|  |
| --- |
| **TAD Stack** |
| Stack = {elements =〈〉} ()  *∀ n ∈ N* |
| {inv*:*   * *It must follow the Last In First Out (LIFO) principle, ∀ x, y ∈ [1, n], where n is the number of elements in the stack, if x was pushed before y, in order to pop x first needs to be popped y.* * *∀ x ∈ Stack, x should be of a consistent type.)}* |
| * Create Stack: Stack * Add Element: Stack X Element Stack * Remove Element: Stack Stack X Element * First Element: Stack Element * Check Stack: Boolean |
| **Create Stack [***Stack***] – Constructor**  *“Creates a new and empty priority queue”.*  *{Pre: None}.*  *{Post: Returns an empty Stack}.*  **Add Element [***push***] – Modifier**  *“Given an element by parameter it is pushed into the top stack”.*  *{Pre: There must be a Stack, and the element must be of a compatible type}.*  *{Post: Returns the updated stack}.*  **Remove Element [***pop***] - Modifier**  *“Removes and returns the at the top of the stack (the last one pushed), else it returns null”.*  *{Pre: There must be Stack}.*  *{Post: Returns the updated Stack and the element removed.}*  **FirstElement [***top***] - Analyser**  “*Returns the element at the top of the stack (the last one pushed), else it returns null”.*  *{Pre: There must be a Stack}.*  *{Post: Returns the elements}.*  **Check Stack [***isEmpty***] - Analyser**  “*Returns a Boolean value corresponding to the elements on the stack, if there is any returns false but if it has no elements returns true”.*  *{Pre: None}.*  *{Post: A corresponding Boolean value is returned}.* |