|  |
| --- |
| TAD NodeHash< K, V> |
| NodeHash<K, V> = genericObject.K = Key , genericObject.V = Content ∧ NodeHash<K,V >n.Content≠ NodeHash<K,V >n + 1.Content, NodeHash<K,V>n.next = NodeHash<K,V>n+1 |
| {Inv.: Key >= 0,  NodeHash<K,V >n.Content≠ Null,  If Node<T>.Key = Node<T>n.Key,  then Node<T>.next = Node<T>n  }. |
| Primitive Operations:   * GetKey: NodeHash<K,V> → Key. * SetKey: NodeHash<K,V> x Key→ NodeHash<K,V> * GetValue: NodeHash<K,V> → Content. * SetValue: NodeHash<K,V> x Content → NodeHash<K,V> * GetNext: NodeHash<K,V> → NodeHash<K,V>.Next * SetNext: NodeHash<K,V> x NodeHash<K,V> → NodeHash<K,V> |

|  |
| --- |
| GetKey()  “This operation returns the Key associated to a specific node.”  {pre: NodeHash<K,V> ≠ Null }  {post: Returns NodeHash<K, V>.Key} |

|  |
| --- |
| SetKey(newKey)  “This operation assigns a unique Key value to a Node.”  {pre: NodeHash<K,V> ≠ Null }  {post: NodeHash<K, V>.Key = newKey,  newKey remains unchanged.  } |

|  |
| --- |
| GetValue()  “This operation returns the content associated to a specific node.”  {pre: NodeHash<K,V> ≠ Null  NodeHash<K,V>.Content ≠ Null  }  {post: Returns NodeHash<K, V>.Key} |

|  |
| --- |
| SetValue(newContent)  “This operation assigns or modifies the content associated to a Node.”  {pre: NodeHash<K,V> ≠ Null  }  {post: NodeHash<K, V>.Content = newContent} |

|  |
| --- |
| GetNext()  “This operation returns the Node associated to the pointer of a previous Node.”  {pre: NodeHash<K,V> ≠ Null  }  {post: Returns NodeHash<K, V>.Next} |

|  |
| --- |
| SetNext(nextNode<K,V>)  “This operation sets the Node associated to the pointer of a previous Node.”  {pre: NodeHash<K,V> ≠ Null  NodeHash<K,V>n = NodeHash<K,V>n + 1 }  {post: Returns NodeHash<K, V>.Next = NodeHash<K,V>n + 1 } |