| **Hash Table ADT** |
| --- |
| Hash Table = { (),( ), (),...,(),size, capacity} |
| 0 ≤ n ∧ Size( Hash Table) = n (number of key-value pairs in the table) ∧ capacity(Hash Table) = c (total capacity of the table)  0 ≤ n ≤ c (the number of elements is less than or equal to the capacity) |
| * Hash Table(c) − → Hash Table * put Hash Table × key × value → Hash Table * get HashTable × key → value * remove HashTable × key → value * size HashTable → Integer |

| **Hash Table(c)− → Hash Table**  Create a new table hash with a specified initial capacity.  Preconditions: c > 0  Postconditions: Hash Table(c) ht(c) = ∅. |
| --- |

| **put Hash Table × key × value → Hash Table**  Inserts a key-value pair into the hash table.  Preconditions:Hash Table(c) ht(c) ={ (),( ), (),...,()} and pair (k,v) or ht(c) = ∅ and pair (k,v)  Postconditions Hash Table(c) ht(c) ={ (),( ), (),...,(), (k,v)}  or ht(c) =(k,v) |
| --- |

| **get HashTable × key → value**  Gets the value associated with the given key.  Preconditions Hash Table(c) ht(c) i.kv  ht(c) ={ (),( ), (),...,()),(k,v)}  Postconditions: Hash Table(c) ht(c) =(k,v) ∈ HashTable: k=key |
| --- |

| **remove HashTable × key → value**  Removes the key-value (k,v) pair associated with the given key and returns the value.  Preconditions: Hash Table(c) ht(c) i.kv  ht(c) ={ (),( ), (),...,),(k,v)}  Postconditions: Hash Table(c) ht(c) = (k,v) HashTable |
| --- |

| **size HashTable → Integer**  Returns the number of key-value pairs in the hash table  Preconditions: Hash Table ht(c)  Postconditions: size(hc) = n, where n is the of key-value pairs in the table. |
| --- |