escenario inicializa una varios Nodos, los cuales tienen como valor de tipo integer que son: 1,2,3,4,5,6,7,8,9 | --------------------- | Devuelve el primer nodo (1) puesto que ya estaba insertado en la lista |
| Queue | Peek(); | Solo inicializa el queue | ----------------------- | Devuelve null, puesto que la queue esta vacia |
| Queue | Peek(); | El escenario inicializa un nodo, el cual tiene como valor de tipo integer que son: 1 | ----------------------- | Devuelve1, puesto que la queue solo tiene un elemento |

**Objetivo:** Saber si el método peek() funciona de manera adecuada simulando ser una estructura contenedora creada por Java en una de sus liberias.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Queue | Poll(); | El escenario inicializa una varios Nodos, los cuales tienen como valor de tipo integer que son: 1,2,3,4,5,6,7,8,9 y se elimina los dos primero | ---------------------- | Devuelve el 3 puesto que se eliminaron los dos primeros |
| Queue | Poll() | Solo inicializa el queue | ---------------------- | No devuelve nada puesto a que no se elimina nada |
| Queue | Poll() | El escenario inicializa un nodo, el cual tiene como valor de tipo integer que son: 1 | ---------------------- | Devuelve el 1 y elimina el resto |

**Objetivo:** Saber si el método Poll() elimina de una manera correcta dado que es una simulación de una estructura contenedora creada por Java en una de sus librerias.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Queue | Offer() | El escenario inicializa una varios Nodos, los cuales tienen como valor de tipo integer que son: 1,2,3,4,5,6,7,8,9 y se elimina los dos primero | Node q2 = new Node(4) | Lo agrega al inicio de la lista, en el front |
| Queue | Offer() | Solo inicializa el queue | Null | No lo agrega,p uesto que es nulo |
| Queue | Offer() | El escenario inicializa un nodo, el cual tiene como valor de tipo integer que son: 1 | Node q2 = new Node(4) | Lo agrega de primero |

**Objetivo:** Saber si el método offer agrega de una manera correcta dentro de la estructura contenedora.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Queue | isEmpty() | El escenario inicializa una varios Nodos, los cuales tienen como valor de tipo integer que son: 1,2,3,4,5,6,7,8,9 | -------------------- | Falso, dado que la estructura contenedora no está vacía |
| Queue | isEmpty() | Solo inicializa el queue | ---------------------- | Verdadero, dado que la estructura contenedora esta vacía. |
| Queue | isEmpty() | El escenario inicializa un nodo, el cual tiene como valor de tipo integer que son: 1 | --------------------- | False, dado que la estructura contenedora tiene al menos un elemento creado |

**Objetivo:** Saber si el método isEmpty devuelve de manera correcta dentro de la estructura contenedora.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Stack | Peek() | El escenario inicializa una varios Nodos, los cuales tienen como valor de tipo integer que son: 1,2,3 | ----------------------- | Devuelve el primer elemento que en este caso es 3 |
| Stack | Peek() | Solo inicializa el stack | ---------------------- | No retorna nada dado a que no hay nada dentro de la estructura |
| Stack | Peek() | El escenario solo inicializa un nodo, el cual tienen como valor de tipo integer: 1 | ----------------------- | Retorna el primer elemento que en este caso es el 1 |

**Objetivo:** Saber si el método peek() devuelve de manera correcta dentro de la estructura contenedora.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Stack | Pop(); | El escenario inicializa una varios Nodos, los cuales tienen como valor de tipo integer que son: 1,2,3 | ---------------------- | Se elimina el 3 y el primero queda como el 2 que en esta |
| Stack | Pop() | Solo inicializa el Stack | ----------------------- | No elimina el nada dado que no hay ningún elemento |
| Stack | Pop() | El escenario solo inicializa un nodo, el cual tienen como valor de tipo integer: 1 | ----------------------- | Elimina el primer elemento que en este caso es el 1 y el stack esta vacio |

