

MINHEAP		
MinHeap= {Heap= <T <sub>1</sub> , T <sub>2</sub> ,...,T <sub>n</sub> >}		
{Inv: $\forall_i, T_1 < T_i$ }		
MinHeap:		→ MINHEAP
insert:	MINHEAP x T	→ MINHEAP
peekMin:	MINHEAP	→ T
isEmpty:	MINHEAP	→ BOOLEAN
addElements:	MINHEAP x ARRAYLIST<T>	→ MINHEAP
extractMin:	MINHEAP	→ T
getHeap:	MINHEAP	→ ARRAYLIST<T>

**Constructor Operations:**

<b>MinHeap()</b>
“Creates a new MinHeap”
{pre: TRUE}
{post: MinHeap= {Heap= <ArrayList<T>>}}

**Modifying Operations:**

<b>insert(Heap, T)</b>
“Inserts a new element into the heap”
{pre: MinHeap={Heap= <T <sub>1</sub> , T <sub>2</sub> ,...,T <sub>n</sub> >}, T=<content>}
{post: MinHeap={Heap= <T <sub>1</sub> , T <sub>2</sub> ,...,T <sub>n</sub> ,T <sub>n+1</sub> > }}

<b>addElements(MaxHeap, ArrayList&lt;T&gt;)</b>
“Adds a new group of elements to the MinHeap (with the certainty that the process maintains the invariant of the structure)”

```
{pre: MinHeap={Heap= < T1, T2,...,Tn> } , Elements={< R1, R2,...,Rn>} }  
{post: MinHeap={Heap= < T1, T2,Ti,Tj,...,Tn , R1, R2,...,Rn >}}
```

### **extractMin(Heap)**

“Returns the smallest element (the first one, knowing that the structure is always following its invariant) and removes it from the heap”

```
{pre: MinHeap={Heap=< T1, T2,...,Tn>} }  
{post: T1=<content>}
```

## **Analyzing Operation:**

### **peekMin()**

“Returns the smallest element (the first one, knowing that the structure is always following its invariant) of the heap”

```
{pre: MinHeap={Heap= < T1, T2,...,Tn>} }  
{post: T1=<content>}
```

### **isEmpty()**

“Returns whether the heap structure has any elements or not”

```
{pre: MinHeap={Heap= < ArrayList<T>>} }  
{post: TRUE if Heap contains no elements, else FALSE}
```

### **getHeap(Heap)**

“Returns a list with all the elements contained in Heap (with the elements organized following the invariant)”

```
{pre: MinHeap={Heap= < T1, T2,...,Tn>} }  
{post: Heap= < T1, T2,...,Tn> }
```