





ILlateT>
+addFirst(type: T): viol
+addEnt(type: T): viol
+addEnt(type: T): viol
+addEnt(type: T): viol
+addEnter(typeNev: T, position: T): viol
+addInter(typeNev: T, position: T): viol
+addInter(typeNev: T, index: int): viol
+removel.ast]: Node=T>
+removel.ast]: Node=T>
+removel.ast]: Node=T>
+deiter(tast): viol
+deiter(tast

iStack<T>
+push(type: T): void
+pop(): Node<T>
+isEmpty(): boolean
+lenght(): int
+top(): Node<T>

iHash<T>
+isEmpty(): boolean
+length(): int
+tabieInsert(newNode: Node<T>)
+tabieRetrieve(): Node<T>
+tabieRetrieve(): Node<T>
+hashFunction(key:int): int

IQueue<T>
+equeue(type: T): void
+fronte(): Node<T>
+dequeue(): Note<T>
+isEmpty(): boolean
+length(): int