Candet Andrei Gabriel

Group 932/1

Lab 3

SOURCE: <https://github.com/cinnamonbreakfast/flcd/tree/main/lab3>

**Symbol Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Preconditions | Postconditions | Observations |
| add | Key and Value must be objects | Value is added to the table | Collision resolution is chaining. A ValueError is thrown if we push the same element again |
| Get | Key must be object | Returns Value according to hash of  Key | If no index could be found for Key in hash, a KeyError is raised. Also, if the value is not within table, a KeyError is raised again |

A HashTable is a data structure which maps keys to values (in our case, a single value for one key). This HashTable uses chaining resolution for collision (list of lists). Hash function is based on hash function from Python (which is hash(key)%length, basically). On each position we store a pair of key & value (as we should get the exact value for a key; for keys having the same hash, their values will be within the same position – chain resolution). Adding the same element will result in a ValueError as the elements should be unique.

HashTable class is wrapped inside Symbol Table class and calls the methods.

**HashTable**

-

\_array

:

+

a

dd(key: Object, value: Object)

+

get(key: Object)

:

Object

hash

+

(

key

:

Object): Number

**Symbol**

**Table**

-

t

able:

Ha

shTable

+

a

dd(key: Object, value: Object)

+

get(key: Object)

:

Object