List, Stack y Queue

Sequence
tp1

- size(): int
- isEmpty(): boolean

Queue<T>
tp1

- data: List<T>
- Queue(): void
- enqueue(dato: T): void
- dequeue(): T
- head(): T
- size(): int
- isEmpty(): boolean
- toString(): String

Stack<T>
tp1

- data: List<T>
- Stack(): void
- push(dato: T): void
- pop(): T
- top(): T
- size(): int
- isEmpty(): boolean
- toString(): String

CircularQueue<T>
tp1

- shift(): T

DoubleEndedQueue<T>
tp1

- enqueueFirst(data: T): void

Object>
java.util.ArrayList<E extends java.lang

- size: int
- ArrayList(c: java.util.Collection<? extends E>): void
- size(): int
- isEmpty(): boolean
- contains(o: java.lang.Object): boolean
- indexOf(o: java.lang.Object): int
- clone(): java.lang.Object
- get(index: int): E
- set(index: int, element: E): E
- add(e: E): boolean
- add(index: int, element: E): void
- remove(index: int): E
- equals(o: java.lang.Object): boolean
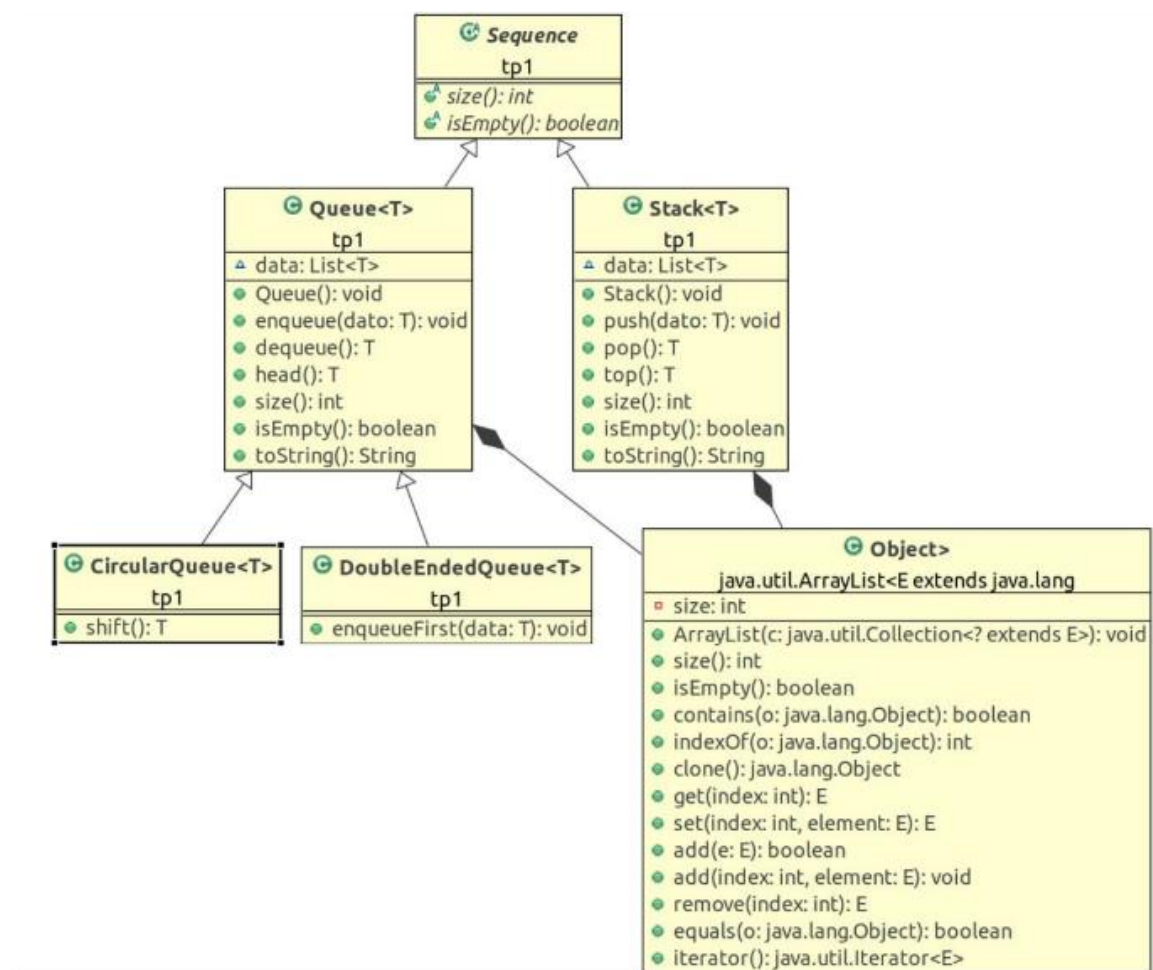- iterator(): java.util.Iterator<E>

BinatyTree

BinaryTree<T>

- data: T
- leftChild: BinaryTree<T>
- 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
- isLeaf(): boolean
- hasLeftChild(): boolean
- hasRightChild(): boolean
- toString(): String
- 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