GameLib TADs

HashTable

TAD - HashTable

HasTableTAD = {
Value: <V>, Key: <K>, Nodes<n_1, n_2, … , n_n >, Size: <S>, HashNodeDelet : <HD>,
ElementsNumber: <E>
}

{ inv: HasTableTAD.S > 0 && S ∈ prime number } { inv: HasTableTAD.Key ≠ null } { inv: HasTableTAD.Nodes.Type = value.Type } { inv: HasTableTAD.ElementsNumber <= Size } { inv: HasTableTAD.ElementsNumber <= Size }

Primitive Operations

Create HasTableTAD	true	HasTableTAD
Search	key	Value
Insert	Key x Value	HasTableTAD
Delete	key	HasTableTAD

Métodos

hasTableTAD()
“Creates a HashTable with key = K and value = V”
{ pre: Type: <V> } { pre: Type: <K> }
{ post: HashTable = { HashTable<K,V>.ElementsNumber = 0 }

search()
¨Return a value by a given key”
{pre: Key: <K>}
{post: Value <V>}

insert()
¨Add a value to the HashTable<V, K> ”
{ pre: Value: <V> } { pre: Key: <K> }

{post: HashTable<V, K>}

delete()
“Delete by a given key”
{pre:Key: <K> }
{post: HashTable<V, K>}

Queue

TAD - Queue		

Queue = {
Value: <V>, Node<n>, Size: <S>
}		

{ inv: Node.Type = Value.Type }		
Primitive Operations		
Create Queue	true	Queue
add	Value	Queue
front		Value
dequeue		Value
IsEmpty		boolean

Métodos

queue()
“Creates a Queue with the type of V”
{ pre: Type: <V> }
{ post: HashTable = { Queue<V>.Size= 0 }

front()
¨Return the first value-added in the Queue<V>”
{pre: Queue<V>.Size >= 1}
{post: Value <V>}

add()
¨Add a value to the Queue<V> ”
{ pre: Value: <V> }
{post: Queue<V>}

deqeueue()
“Return and delete the first value-added in the Queue<V>”
{pre: Queue<V>.Size >= 1}
{post: Value <V>}

Stack

TAD - Stack		

Stack = {
Value: <V>, Node<n>, Size: <S>
}		

{ inv: Node.Type = Value.Type }		
Primitive Operations		
Create Stack	true	Stack
Push	Value	Stack
Top		Value
Pop		Value
IsEmpty		boolean

Métodos

stack()
“Creates a Stack with the type of V”
{ pre: Type: <V> }
{ post: HashTable = { Satack<V>.Size= 0 }

push()
¨Add a value to the Stack<V> ”
{ pre: Value: <V> }
{post: Stack<V>}

top()
“Return the last value-added in the Stack<V>”
{pre: Stack<V>.Size >= 1}
{post: Value <V>}

pop()
“Return and delete the last value-added in the Stack<V>”
{pre: Stack<V>.Size >= 1}
{post: Value <V>}