Documentatie: Log boek server

Klassen:

**TreeNode**

De klasse TreeNode vertegenwoordigt een knooppunt in de binaire zoekboom. Elk knooppunt bevat een sleutel en een waarde, evenals referenties naar het linker- en rechterkind. Deze klasse biedt methoden om toegang te krijgen tot en wijzigingen aan te brengen in de eigenschappen van het knooppunt.

Eigenschappen:

key: De sleutelwaarde van het knooppunt.

value: De bijbehorende waarde van het knooppunt.

left: Referentie naar het linkerkind van het knooppunt.

right: Referentie naar het rechterkind van het knooppunt.

Methoden:

getKey(): Geeft de sleutelwaarde van het knooppunt terug.

setKey(String key): Stelt de sleutelwaarde van het knooppunt in.

getValue(): Geeft de waarde van het knooppunt terug.

setValue(String value): Stelt de waarde van het knooppunt in.

getLeft(): Geeft de referentie naar het linkerkind van het knooppunt terug.

setLeft(TreeNode left): Stelt de referentie naar het linkerkind van het knooppunt in.

getRight(): Geeft de referentie naar het rechterkind van het knooppunt terug.

setRight(TreeNode right): Stelt de referentie naar het rechterkind van het knooppunt in.

BinarySearchTree

**BinarySearchTree**implementeert de functionaliteit van een binaire zoekboom. Het bevat een referentie naar het wortelknooppunt van de boom en biedt methoden voor het invoegen, opvragen en controleren van knooppunten in de boom.

Eigenschappen:

root: Het wortelknooppunt van de boom.

Methoden:

put(String key, String value): Voegt een knooppunt met de opgegeven sleutel en waarde toe aan de boom.

put(TreeNode node, String key, String value): Hulpfunctie om een knooppunt met de opgegeven sleutel en waarde toe te voegen aan de boom.

get(String key): Geeft de waarde terug die overeenkomt met de opgegeven sleutel.

get(TreeNode node, String key): Hulpfunctie om de waarde te krijgen die overeenkomt met de opgegeven sleutel.

containsKey(String key): Controleert of de opgegeven sleutel aanwezig is in de boom.

getAllKeys(): Geeft een lijst terug van alle sleutels in oplopende volgorde.

Extra informatie over Binary Search Tree:

Binary Search Tree (BST) is een geordende, binaire boomgegevensstructuur die wordt gebruikt om gegevens op te slaan en efficiënt op te halen. In een BST zijn de waarden in de knooppunten georganiseerd op een zodanige manier dat voor elk knooppunt de waarden in het linkerkind klein of gelijk zijn aan de waarde van het knooppunt, en de waarden in het rechterkind groter zijn. Dit zorgt voor snelle zoek- en opvraagoperaties.

**NetworkApp**

De klasse NetworkApp is verantwoordelijk voor het accepteren van inkomende verbindingen van clients en het starten van individuele ClientHandler-threads voor elke verbinding. Het luistert op een bepaalde poort en maakt gebruik van socket-programmering om te communiceren met clients.

**NetworkClient**

De klasse NetworkClient vertegenwoordigt een client die verbinding maakt met de server (NetworkApp). Het maakt gebruik van socket-programmering om verbinding te maken met de server en berichten uit te wisselen.

**ClientHandler**

De klasse ClientHandler is verantwoordelijk voor het afhandelen van een individuele clientverbinding op de serverzijde. Elke ClientHandler-instantie wordt uitgevoerd als een afzonderlijke thread en beheert de communicatie met een specifieke client.

**AcceptHandler**

De klasse AcceptHandler is verantwoordelijk voor het continu accepteren van inkomende clientverbindingen op de serverzijde. Het werkt als een aparte thread en start een nieuwe ClientHandler-thread voor elke geaccepteerde verbinding.