**TADs**

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

|  |
| --- |
| **Stack** |
| Constructor for the Stack class. Initializes a new stack with a default capacity.  { pre: TRUE}  { post: stack = { data = NIL, size = 0 } } |

|  |
| --- |
| **push(***stack, elem***)** |
| Pushes an element on the top of the stack.  { pre: TRUE }  { post: queue = { <E1>, <…> , <En = elem> } } |

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

|  |
| --- |
| **pop(***queue***)** |
| Removes and returns the element from the top of the stack.  { pre: size > 0 }  { post: pop = <En>, queue = { <E1>, <…> , <En> } } |

|  |
| --- |
| **isEmpty(***queue***)** |
| Checks if the stack is empty.  { pre: TRUE }  { post: True if queue = { data = NIL, size = 0 }  False if queue = { data != NIL, size > 0 } } |

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

|  |
| --- |
| **resize(***queue***)** |
| Doubles the capacity of the internal data array when the stack is full.  { pre: data.length == size }  { post: queue = { data = <E1>, <…> , <Esize>, size = size\*2 } } |