TEST CASES FOR PRIORITYQUEUE

deleteStandardTest

Test Case	deleteStandardTest
Class	PriorityQueue
Method	delete(int key)
Scenario	Delete the node with key=2 from a queue of size 4
Inputs	PriorityQueue with nodes A(1), B(2), C(3), D(4)
Expected Output	Maximum node has key=4

deleteLimitsTest

Test Case	deleteLimitsTest
Class	PriorityQueue
Method	delete(int key)
Scenario	Delete a node from an empty queue
Inputs	PriorityQueue with size 5
Expected Output	Maximum node is null

deleteInterestingTest

Test Case	deleteInterestingTest
Class	PriorityQueue
Method	delete(int key)
Scenario	Delete the node with key=0 from a queue of size 5 with nodes A(3), B(1), C(2)
Inputs	PriorityQueue with nodes A(3), B(1), C(2)
Expected Output	Maximum node has key=2

insertStandardTest

Test Case	insertStandardTest
Class	PriorityQueue
Method	insert(PriorityQueueNode<K, V> node)
Scenario	Insert four nodes with priorities 2, 3, 1, and 4 into a queue of size 5
Inputs	PriorityQueue with no nodes
Expected Output	Maximum node has key=4

insertLimitsTest

Test Case	insertLimitsTest
Class	PriorityQueue
Method	insert(PriorityQueueNode<K, V> node)
Scenario	Attempt to insert a node into a full queue
Inputs	PriorityQueue with one node
Expected Output	Insertion fails

insertInterestingTest

Test Case	insertInterestingTest
Class	PriorityQueue
Method	insert(PriorityQueueNode<K, V> node)
Scenario	Insert four nodes with priorities 1, 2, 2, and 4 into a queue of size 5
Inputs	PriorityQueue with no nodes
Expected Output	Maximum node has key=4

Class	Method	Scenario	Inputs	Expected Outputs
PriorityQueue	maximum()	Standard	Inserted elements: (A, 10), (B, 7), (C, 12), (D, 5) into a priority queue of capacity 4	Element with highest priority: C

Class	Method	Scenario	Inputs	Outputs
PriorityQueue	maximumLimitTest()	Test when queue is empty	PriorityQueue with capacity 0	null

Class	Method	Scenario	Inputs	Expected Output
	maximumInterestingTest()	Extracted max is not root	PriorityQueue with 4 elements "A": 10, "B": 7, "C": 12, "D": 5	"A"

Class	Method	Scenario	Inputs	Expected Output
PriorityQueue<String, Integer>	extractMaxStandardTest()	Standard Case	Elements: ("A", 10), ("B", 7), ("C", 12), ("D", 5)	"C"

Test case	Class	Method	Scenario	Inputs	Expected Outputs
extractMaxLimitTest	PriorityQueue	extractMax()	Attempting to extract maximum element from an empty priority queue	An empty priority queue	`null`

Test Case	Class	Method	Scenario	Inputs	Outputs
extractMaxInterestingTest	PriorityQueue	extractMax	Removing two maximum elements from a priority queue of size 4	PriorityQueue with elements: ["A" (10), "B" (7), "C" (12), "D" (5)]	Maximum element of the queue is "B"

Test Method	Class	Method	Scenario	Inputs	Outputs
quickSortMajorMinorStandardTest	PriorityQueue<String, Integer>	quickSortMajorMinor(int, int)	Test sorting elements	Insert 5 nodes: "A" with priority 3, "B" with priority 5, "C" with priority 2, "D" with priority 1 and "E" with priority 4.	Extract max from queue five times: "B", "E", "A", "C", and "D" respectively.

Test Case	Class	Method	Scenario	Inputs	Expected Output
quickSortMajorMinorLimitTest	PriorityQueue	quickSortMajorMinor	Sorting a queue with only one element	A=10	A=null

Test case	Class	Method	Scenario	Input	Expected Output
quickSortMajorMinorInterestingTest	PriorityQueue<String, Integer>	quickSortMajorMinor(int, int)	Sort a priority queue using quicksort in descending order	"A" (2), "B" (5), "C" (1), "D" (5), "E" (3)	"D", "B"

Class Name	Method Name	Scenario	Inputs	Expected Output
PriorityQueue	quickSortMinorMajor	Testing with a queue of size 5 with nodes having different priorities	Inserting nodes with priorities A-3, B-5, C-2, D-1, and E-4 into the queue	Extracting nodes in decreasing order of priority (B, E, A, C, D)

Class	Method	Scenario	Input	Expected Output
PriorityQueue	quickSortMinorMajorLimitTest	Test that sorting a priority queue with one element using the QuickSortMinorMajor method returns the element	A node with value "A" and priority 10 is inserted into the priority queue with size 1.	The priority queue should extract the node with value "A" and priority 10, and the queue should be empty.

Class name	Method name	Scenario	Inputs	Expected output
PriorityQueue	quickSortMinorMajorInterestingTest	Valid input of 5 nodes	(A,2), (B,1), (C,5), (D,5), (E,3)	The nodes with keys 5 and 5 are extracted in order

Class Name	Method Name	Scenario	Inputs	Expected Output
PriorityQueue	isEmpty	PriorityQueue is empty	PriorityQueue object initialized with capacity of 5	true

Class Name	Method Name	Scenario	Inputs	Expected Output
PriorityQueue	isEmpty	Empty queue	Queue of size 0	true

Class Name	Method Name	Scenario	Inputs	Expected Output
PriorityQueue	isEmptyInteresting	Priority queue is created with capacity 5 and an element is inserted	new PriorityQueue<>(5); queue.insert(new PriorityQueueNode<>("A", 20));	false

Class name	Method name	Scenario	Inputs	Expected Output
PriorityQueue	occupiedSize	Standard	PriorityQueue instance with capacity 5	0

Class Name	Method Name	Scenario	Inputs	Expected Output
PriorityQueue	occupiedSizeLimits	Limit scenario with 3 elements	Priorities: [20, 25, 10]	3

Class name	Method name	Scenario	Inputs	Expected Output
PriorityQueueTest	occuiedSizeInteresting	Test occupiedSize method with interesting scenario	-	1