

## Klasse UserTree

Allgemeine Beschreibung: Die Klasse UserTree ist für die Speicherung sowie für die Suche der einzelnen User zuständig.

Die User werden in einem tritären Baum gespeichert in der Form:

Der erste User kommt als Wurzel in den Baum. Der nächste User wird dann einsortiert ob sein Name "größer" ist als der der Wurzel. Dabei wird der erste Buchstabe beider Namen verglichen, wenn der neue Nutzer kleiner ist kommt er in den linken Abschnitt, wenn er gleich ist in den mittleren und wenn größer in den rechten Abschnitt des Baumes. Wenn an diesen Abschnitten schon User sind wird mit denen wieder auf die gleiche Art verglichen bis ein leerer Platz gefunden wurde, außer beim mittleren Abschnitt da wird dann mit dem 2 Buchstaben verglichen und wenn der neue Nutzer wieder in den Mittleren abschnitt kommen würde, würde er mit den 3 Buchstaben verglichen usw.

### *Funktionen:*

*addRoot(User newRoot)*

wird zum erstellen des UserTree benötigt

braucht als Parameter ein Objekt der Klasse User.

diese Funktion ist privat da ein UserTree seinen Root nicht wechselt.

*StringtoChar(String userName)*

wird in den Funktionen findUser und addUser und benötigt um die Buchstaben in den Strings vergleichen zu können und ist daher privat.

es wird der Name des Users als String benötigt.

Wandelt ein String in einen char-Array um.

*findUser(String userName)*

benötigt den Namen des Users als String.

ist public damit andere Programme im Baum suchen können.

Sucht einen User in dem UserTree genau so wie der in der allgemeinen Beschreibung.

wenn der User gefunden wurde wird der Pointer auf das Objekt zurückgegeben.

*addUser(User user)*

benötigt ein Objekt der Klasse User

ist public damit andere Programme was zum Baum hinzufügen können.

sortiert den User wie in der allgemeinen Beschreibung ein.

es gibt keinen Rückgabewert

*TreeRekursiv(User node)*

benötigt ein Objekt der Klasse User.

geht rekursiv den ganzen Baum durch und gibt die einzelnen Blätter bzw. User aus.

da diese Funktion rekursiv ist und daher nicht so gut zu handhaben ist, ist sie privat.

*ShowAllUser()*

ist public, startet die TreeRekursiv-Funktion.