

## Übung 04: Heaps

Abgabetermin: 06.05.2008 12:00

Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Gruppe:  G1 (Wolfinger)

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Pkte
Aufgabe 04.1	18	<input type="checkbox"/>	Java-Programm Testfälle und Ergebnisse	Java-Programm	<input type="checkbox"/>	
Aufgabe 04.2	6	<input type="checkbox"/>	Heaps in Binärbaum- und Arraydarstellung	-	<input type="checkbox"/>	

### Aufgabe 04.1: Product Backlog als Heap

Scrum ist eine agile Methode zur Softwareentwicklung. Zunächst werden alle Arbeiten, die in einem Projekt anfallen, in einem *Product Backlog* festgehalten. Ein *Product Backlog* ist eine Liste, die alle Features des zu entwickelnden Produkts enthält. Alle Funktionalitäten die der Kunde wünscht werden erfasst und nach Priorität geordnet. Das *Product Backlog* verändert sich durch das Verändern von Prioritäten, sowie durch wiederholtes Hinzufügen und Entfernen diverser Funktionalitäten.

Es soll (stark vereinfacht) die Verwaltung des *Product Backlogs* simuliert werden. Zuerst werden Features (Name und Beschreibung) mit Prioritäten (1 bis 100) eingefügt. Dann kann folgendes ablaufen:

- Ein Feature mit der höchsten Priortät (je höher desto dringender) wird implementiert und dann vom *Product Backlog* entfernt.
- Es können neue Features und Prioritäten eingefügt werden.
- Prioritäten von existierenden Features können sich ändern.

```
class BacklogImpl implements Backlog {
    public Backlog() {...}
    // members of interface Backlog
    ...
}
```

```
class Feature {
    private String name;
    private String description;
    public Feature(...) {...}
    public String toString() {...}
    public getName() {...}
    public getDescription() {...}
}
```

Verwalten Sie die Features des *Product Backlog* in der Klasse `BacklogImpl` als Heap. Verwenden Sie dabei das in der Vorgabedatei `Backlog.java` definierte interface `Backlog` als öffentliche Schnittstelle. Implementieren Sie den Heap mit einem Array (keinesfalls mit einem Baum). Sie können dafür z.B. die Klasse `java.util.ArrayList` verwenden. Hinweis: die Priorität der Features soll *\*nicht\** in der Klasse `Feature` gespeichert werden.

```

public interface Backlog {
    // add a feature to the backlog with the given priority
    void insert(Feature f, int priority);
    // get next feature and remove from backlog
    Feature next();
    // get next feature (and do not remove from backlog)
    Feature peek();
    // true - when backlog contains feature(s), false - when empty
    boolean hasNext();
    // get number of features in backlog
    int length();
    // find the feature with the give name and change its priority
    // return false, if feature is not found, otherwise true
    boolean changePriority(String name, int newPriority);
    // display content of backlog, sorted by priority (descending)
    void print();
}

```

Implementieren Sie einen Testtreiber. Der Testtreiber soll mit Hilfe der Vorgabedatei features.txt die `insert`-Funktion testen. Jede Zeile der Vorgabedatei enthält ein Feature im Format: `PRIORITY "NAME" "DESCRIPTION"` (siehe Beispiel unten). Definieren Sie weitere Testfälle für die anderen Methoden der Schnittstelle `Backlog`.

Beispiel:

```

51 "F2" "Beschreibung von F2"
23 "F3" "Beschreibung von F3"
34 "F1" "Beschreibung von F1"
11 "F9" "Beschreibung von F9"

```

#### Aufgabe 04.2: Einfügen und Löschen in Heaps

Gegeben ist folgendes Array:

0	1	2	3	4	5	6	7	8
	T	S	O	G	R	I	N	A

- Stellen Sie den durch das Array beschriebenen Heap als Binärbaum dar.
- Fügen Sie die gegebenen Schlüssel nacheinander in den Heap ein und zeichnen Sie den Heap nach jedem Schritt als Binärbaum. Kennzeichnen Sie, welche Vertauschoperationen im Heap beim Einfügen erfolgen.

#### EXAMPLE

- Stellen Sie den Heap aus b) wieder als Array dar und entfernen Sie den höchstpriorären Schlüssel. Wiederholen Sie das Entfernen 3x und zeichnen Sie nach jedem Schritt den Heap als Array. Kennzeichnen Sie welche Vertauschoperationen beim Entfernen erfolgen.