

Praktikum zu  
**Einführung in die Informatik für  
LogWings und WiMas**  
Wintersemester 2014/15

**Übungsblatt 9**

Bearbeitungszeit:

05.01.15 -

09.01.15

**Aufgabe 9.1 – Felder als Datensatz — Heaps**

**Bearbeitungszeit: 20 Minuten**

Zeichnen Sie die Baumstruktur eines Min-Heaps nach Anwendung folgenden Heap-Operationen auf einem zunächst leeren Heap. Zeichnen Sie einen Baum für jeden Teilschritt, das heißt auch jeden Schritt, in welchem nicht die Heap-Eigenschaft erfüllt ist:

- Einfügen: 5
- Einfügen: 8
- Einfügen: 7
- Einfügen: 3
- Einfügen: 6
- Einfügen: 4
- Einfügen: 1
- 7 mal Minimum entfernen

Notieren Sie zudem die Zahlen in der Reihenfolge, in der sie entfernt werden.

**Aufgabe 9.2 – Bäume und Felder: Verwaltung**

**Bearbeitungszeit: 20 Minuten**

Wie die in einem Feld gesammelten Daten interpretiert werden, ist dem Programmierer überlassen. In der Vorlesung haben Sie gelernt, wie die Daten in einem Feld als Baum interpretiert werden können. Um mit Daten in Feldern einfacher umgehen zu können, werden wir zunächst die Klasse **ArrayUtil** um ein paar Funktionen erweitern.

Schreiben Sie in Ihrer Klasse **ArrayUtil** folgende Methoden:

- Eine Methode, die ein `int[]` und zwei Indizes als `int` entgegen nimmt, und die Werte an den übergebenen Stellen miteinander tauscht. Wenn einer der übergebenen Indizes außerhalb der Grenzen des Feldes liegt, soll nichts geschehen.

**Beispiel:** `[3,8,5,9], 1, 3 ⇒ [3,9,5,8]`

- Eine Methode, die ein `int[]` und einen `int` entgegen nimmt und den übergebenen Wert an die erste Stelle im Feld schreibt, die eine 0 enthält. Wenn keine solche Stelle existiert, soll nichts geschehen.

**Beispiel:** `[2,4,0,0,0,0], 7 ⇒ [2,4,7,0,0,0]`

- Eine Methode, die ein `int[]` und einen Index als `int` entgegen nimmt und den Wert des Feldes an diesem Index zurück gibt. Dabei soll der Wert an dieser Stelle auf 0 gesetzt werden. Sollte der Index nicht in den Grenzen des Feldes liegen, soll nichts geschehen und 0 zurück gegeben werden.

**Beispiel:** `[4,9,10,2,1,0,0]`, `3`  $\Rightarrow$  `[4,9,10,0,1,0,0]`

### Aufgabe 9.3 – Heaps in einer statischen Implementierung

**Bearbeitungszeit: 40 Minuten**

**Wichtig: Alle Elemente unserer Heaps sind von 0 verschieden!**

**Hinweis: Da wir das erste Element des Feldes nicht verwenden, muss das Feld um 1 größer sein, als die Anzahl der zu sortierenden Zahlen!**

Schreiben Sie eine neue Klasse namens `StaticHeapTest`. Verwenden Sie während der Implementierung die Funktionen ihrer Klasse `ArrayUtil`, sowie die der auf der Veranstaltungsseite vorgegebenen Klasse `IOUtil`, welche es Ihnen erlaubt, über statische Methoden vom Benutzer Daten abzufragen. Lesen Sie sich die Kommentare dieser Klasse durch, um ihre Funktionsweise zu verstehen. Kopieren Sie die Klasse `IOUtil` in ihr `util`-Paket.

Die **main**-Methode Ihrer Test-Klasse soll folgendes tun:

- Den Benutzer nach der Anzahl der zu sortierenden Zahlen fragen.
- Ein `int`-Feld passender Größe erstellen.
- Den Benutzer anschließend entsprechend oft nach einer Zahl fragen, welche in dem Feld abgespeichert wird.
- Nach jeder eingelesenen Zahl soll in dem Feld die Heap-Eigenschaft hergestellt werden.
- Nach dem einlesen entsprechend vieler Zahlen sollen die eingelesenen Zahlen in sortierter Reihenfolge ausgegeben werden, indem Sie aus dem Heap entfernt werden.

Schreiben Sie zur Verwaltung Ihres Heaps zwei Methoden, die wie in der Vorlesung die Knoten im Baum verschieben und so die Heap-Eigenschaft im Feld garantieren sollen:

- Eine namens **ascent**, welche den Heap und den zu betrachteten Index eines Knotens als Parameter entgegen nimmt. Mithilfe der Methoden Ihrer `ArrayUtil`-Klasse soll sie den Knoten an eine passende Position im Heap sortieren, indem der Knoten im Baum nach *oben* verschoben wird. Orientieren Sie sich bei der Implementierung dieser Methode an die Folie 51 aus Kapitel 5.2.
- Eine namens **decent**, welche den Heap und den zu betrachtenden Index eines Knotens als Parameter entgegen nimmt. Mithilfe der Methoden Ihrer `ArrayUtil`-Klasse soll sie den Knoten an eine passende Position im Heap sortieren, indem der Knoten im Baum nach *unten* verschoben wird. Orientieren Sie sich bei der Implementierung dieser Methode an die Folie 74 aus Kapitel 5.2.

Überlegen Sie sich zudem, ob Sie den Hilfsmethoden einen zusätzlichen Parameter übergeben, der die aktuelle Anzahl an Elementen im Heap repräsentiert, oder ob Sie die Blätter des Baumes auf eine andere Art und Weise finden können<sup>1</sup>.

### Aufgabe 9.4 – Heaps als Objekte

**Bearbeitungszeit: 40 Minuten**

Auf der Veranstaltungswebseite finden Sie 2 Java-Dateien zu dieser Aufgabe: Die Klasse `Heap` und die Klasse `HeapTest`. Die Datei `Heap` beinhaltet den Quellcode einer objektorientierten Version der Heapverwaltung. Lesen Sie ihn und versuchen Sie die Arbeitsweise nachzuvollziehen.

Erstellen Sie in der **main**-Methode der Test-Klasse einen neuen Heap der Kapazität 7 und führen Sie die Schritte aus Aufgabe 9.1 auf diesem Heap mit den entsprechenden Methoden aus. Geben Sie die entfernten Werte auf der Konsole aus.

Der Quellcode der Klasse **Heap** beinhaltet genau einen Fehler, der für eine fehlerhafte Ausgabe führt. Versuchen Sie ihn zu finden und zu beheben. Experimentieren Sie dazu mit dem Debugger und dem Quellcode. Der Praktikumsleiter wird Ihnen die Lösung pünktlich genug verraten, damit Sie die Implementierung ihres Tests überprüfen können.