

Übung 4 – Priority Queue / Binary Heap

In dieser Aufgabe soll eine Priority Queue mit einem binären Heap realisiert werden. Diese Priority Queue soll folgende Schnittstelle implementieren:

```
interface PriorityQueue {  
    void insert(Object obj, long priority) throws QueueFullException;  
    // "enqueue": Objekt obj mit Priorität priority in die Warteschlange einfügen.  
    // Konvention: Je kleiner der Wert von priority, desto höher ist die Priorität.  
  
    Object deleteMin() throws QueueEmptyException;  
    // "dequeue": Objekt mit höchster Priorität (d.h. kleinstem Wert priority) aus  
    // Warteschlange entfernen und zurückgeben.  
  
    int size();  
    boolean isEmpty();  
    boolean isFull();  
    void clear();  
}
```

Ein Programmgerüst für die Klasse *Heap* ist in der Datei *Heap.java* vorgegeben (siehe Datei *u4.zip*). Zu *insert* und *deleteMin* wurden im Unterricht die Algorithmen vorgestellt, die hier verwendet werden sollen.

Damit Sie ihre Implementierung testen können, steht ein Hilfsprogramm *HeapTest* zur Verfügung. Dieses erlaubt die interaktive Erfassung von Prioritäten von der Standardeingabe und gibt den Heap als Array von Prioritäten (Test von *insert*) und die Prioritäten nach Grösse geordnet aus (Test von *deleteMin*) aus.

Damit der Heap als Feld ausgegeben werden kann muss zusätzlich zu den im Interface vorgegebenen Methoden in der Klasse *Heap* die Methode *toLongArray()* implementiert werden.

```
public long[] toLongArray() {  
    // Erzeugt ein neues long[] Array und kopiert dann die Werte der Prioritäten aus dem Heaparray .  
    // Die Grösse des zurückgegebenen Feldes soll der Anzahl Elemente in der Queue entsprechen  
    // (= size()), d.h., das dummy Element soll nicht berücksichtigt werden. An der Position 0 soll  
    // die kleinste Priorität (= Priorität des Wurzelelementes) stehen.  
}
```

Testen sie ihre Lösung ausgiebig und zeichnen Sie die von *HeapTest* als Array ausgegebenen Heaps als Baumstruktur auf, um die Ordnungs- und Strukturbedingungen im Heap von Hand überprüfen zu können (oder erweitern Sie das Testprogramm damit dieses diese Prüfung vornimmt).