TAD TABLES

By: Juan Felipe Jojoa Crespo A00382042

Felipe Rojas Prado A00393918

|  |
| --- |
| TAD <HashTable> |
|  |
| HashTable = {SIZE, hashing, keyValidation, Htable}  Invariant: {key, equalsIgnoreCase, Htable(allKeys)} |
| hasKey(): HashTable \* key -> boolean  hashTableInsertion(SIZE, hashing, keyValidation): -> new HashTable  setData(key, value): (new Hashtable \* key ^ value) -> void  getData(key): new HashTable -> value  removeData(key): (new HashTable \* key) -> boolean |

|  |
| --- |
| hashTableInsertion()(constructor) |
| It is responsible for inserting and initializing the hashtable. |
| precondition: SIZE, hashing, keyValidation |
| postcondition: Type String ALERT MESSAGE: The hashtable has been inserted correctly. |

|  |
| --- |
| setData(HashTable) Modificadora |
| It is in charge of putting a value to the key or identifier in the hashtable. |
| precondition: key, value & newHashTable != null |
| postcondition: Type String ALERT MESSAGE: The value has been set correctly. |

|  |
| --- |
| getData(HashTable) Analizador |
| Gets the value of a key that has already been set in the hashtable. |
| Precondition: key |
| postcondition: value |

|  |
| --- |
| removeData(HashTable) Modificadora |
| It is in charge of formatting or deleting a value set in a key previously in the hashtable. |
| precondition: key  condition: key!=null & HashTable!=null |
| postcondition: Type String ALERT MESSAGE: The value of the key has been removed to the initial state. |

|  |
| --- |
| hasKey(HashTable) Analizador |
| It is the validation that is in charge of seeing if a key is already set. |
| precondition: key  condition: key!=null & HashTable!=null |
| postcondition: TRUE Or FALSE, Boolean if founded. |

|  |
| --- |
| TAD <PriorityQueue> |
|  |
| priorityQueue = {SIZE, comparatorFunction (Interface comparable)}  invariant: {comparatorFunction(key1,key2) = True} |
| sizeValidator(): -> passengerData  queueEnd(): -> getNode  priorityQueueInsertion(SIZE): -> priorityQueue  removeFromQueue(): priorityQueue -> getNode  addQueue(passengerData): priorityQueue \* getNode -> void |

|  |
| --- |
| prioriryQueueInsertion()(constructor) |
| It is in charge of inserting and initializing a priority queue. |
| precondition: SIZE  condition: SIZE!=null |
| postcondition: Type String List: priorityQueuePassengerData |

|  |
| --- |
| sizeValidator(PriorityQueue) constructor |
| It is the function responsible for determining the size of the queue. |
| precondition: Can't be null. (!=null) |
| postcondition: Type String ALERT MESSAGE: Queue size set correctly. |

|  |
| --- |
| addQueue(PriorityQueue) Modificadora |
| It is in charge of adding data (Of the passengers) to the queue. |
| precondition: passengerData |
| postcondition: Type String List: priorityQueuePassengerData + new added passengerData. |

|  |
| --- |
| removeFromQueue(PriorityQueue) Modificadora |
| It is responsible for removing data (from passengers) to the queue. |
| Precondition: passengerDataID  condition: passengerData!=null |
| postcondition: Type String List: priorityQueuePassengerData - removed passengerData. |

|  |
| --- |
| queueEnd(PriorityQueue) Analizador |
| It is in charge of printing the queue of the priorityQueue. |
| Precondition: passengerData  condition: passengerData!=null |
| postcondition: Type String List: queue |

|  |
| --- |
| TAD <HeapSort> |
|  |
| HeapSort = {ancestor,rightNode, leftNode, SIZE}  Invariant: {ancestor > rightNode & ancestor > leftNode} |
| heapSortInsertion(SIZE): -> HeapSort  addHeap(passengerData): HeapSort \* Node(Verifiy right and left validation) -> void  searchHeap(keyHeap): -> Node  calculateSize(): - > passengerData  removeHeap(keyHeap): HeapSort -> Node(Verifiy right and left validation and balance) |

|  |
| --- |
| heapSortInsertion()(constructor) |
| It is responsible for inserting and initializing the heap in the system. |
| precondition: SIZE  condition: SIZE!=null |
| postcondition: Type String List: new Heap |

|  |
| --- |
| addHeap(HeapSort) Modificadora |
| It is in charge of adding the data in the Heap. |
| precondition: passengerData |
| postcondition: Type String List: Heap |

|  |
| --- |
| removeHeap(HeapSort) Modificadora |
| It is in charge of removing the data in the Heap. |
| Precondition: passengerData  condition: passengerData!=null |
| postcondition: Type String List: Updated Heap |

|  |
| --- |
| searchHeap(HeapSort) Analizador |
| It is in charge of finding a specific node in the Heap (Passenger). |
| precondition: passengerData  condition: passengerData!=null |
| postcondition: Type String ALERT MESSAGE: Node (Passenger) successfully located on the Heap. |

|  |
| --- |
| calculateSize(HeapSort) Analizador |
| It is in charge of calculating the size of the heap in the system once the nodes have been established. |
| precondition: Heap cant be null. |
| postcondition: Type String ALERT MESSAGE: Calculate according to the nodes the size of the Heap in integers. |