

1. erstelle eine Hashmap z_h, V
klasse mit size

2. erstelle eine Entry klasse z_h, V
mit Attributen

1. für key

2. für value

3. $Entry(h, V) next$

und ein Konstrukt mit
(h key, V value)

3. in äußere Klasse erstelle
ein Array mit Entry

4. Konstrukt für initialis
der Array

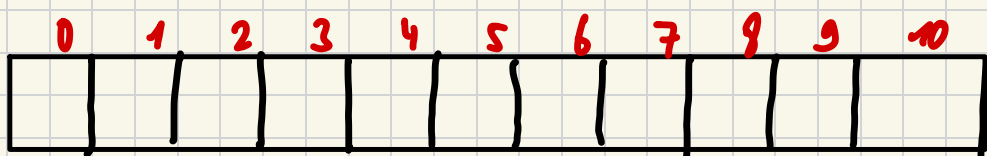
5. put Methode (h key, V value)

um key als int zu bekommen und dann modulo size

5.1 $int hash = key.hashCode \% size$ // z.B. 4

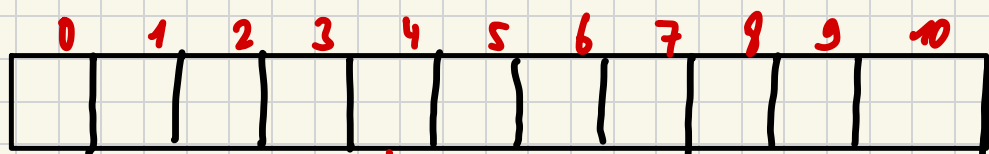
5.2 $Entry z_h, V e = Array[hash]$ 4

5.3



falls $e == null$, dann ist das Index leer, also setzen
 $table[hash] = new Entry(h, V)(key, value)$

falls nicht!

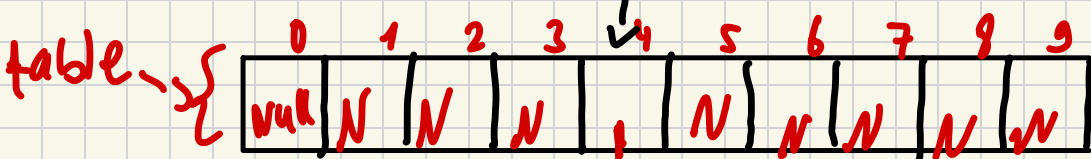


dann laufe weiter

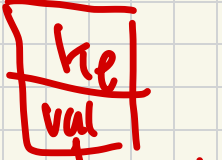
- get methode: public V get (K key)

key = 3009594 010 arraySize = 4

Entry<K,V> temp: table[hash] 114



temp: table[hash]



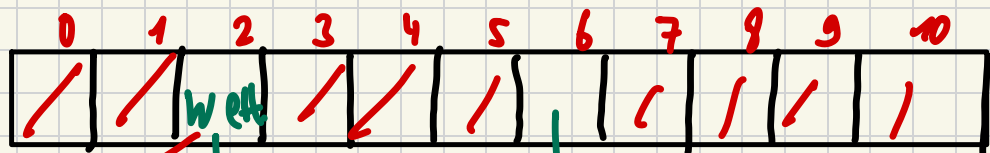
Hier nach dem key Suchen
falls gefunden, seinen wert zurückgeben.



falls gefunden, wird temp. set value
zurückgegeben.

Null

- Print All:



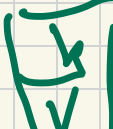
tempc



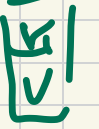
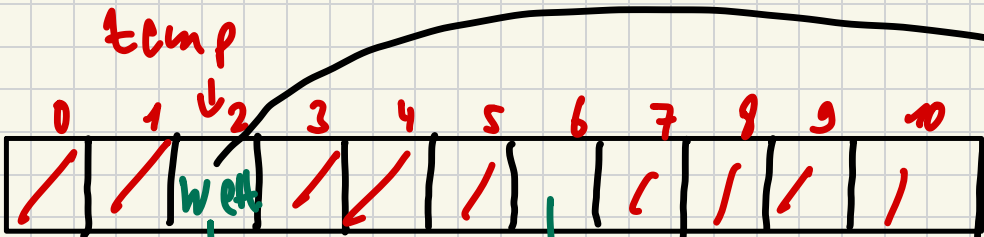
Print out



Null



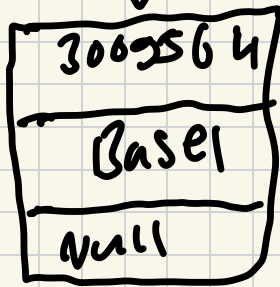
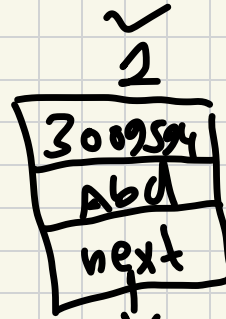
- remove (object key): 2. B lösche index 2



Null

prev →

temp →

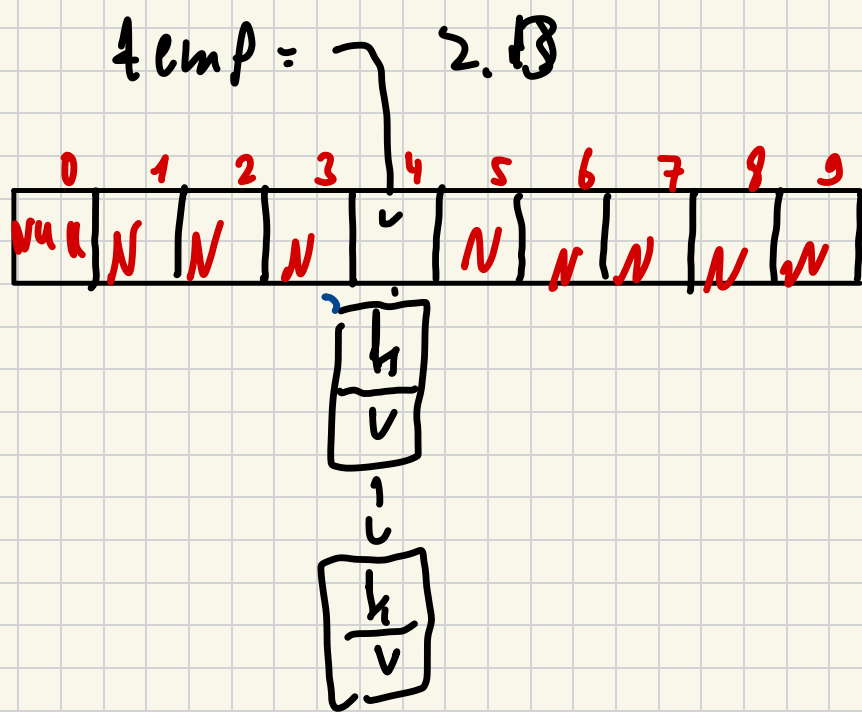


3009594, "Abd"

3009564, "Basel"

remove

- Contains (k, next) order (V values)



- size() :

