<<abstract>> Node #_Degree : int # Parent : Node # NumKeys: int #_Keys : int[] #Node(degree : int) +SearchKey(key:int): (int, Node) +Split(): ((int, Data), Node) +IsFull(): bool +IsUnderflow(): bool +InsertKey(key: int, data: Data): ((int, Data), Node) +DeleteKey(key:int):void +Traverse(): string +GetNumKeys(): int <<abstract>> **BTreeNode** #_Contents : Data[]

#BTreeNode(degree : int)

NonLeafNode

- children : BTreeNode

+NonLeafNode(degree : int, keys

: int[], data : Data, children :

BTreeNode[])

+searchKey(key: int): (int,

BTreeNode)

+SearchKey(key:int): (int,

BTreeNode)

+InsertKey(currentNode:

BTreeNode, key: int, data: Data)

: ((int, Data), Node)

+Split(): ((int, Data), Node)

+IsFull(): bool

+IsUnderflow(): bool

+DeleteKey(key: int): void

+Traverse(currentNode:

BTreeNode, result : string) : string

LeafNode

+LeafNode(degree : int, keys :

int[], contents : Data[])

-Search(key: int): int

+SearchKey(key: int): (int, Node)

+Split(): ((int, Data), Node)

+IsFull(): bool

+IsUnderflow(): bool

+InsertKey(key: int, data: Data):

((int, Data), Node)

+DeleteKey(key: int): void

+Traverse(): string