**TADs**

|  |  |  |  |
| --- | --- | --- | --- |
| **HashTable TAD** | | | |
| HashTable = { <key1 → value1 >, <key2 → value2 >, <…> , < keyn → valuen > } | | | |
| { inv: keyi = keyk ↔ i = j } | | | |
| Primitive operations: | | | |
| * HashTable | Integer | * HashTable | Constructor |
| * hash | Key | * Integer | Analyzer |
| * getArray | HashTable | * Array | Analyzer |
| * add | HashTable X Key X Value | * String | Constructor |
| * get | HashTable X Key | * Value | Analyzer |
| * set | HashTable X Key X Value | * Value | Modifier |
| * containsKey | HashTable X Key | * Boolean | Analyzer |
| * remove | HashTable X Key | * HashTable | Modifier |
| * clone | HashTable | * HashTable | Constructor |

|  |
| --- |
| **HashTable(***s***)** |
| Constructor for the HashTable class. Initializes a hash table with the specified size.  { pre: TRUE}  { post: *HashTable* = { array = NIL , size = s } |

|  |
| --- |
| **hash(***key***)** |
| Computes the hash code for a given key, ensuring it falls within the array's size boundaries.  { pre: TRUE }  { post: n ≥ 0, n < size} |

|  |
| --- |
| **getArray(***hashTable***)** |
| Retrieves the internal array of hash nodes used in the hash table.  { pre: TRUE }  { post: array } |

|  |
| --- |
| **add(***hashTable, key, value***)** |
| Adds a key-value pair to the hash table using separate chaining for collision resolution.  { pre: TRUE }  { post: array = <key1 → value1 >, <…> , <key→ value>, <…> , < keyn → valuen > } |

|  |
| --- |
| **get(***hashTable, key***)** |
| Retrieves the value associated with a given key from the hash table.  { pre: TRUE }  { post: <keyi = key , valuei > } |

|  |
| --- |
| **set(***hashTable, key, value***)** |
| Updates the value associated with a given key in the hash table.  { pre: <keyi = key , valuei > }  { post: <keyi , valuei = value> } |

|  |
| --- |
| **containsKey(***hashTable, key***)** |
| Checks if the hash table contains a specific key.  { pre: array = <key1 → value1 >, <…> , < keyk → valuek >, <…> , < keyn → valuen > }  { post: True if <key1 → value1 >, <…> , < keyk = key→ valuek >, <…> , < keyn → valuen > } |

|  |
| --- |
| **remove(***hashTable, key***)** |
| Removes a key-value pair from the hash table based on the key.  { pre: array = { <key1 → value1 >, <…>, <keyi = key, valuei >, <…> , = <keyn → valuen > } }  { post: array = { <key1 → value1 >, <…> , = <keyn → valuen > } } |

|  |
| --- |
| **clone(***hashTable***)** |
| Overrides the clone method to create a deep copy of the Hash Table.  { pre: TRUE }  { post: *hashTable*.clone } |