**Objetivo:** Saber si el método Pop() elimina de manera correcta dentro de la estructura contenedora creada.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Stack | isEmpty() | El escenario inicializa una varios Nodos, los cuales tienen como valor de tipo integer que son: 1,2,3 | ------------------------ | Devuelve falso, dado que la lista contenedora no esta vacía |
| Stack | isEmpty() | Solo inicializa el Stack | ---------------------- | Devuelve verdadero, dado que la lista contenedora esta vacía |
| Stack | isEmpty() | El escenario solo inicializa un nodo, el cual tienen como valor de tipo integer: 1 | ----------------------- | Devuelve falso, dado que la lista contenedora no esta vacia. |

**Objetivo:** Saer si el método isEmpty() devuelve de manera correcta los valores para indicar si la lista esta vacia.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Stack | Offer() | El escenario inicializa una varios Nodos, los cuales tienen como valor de tipo integer que son: 1,2,3 | Node q1 = **new** Node(5); | Agrega al elemento de primero dentro de la estructura contenedora |
| Stack | Offer() | Solo inicializa el Stack | Node q1 = **new** Node(1); | Agrega al elemento de primero dentro de la estructura contenedora |
| Stack | Offer() | El escenario solo inicializa un nodo, el cual tienen como valor de tipo integer: 1 | Node q1 = **new** Node(2); | Agrega al elemento de primero dentro de la estructura contenedora |

**Objetivo:** Saber si el método offer() agrega de manera correcta dentro de la estructura contenedora.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| HashTable | getValue(Key) | El escenario inicializa varias llaves y valores tales como (15,140),  (15,160),(0,140) | Key k =15 | Devuelve el valor de 160, dado que hubo una colisión y retorna el ultimo que se dio |
| HashTable | getValue(Key) | Solo se inicializa la hashTable | Key k = 200 | Retorna null dado que no se encontró nada |
| HashTable | getValue(Key) | El escenario solo inicializa una llave y un valor (15,140) | Key k =15 | Devuelve el valor de 140, dado que es el que esta en el arreglo |

**Objetivo:** Saber si el método getValue(key) devuelve el valor de una manera correcta.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| HashTable | Search(Key) | El escenario inicializa varias llaves y valores tales como (15,140),  (15,160),(0,140) | Key k = 15 | Devuelve el 160 dado que es el valor que ahí se encuentra |
| HashTable | Search(Key) | Solo se inicializa la hashTable | Key k = 16 | Devuelve null, dado que se encuentra vacio todas las posiciones |
| HashTable | Search(Key) | El escenario solo inicializa una llave y un valor (15,140) | Key k = 15 | Devuelve el valor de 140, dado que es el valor que ahí se encuentra. |

**Objetivo:** Saber si el método Searck(Key) Busca y devuelve de una manera correcta dentro de la estructura HashTable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| HashTable | Insert(K,V) | El escenario inicializa varias llaves y valores tales como (15,140),  (15,160),(0,140) | (16,3012) | Agrega el valor dentro de esa llave |
| HashTable | Insert(K,V) | Solo se inicializa la hashTable | (15,16000) | Agrega el valor dentro de esa llave |
| HashTable | Insert(K,V) | El escenario solo inicializa una llave y un valor (15,140) | (0,33) | Agrega el valor dentro de esa llave |

**Objetivo:** Saber si el método Insert(K,V) agrega de manera correcta dentro de la estructura HashTable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| HashTable | Delate(K) | El escenario inicializa varias llaves y valores tales como (15,140),  (15,160),(0,140) | Key k = 16 | No elimina el valor dentro de esa llave ya que no contiene nada |
| HashTable | Delate(K) | Solo se inicializa la hashTable | Key k = 33 | No elimina nada, puesto que no contiene nada |
| HashTable | Delate(K) | El escenario solo inicializa una llave y un valor (15,140) | Key k = 15 | Elimina la llave dentro de esa posición. |

**Objetivo:** Saber si el método Delate, elimina de una manera correcta dentro de la hashTable.