heap.peek();

heap.size();

import java.util.Arrays;

import java.util.PriorityQueue;

public class TestHeap {

public static void main(String[] args) {

double[] arr = {9,0,5,1,2,8,4,3,7,6};

System.out.println("arr = "+ Arrays.toString( arr ) );

PriorityQueue<Double> minHeap = new PriorityQueue<Double>();

for(double num : arr)

minHeap.offer( num );

System.out.print("minHeap = ");

while( !minHeap.isEmpty() )

System.out.print( minHeap.poll() + ", ");

PriorityQueue<Double> maxHeap = new PriorityQueue<Double>(11, new MyMaxHeapComparator() ); // no type needed for MyMaxHeapComparator() here.

for(double num : arr)

maxHeap.offer( num );

System.out.print("\nmaxHeap = ");

while( !maxHeap.isEmpty() )

System.out.print( maxHeap.poll() + ", ");

}

}

import java.util.Comparator;

public class MyMaxHeapComparator implements Comparator<Double>{

public int compare(Double d1, Double d2){

if( d1 > d2)

return -1;

else if( d1 == d2)

return 0;

else

return 1;

}

}