



BinaryTree<Key extends Comparable<Key>, Value>
-M: final int -root: Node -height: int -n: int
+BinaryTree() +isEmpty(): boolean +size(): int +height(): int +get(key: Key): Value -search(x: Node, key: Key, ht: int): Value +put(key: Key, val: Value): void -insert(h: Node, key: Key, val: Value, ht: int): Node -split(h: Node): Node +toString(): String -toString(h: Node, ht: int, indent: String): String -less(k1: Comparable, k2: Comparable): boolean -eq(k1: Comparable, k2: Comparable): boolean

MultiThread extends Thread
+run(): void

Main extends Application
+main(args: String[]): void +start(stage: Stage): void

Login
+login(stage: Stage): Scene +register(stage: Stage): Scene

Library
-userOnline: User -books: BinaryTree<Integer, String> -users: Map<Integer, String> -checkedOutBooks: Queue<Books>
+memberHome(stage: Stage, id: int, name: String): Scene +adminHome(stage: Stage, id: int, name: String): Scene +profile(stage: Stage): Scene +searchBooks(stage: Stage, userType: int): Scene +viewLibrary(stage: Stage, userType: int): Scene +saveBook(id: int, bookname: String): void

Book
-bookID: int -name: String
+Book(bookID: int, name: String) +getBookID(): int +setBookID(bookID: int): void +getBookName(): String +setBookName(name: String): void +toString(): String