|  |  |  |
| --- | --- | --- |
| STACK | | |
| Stack= {Top= <StackNode<T>>, Size=<Integer>} | | |
| {Inv: size≥0 ⋀ size ∈ Z} | | |
| Stack:  Push:  Pop:  Peek:  IsEmpty:  GetSize: | STACK x T  STACK  STACK  STACK  STACK | 🡪 STACK  → RACIONAL  🡪 T  🡪 T  → BOOLEAN  → INTEGER |

**Constructor Operations:**

|  |
| --- |
| **Stack()**  “Creates a new Stack”  {pre: TRUE}  {post: Stack= {Top= NULL, Size = 0} } |

**Modifying Operations:**

|  |
| --- |
| **Push()**  “Adds a new element to the Stack structure, by also setting a new Top”  {pre: Stack={Top=<StackNode<T>>, Size=n}, T=<content>}  {post: Stack={Top=<StackNode<T>>, Size=(n+1)} } |

|  |
| --- |
| **Pop()**  “Return the element on the top of the Stack and deletes it from the structure”  {pre: TRUE}  {post: T=<content> if Size ≠0, else NULL} |

**Analyzing Operation:**

|  |
| --- |
| **Peek()**  “Returns the value of the element at the top of the Stack”  {pre: TRUE}  {post: T=<content> if Size ≠0, else NULL } |

|  |
| --- |
| **IsEmpty()**  “Indicates whether the Stack has elements or not”  {pre: TRUE}  {post: TRUE if Size≠0, else FALSE } |

|  |
| --- |
| **GetSize()**  “Returns the number of elements in the Stack structure”  {pre: TRUE}  {post: Size=n } |

|  |  |  |
| --- | --- | --- |
| STACKNODE | | |
| StackNode = {Top= <StackNode<T>>, Bottom= <StackNode<T>>, Content=T} | | |
| {Inv: Content ≠NULL} | | |
| StackNode:  SetTop:  SetBottom:  SetContent:  GetTop:  GetBottom:  GetContent: | T  STACK x T  STACK  STACK  STACK  STACK  STACK | 🡪 STACKNODE  → STACKNODE  🡪 STACKNODE  🡪 STACKNODE  → STACKNODE  → STACKNODE  🡪 T |

**Constructor Operations:**

|  |
| --- |
| **StackNode()**  “Creates a new StackNode”  {pre: T=<content>}  {post: StackNode= {Content= T, Top = NULL, Bottom=NULL} } |

**Modifying Operations:**

|  |
| --- |
| **SetTop()**  “Changes the Top of the StackNode”  {pre: NewStackNode = {Top= <StackNode<T>>, Bottom= <StackNode<T>>, Content=T}}  {post: StackNode = {Top= NewStackNode, Bottom= <StackNode<T>>, Content=T}} |

|  |
| --- |
| **SetBottom()**  “Changes the Bottom of the StackNode”  {pre: NewStackNode = {Top= <StackNode<T>>, Bottom= <StackNode<T>>, Content=T}}  {post: StackNode = {Top= <StackNode<T>>, Bottom= NewStackNode, Content=T}} |

|  |
| --- |
| **SetContent()**  “Changes the Content of the StackNode”  {pre: NewT=<content>}  {post: StackNode = {Top= <StackNode<T>>, Bottom= <StackNode<T>>, Content=NewT}} |

**Analyzing Operation:**

|  |
| --- |
| **GetTop()**  “Returns the Top of the StackNode”  {pre: TRUE}  {post: StackNode = {Top= <StackNode<T>>, Bottom= this StackNode, Content=T}} |

|  |
| --- |
| **GetBottom()**  “Returns the Bottom of the StackNode”  {pre: TRUE}  {post: StackNode = {Top= this StackNode, Bottom= <StackNode<T>>, Content=T}} |

|  |
| --- |
| **GetContent()**  “Returns the Content of the StackNode”  {pre: TRUE}  {post: T=<content>}} |