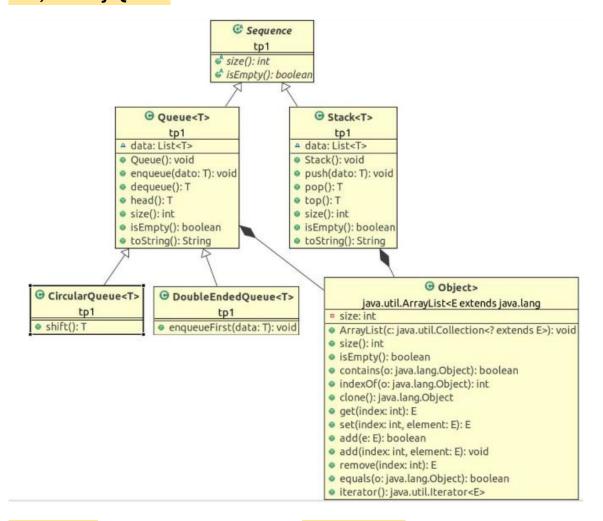
List, Stack y Queue



BinatyTree

⊙ BinaryTree<T>

- a data: T
- leftChild: BinaryTree<T>
- p rightChild: BinaryTree<T>
- BinaryTree(): void
- BinaryTree(T): void
- getdata(): T
- setdata(T): void
- getLeftChild(): BinaryTree<T>
- getRightChild(): BinaryTree<T>
- addLeftChild(BinaryTree<T>): void
- addRightChild(BinaryTree<T>): void
- removeLeftChild(): void
- removeRightChild(): void
- isEmpty(): boolean
- o isLeaf(): boolean
- hasLeftChild(): boolean
- hasRightChild(): boolean
- toString(): String
- o contarHojas(): int
- espejo(): BinaryTree<T>
- entreNiveles(int, int): void

GeneralTree

→ GeneralTree<T>

- data: T
- children: List<GeneralTree<T>>
- GeneralTree(): void
- GeneralTree(T): void
- GeneralTree(T, List<GeneralTree<T>>): void
- getData(): T
- setData(T): void
- getChildren(): List<GeneralTree<T>>
- setChildren(List<GeneralTree<T>>): void
- addChild(GeneralTree<T>): void
- isLeaf(): boolean
- hasChildren(): boolean
- isEmpty(): boolean
- removeChild(GeneralTree<T>): void
- altura(): int
- nivel(T): int
- ancho(): int

attree