|  |  |
| --- | --- |
| NAME | QUEUE |
| REPRESENTATION | Java Queue |
| INVARIANT | Size >= 0 |
| OPERATIONS | **Primitive Operation**  Queue 🡪 Queue  Offer: Element  Poll 🡪 Element  Peek 🡪 Element  Size 🡪 Integer  IsEmpty🡪boolean  Clear 🡪 void |

|  |
| --- |
| Queue 🡪 Queue  “Create a new queue”  {Pre: true}  {Pos: An empty queue} |

|  |
| --- |
| Offer: Element  “Insert a new Element in the Queue”  {Pre: ø}  {Pos: The element is inserted at the end of the queue} |

|  |
| --- |
| Poll 🡪 Element  “Delete the front element of the QUEUE”  {Pre: ø}  {Pos: Get and erase the front item} |

|  |
| --- |
| Peek 🡪 Element  “Get the front element of the QUEUE”  {Pre: ø}  {Pos: Returns the front element} |

|  |
| --- |
| Size 🡪 integer“Get the QUEUE size” {Pre: size >= 0}  Pos: Returns the QUEUE size |

|  |
| --- |
| Clear“clear the QUEUE” {Pre: size >= 0}  {Pos: Returns an empty QUEUE} |

|  |
| --- |
| IsEmpty 🡪 boolean“verify if the QUEUE is empty” {Pre: size >= 0}  {Pos: true if queue is empty, false if is not} |

|  |  |
| --- | --- |
| NAME | HASH TABLE |
| REPRESENTATION | HashTable in Java Example | Java HashTable Tutorial |
| INVARIANT | Inv {All keys cannot be null,  Every value will have a key} |
| OPERATIONS | **Primitive Operation** HashTable 🡪 HashTableHashFunction: Key Search: Key 🡪 Value  Insert: Key, Value  Delete: Key |

|  |
| --- |
| HashTable 🡪 HashTable  “Creates a new hashTable”  {Pre: true}  {Pos: An empty hashTable} |

|  |
| --- |
| HashFunction: Key  “Convert the key to an integer”  {Pre: True}  {Pos: An integer that indicates the object position in the array} |

|  |
| --- |
| Search: Key 🡪 Value  “Get the value searched”  {Pre: ø}  {Pos: Returns the element found} |

|  |
| --- |
| Insert: Key, Value  “Add a new element in the hash table”  {Pre: Key && Value ≠ null}  {Pos: Insert the element in the hash table} |

|  |
| --- |
| Delete: Key  “Delete an element with a key”  {Pre: Key ≠ null}  {Pos: Erase the found element} |

|  |  |
| --- | --- |
| NAME | STACK |
| REPRESENTATION | Stack Class in Java Explained with Examples | CodeAhoy |
| INVARIANT | Size >= 0 |
| OPERATIONS | **Primitive Operation**  Push: Element  isEmpty 🡪 Boolean  Top 🡪 Element  Pop 🡪 Element  Clear  Size 🡪integer |

|  |
| --- |
| Push: Element  “Add a new element in the Stack”  {Pre: Element ≠ null}  {Pos: The element has added into the stack} |

|  |
| --- |
| isEmpty 🡪 Boolean  “Says if the Stack is empty”  {Pre: ø}  {Pos: Yes or Not if its empty} |

|  |
| --- |
| Top 🡪 Element  “Get the first element”  {Pre: ø}  {Pos: returns the first element in the stack} |

|  |
| --- |
| Pop 🡪 Element  “Delete the first element”  {Pre = ø}  {Pos= The first element must be eliminated} |

|  |
| --- |
| Clear  “Clear the entire stack”  {Pre: ø}  {Pos: an empty stack} |

|  |
| --- |
| Size 🡪 integer  “Get the size of the Stack”  {Pre: ø}  {Pos: Get an integer with the size of the STACK} |

|  |  |
| --- | --- |
| NAME | HEAP |
| REPRESENTATION | Heap Data Structure - GeeksforGeeks |
| INVARIANT |  |
| OPERATIONS | **Primitive Operation**  Heap🡪 Heap  buildMaxHeap  maxHeapify  exist 🡪 Boolean |