

List
<ul style="list-style-type: none"> - count : int - head* : Node<T>
<ul style="list-style-type: none"> + List() : + ~List() : + isEmpty() : bool + getCount : int + insert(:T& , : int) : void + insertFirst(: T&) : void + insertLast(: T&) : void + remove(: int) : void + removeFirst() : void + removeLast() : void + removeAll() : void + getData(: int) : T& + getFirstData() : T& + getLastData() : T& + setData (: T&, : int) : void + setFirstData(: T&) : void + setLastData(: T&) : void

Node
<ul style="list-style-type: none"> + data : T& + next* : Node<T>
<ul style="list-style-type: none"> + Node(: T&) : + Node(: T&, : Node<T>*) :

Queue
<ul style="list-style-type: none"> - front : Node<T>* - rear : Node<T>*
<ul style="list-style-type: none"> + Queue() : + enqueue(: T&) : void + dequeue() : T& + getFront() : T& + getRear() : T& + getCount() : int + empty() : void + isEmpty() : bool

Person
<ul style="list-style-type: none"> - name : std::string - Birthday : std::string
<ul style="list-style-type: none"> + Person(: string, : string) : + ~Person() : virtual

BST_Node
<ul style="list-style-type: none"> + key : T& + data : T& + left* : BST_Node<T> + right* : BST_Node<T>
<ul style="list-style-type: none"> + BST_Node(: T&, : T&, : BST_Node<T>* = nullptr, BST_Node<T>* = nullptr) : + ~BST_Node() : virtual

BST
<pre> # count : int # head : BST_Node<T>* + BST() : + ~BST() : virtual + getHead() : BST_Node<T>* + insert(: T&, : T&) : void + remove(: T&) : void + find(: T&) : BST_Node<T>* + deleteAll() : void + modify(: T, : T&, : bool replacekey = true) : void # insert(: T&, : T&, : BST_Node<T>*) : BST_Node<T>* # remove(: T&, : BST_Node<T>*) : BST_Node<T>* # findMin(: BST_Node<T>*) : BST_Node<T>* # find(: BST_Node<T>*, : T&) : BST_Node<T>* # queueNodes(: Queue<BST_Node<T>>&, : BST_Node<T>*) : BST_Node<T>* </pre>