<<interface>> StackInterface

+push(newEntry T): void

+pop(): T +getPeek(): T

+isEmpty(): boolean

+clear(): void

ArrayBag

- -_bag: T[]
- -_ numOfItems: integer-_capacity: integer- isEmpty: boolean
- +push(newEntry T): void
- +pop(): T
- +getPeek(): T
- +isEmpty(): boolean
- +clear(): void
- isVariable(charcter wlkr): boolean
- caseOfInteger(character wlkr, ResizeableArrayStack valStack): void
- isOperator(character wlkr): boolean
- $\hbox{-} case Of Operator (character wlkr, Resizeable Array Stack val Stack): void\\$
- valculate(character wlkr, ResizeableArrayStack): void

LinkedStack

- head: Node
- _numOfItems: integerisEmpty: boolean
- +push(newEntry T): void
- +pop(): T
- +getPeek(): T
- +isEmpty(): boolean
- +clear(): void
- +convertToPostfix(inFix String): LinkedStack pstStack
- iteration(LinkedStack pstFix, LinkedStack opStack) //Fist half
- postIteration(LinkedStack pstFix, LinkedStack opStack) LinkedStack postFix // second half
- isAlpha(wlkr character): boolean
- caseOfAlpha(wlkr character, pstFix LinkedStack): void
- isPower(wlkr character): boolean
- caseOfPower(wlkr character, pstFix LinkedStack): void
- isOperator(wlkr character): boolean
- caseOfOperator(wlkr character, pstFix LinkedStack, opStack LinkedList): void
- int giveValue (wlkr character): int correspondingValue
- isWeaker(wlkr character): boolean
- isLeftParenthesis(wlkr character): boolean
- caseOfLeftParenthesis(wlkr character, opStack LinkedStack): void
- isRightParenthesis(wlkr character): boolean
- caseOfRightParenthesis(wlkr character, pstStack LinkedList, opStack LinkedStack): void

Node

- -_data: T - next: Node
- +getData(): T
- +setData(): void
- +getNext(): Node +setData(): void