

QUEUE TEST

Stage

Name	Class	Stage
setupEmptyQueue	QueueTest	An empty queue ready to enqueue elements

Design of test cases

Goal of the test: Verify that the method enqueues an element and also dequeues it successfully				
Class	Method	Stage	In's	Out's
Queue	enqueue	setupEmptyQueue	e1, e2, e3	A queue with the elements e1, e2 and e3 in it
Queue	dequeue	setupEmptyQueue		Returns the element e1. FIFO order(First in- first out)

Goal of the test: Verify that the peek method returns the element on front without getting it out of the queue				
Class	Method	Stage	In's	Out's
Queue	peek	setupEmptyQueue	e1, e2	Returns the element e1

Goal of the test: Verify that the method isEmpty is working correctly				
Class	Method	Stage	In's	Out's
Queue	isEmpty	setupEmptyQueue		Returns true
Queue	isEmpty	setupEmptyQueue	e1	Returns false

Goal of the test: Verify that the method gives back the size of the queue				
Class	Method	Stage	In's	Out's
Queue	size	setupEmptyQueue		0
Queue	size	setupEmptyQueue	e1, e2, e3	3

STACK TEST

Stage

Name	Class	Stage
setupEmptyStack	StackTest	An empty stack

Design of test cases

Goal of the test: Verify that we can push an element into the stack and pop it too				
Class	Method	Stage	In's	Out's
Stack	push	setupEmptyStack	e1, e2, e3	A stack with the three elements in it
Stack	pop	setupEmptyStack		Returns the element e3. LIFO order (Last in - First out)

HASHTABLE TEST

Stage

Name	Class	Stage
initEmpty7Slots	HashTableTest	An empty hash table with 7 slots
initEmpty1Slot	HashTableTest	An empty hash table with 1 slot

Design of the cases

Goal of the test: Verify that we can correctly add an element into the hash table				
Class	Method	Stage	In's	Out's
HashTable TestKey	add	initEmpty7Slots	key1, val1	true

Goal of the test: Verify that we can correctly remove an element of the hash table and that removing one element should not affect the others				
Class	Method	Stage	In's	Out's
HashTable TestKey	remove	initEmpty7Slots add(key1, val1)	key1	Returns the element that just has removed (val1)
HashTable TestKey	Remove getValueOf	initEmpty7Slots add(key1, val1) add(key2, val2)	key1	Removes the val1 and checks that the val2 still exists on the hash table

Goal of the test: Verify that the method getValueOf returns correctly the value of an given key				
Class	Method	Stage	In's	Out's
HashTable TestKey	getValueOf	initEmpty7Slots add(key1, val1)	key1	val1

Goal of the test: Verify that the method isEmpty correctly checks when a hash table is empty or not

Class	Method	Stage	In's	Out's
HashTable TestKey	isEmpty	initEmpty7Slots		true
HashTable TestKey	isEmpty	initEmpty7Slots add(key1, val1)		false

Goal of the test: Verify that the method isFull correctly checks when a hash table is full or not

Class	Method	Stage	In's	Out's
HashTable TestKey	isFull	Init1EmptySlot add(key1, val1)		true
HashTable TestKey	isFull	remove(key1)		false

HEAP TEST

Stages

Name	Class	Stage
Setup7Empty	HeapTest	Creates an empty heap with 7 slots. Also creates an array filled with 7 nodes

Design of test cases

Goal of the test: Verify that the method add is correctly adding an element on the heap and checks if the method isHeap is working

Class	Method	Stage	In's	Out's
Heap	add	setup7Empty	Node[]	true
Heap	isHeap			true

Goal of the test: Verify that the method to remove elements of the heap is working

Class	Method	Stage	In's	Out's
Heap	remove	setup7Empty add(Node[])	[]temp	Every n of []temp equals to the val of Node[] to remove

SORT TESTS

Stage

Name	Class	Stage
setup	SortTest	Creates two arraylists of Integers, one is disordered and the other one is in order. Also creates a comparator of Integers to pass as parameter to the sort methods

Design of test cases

Goal of the test: Verify that the merge sort is working				
Class	Method	Stage	In's	Out's
Sort	mergeSort	setup	{1, 6, 17, 0, 10, 2, 2}	{0, 1, 2, 2, 6, 10, 17}

Goal of the test: Verify that the bubble sort is working				
Class	Method	Stage	In's	Out's
Sort	bubbleSort	setup	{1, 6, 17, 0, 10, 2, 2}	{0, 1, 2, 2, 6, 10, 17}

Goal of the test: Verify that the heap sort is working				
Class	Method	Stage	In's	Out's
Sort	heapSort	setup	{1, 6, 17, 0, 10, 2, 2}	{0, 1, 2, 2, 6, 10, 17}

Goal of the test: Verify that the quick sort is working				
Class	Method	Stage	In's	Out's
Sort	quickSort	setup	{1, 6, 17, 0, 10, 2, 2}	{0, 1, 2, 2, 6, 10, 17}