**TADs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Queue TAD** | | | |
| Queue = { <E1>, <E2>, <…> , <En>} | | | |
| { inv: TRUE } | | | |
| Primitive operations: | | | |
| * Queue |  | * Queue | Constructor |
| * isEmpty | Queue | * Boolean | Analyzer |
| * peek | Queue | * Element | Analyzer |
| * poll | Queue | * Element | Modifier |
| * offer | Queue X Element | * Queue | Modifier |
| * size | Queue | * Integer | Analyzer |

|  |
| --- |
| **Queue** |
| Constructor of the Queue class that creates an empty queue. { pre: TRUE}  { post: queue = { front = NIL, back = NIL, n = 0 } } |

|  |
| --- |
| **isEmpty(***queue***)** |
| Determines whether the queue is empty. Returns 'true' if the queue is empty and 'false' if it contains elements.  { pre: TRUE }  { post: True if queue = { front = NIL, back = NIL, n = 0 }  False if queue = { front != NIL, n > 0 } } |

|  |
| --- |
| **peek(***queue***)** |
| Returns the element at the front of the queue without removing it.  { pre: n > 0 }  { post: peek = <E1> } |

|  |
| --- |
| **poll(***queue***)** |
| Remove and return the element at the beginning of the queue.  { pre: n > 0 }  { post: poll = <E1>, queue = { <E2>, <…> , <En> } } |

|  |
| --- |
| **offer(***queue, elem***)** |
| Adds an element to the end of the queue.  { pre: TRUE }  { post: queue = { <E2>, <…> , <En = elem> } } |

|  |
| --- |
| **size(***queue***)** |
| Returns the number of elements in the queue.  { pre: TRUE }  { post: size = n } |