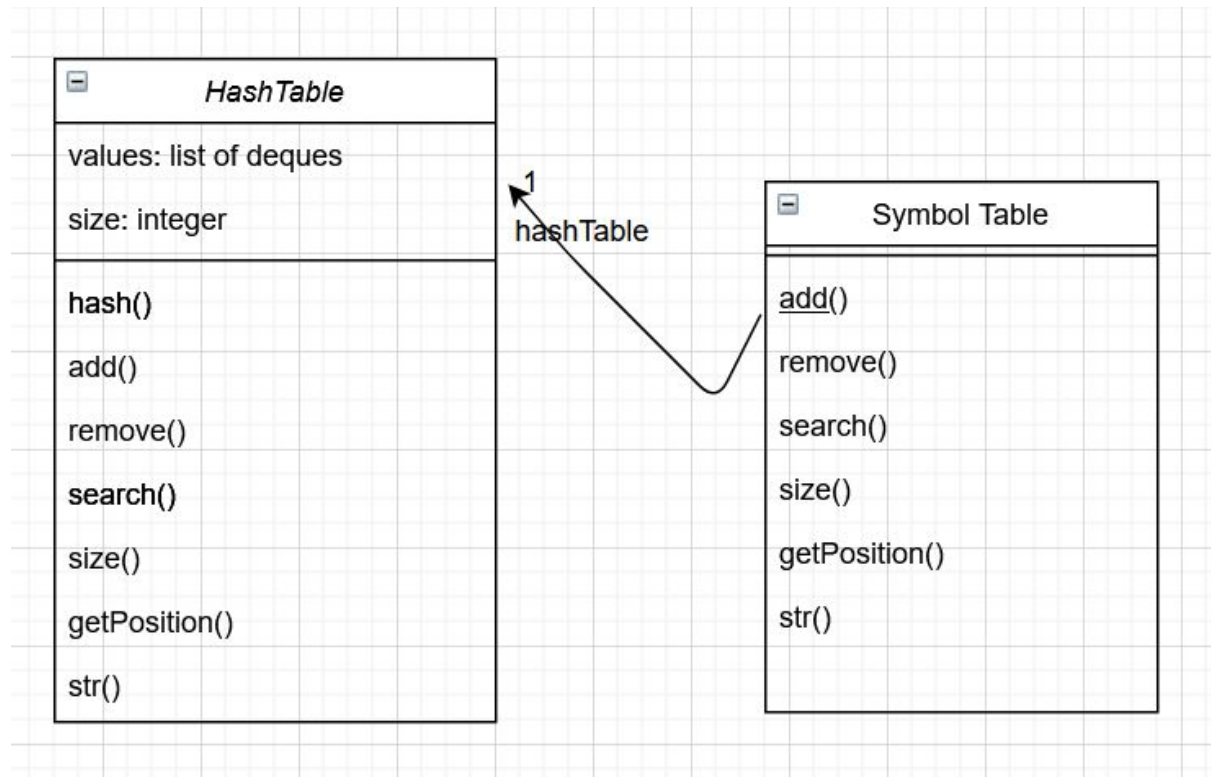


<https://github.com/GeorgianaLoba/FLCD>



I've decided to use Python as a programming language and as a data structure a HashTable.

My HashTable is implemented using a list of deques from the collections library.

The deque (Doubly Ended Queue) is extremely cool since it's a double sided queue which has **O(1)** time complexity for append and pop operations on either end. By using deques, I don't have to bother with resizes and collisions. Personally, I hated doing resizes and the solving of collisions and therefore, the deques seemed like the best solution to avoid them. They resize automatically when a new key is added. As a hash function, I compute the sum of the ascii codes of the characters. Adds are facile, I simply search for the key and if it doesn't exist, I add it in the hash table and modify the size counter. When removing a key, I search for it and if it exists, I remove it and decrement the size counter. The getPosition is more interesting, by using the hash function on the key, I get the position of the deque and then on the deque, I search for the key. Therefore, the getPosition will return a tuple of 2 indexes, the first being the position of the deque in the list and the second being the position of the key inside the deque.