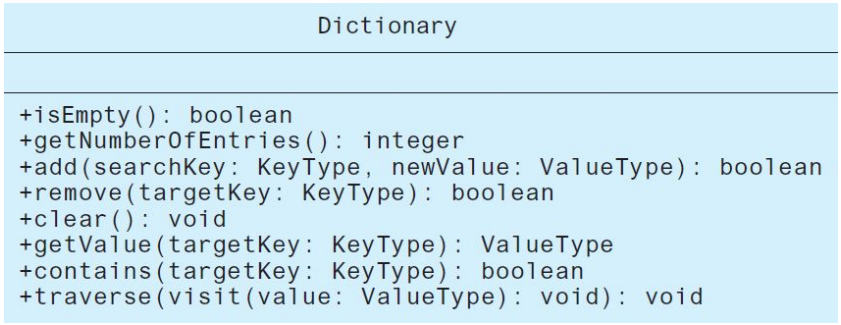
Coding 06: Hash Tables

In this assignment you will create a Dictionary object using a Hash Table (as discussed in class) which will work with the same struct we used for Stacks and Trees (a struct with an integer id and a string for 'data').

Implement all the standard methods as described in figure 18-2 of your text.



Requirements

- Create a dictionary object that contains a hash table which is an array of pointers to the structs.
- Assume the structs to be added to the table are created *outside* the object and a pointer to the struct is passed into the Dictionary for storage in the Hash Table. Note, this is a different definition than shown in the table above.
- Assume when you want to retrieve, delete, or test for the existence of an entry, the id is passed in to locate the position in the hash table.
- Create your hash table array to be size 19.
- Use the Modulo method as a hashing function.
- ~~Use Linear Probing to resolve collisions.~~
- When your program starts, and your object is instantiated, read in data.csv and add all the entries to the table.
- Your main( ) is just your testing ground for your object. You must fully demonstrate all hash table functionality including collisions.
- You must demonstrate what happens when you delete something from the table in a location where something else previously collided; how would you find the thing that previously collided (as discussed in class).
- Demonstrate the full functionality of your program and all your methods.

Extra Credit: Up to 20% extra for implementing Separate Chaining instead of Linear Probing.