

Übung 6: Heap

Abgabetermin: 08.05.2012

Name: _____

Matrikelnummer: _____

Gruppe: G1 Di 10:15 G2 Di 12:45

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Punkte
Aufgabe 1	24	<input type="checkbox"/>	Java-Programm Testfälle und Ergebnisse	Java-Programm	<input type="checkbox"/>	

Aufgabe 1: Prioritätswarteschlange für Zeichen (24 Punkte)

Implementieren Sie eine Prioritätswarteschlange für Zeichen mit einem Heap. Größere Zeichen sollen höhere Priorität als kleinere haben, d.h. 'z' kommt vor 'a'. Die Schnittstelle ist durch die abstrakte Klasse *PriorityQueue* gegeben: *offer* fügt ein Zeichen ein; *poll* liefert das größte Zeichen und entfernt es; *size* liefert die Anzahl der Zeichen; *iterator* liefert einen Iterator (mit Schnittstelle *Iterator*) mit dem die Prioritätswarteschlange in Heapordnung (d.h. vom größten zum kleinsten Zeichen) durchlaufen werden kann; *clear* löscht alle Zeichen; *peek* liefert das größte Zeichen ohne es zu entfernen; *remove* entfernt das erste Vorkommen des Zeichens, falls es vorkommt.

```
package at.jku.ssw;
public abstract class PriorityQueue {
    public abstract void offer(char value);
    public abstract char poll();
    public abstract int size();
    public abstract Iterator iterator();
    public abstract void clear();
    public abstract char peek();
    public abstract boolean remove(char value);
}
```

```
public abstract class Iterator {
    public abstract boolean hasNext();
    public abstract char next();
}
```

Implementieren Sie die Klassen *ArrayPriorityQueue* und *ArrayPriorityQueueIterator* im Paket *at.jku.students*.

```
package at.jku.students;
public class ArrayPriorityQueue
    extends PriorityQueue {
    char[] values = new char[1];
    int count = 1;
    public String makeDot() {
        return DotMaker.makeDotForHeap(
            Arrays.copyOf(values, count));
    }
    ...
}
public class ArrayPriorityQueue Iterator
    extends Iterator {
    ...
}
```

```
PriorityQueue pq
    = new ArrayPriorityQueue();
pq.offer('T');
pq.offer('O');
pq.offer('G');
pq.offer('X');
pq.offer('S');
makeDot((ArrayPriorityQueue) pq, "Test.dot");
Out.print(pq.size() + ": ");
while (pq.size() > 0) {
    Out.print(" " + pq.poll());
} // Ausgabe 5: XTSOG
```

Implementierungshinweise:

- Verwenden Sie ein Array um den Heap zu implementieren. Lassen Sie das Array am Index 0 für ein Dummy-Element leer. Lassen Sie das Array dynamisch wachsen, indem Sie bei Bedarf die Länge verdoppeln.

Abzugeben ist: Java-Programm, Testfälle

