

| TAD Stack |
|---|
| Stack = << e1, e2, e3, ...,en>, top> |
| Invariant: $0 \leq n \wedge \text{Stack.size} = n \wedge \text{top} = \text{en}$ |
| Construction Operations: *Create: ---> Stack Modifier Operations: *push: Stack x Element --> Stack *pop: Stack --> Stack *top: Stack --> Element Analyzer operations: *isEmpty: Stack --> boolean |

| |
|---|
| Create (Value) "Creates an empty Stack" { pre: TRUE } { post: Stack s = ∅ } |
|---|

| |
|---|
| push(e) "Adds the new element e to Stack s" {pre: Stack s = <e1, e2, e3, ...,en> and element e or s = ∅ and element e} { post: Stack s = <e1, e2, e3, ...,en, e> or s = <e> } |
|---|

| |
|--|
| pop() "Extracts from the Stack s the most recently inserted element" {pre: Stack s is not ∅, i.e. s = <e1, e2, e3, ..., en-1, en>} { post: Stack s = <e1, e2, e3, ...,en-1>} |
|--|

| |
|--|
| top(): "Recovers the value of the element on the top of the stack " {pre: Stack s is not ∅, i.e. s = <e1, e2, e3, ..., en-1, en>} { post: element e_n} |
|--|

| |
|---|
| isEmpty(): "Determines wheter the stack s is empty" {pre: Stack s} { post: True is s = ∅, False otherwise } |
|---|