

Übung 04: Heaps

Abgabetermin: 08.05.2007 12:00

Name: _____

Matrikelnummer: _____

Gruppe: G1 (Dhungana) G2 (Wolfinger) G3 (Wolfinger)

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Pkte
Aufgabe 04.1	19	<input type="checkbox"/>	Java-Programm Testfälle und Ergebnisse	Java-Programm	<input type="checkbox"/>	
Aufgabe 04.2	5	<input type="checkbox"/>	Heaps in Array- Darstellung	-	<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 Priorität (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 {
    String name;
    String description;
    public Feature(...) {... }
    public String toString() {... }
}
```

Verwalten Sie die *Feature's* 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 addFeature(Feature f, int priority);
    // remove a feature
    void removeFeature(String name);
    // remove all features with a given priority
    void removeFeatures(int priority);
    // 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);
    // return next feature to be implemented and remove it from backlog
    Feature implementNextFeature(); // with the highest priority
    // display content of backlog, sorted by priority (descending)
    void print();
    // get number of features in backlog
    int getLength();
    // true - when backlog contains feature(s), false - when empty
    boolean hasNext();
}
```

Definieren Sie sinnvolle Testfälle und implementieren Sie einen Testtreiber. Der Testtreiber soll dazu die in der Vorgabedatei features.txt definierten Features einlesen. Jede Zeile der Vorgabedatei enthält ein Feature im Format: PRIORITY "NAME" "DESCRIPTION"

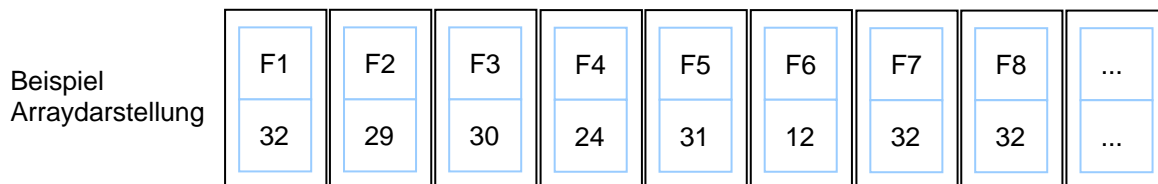
Beispiel: 51 "F2" "Beschreibung von F2"
 23 "F3" "Beschreibung von F3"
 34 "F1" "Beschreibung von F1"
 11 "F9" "Beschreibung von F9"

Aufgabe 04.2: Einfügen und Löschen in Heaps (zur Präsentation an der Tafel vorbereiten)

Simulieren Sie mit Zeichnungen, wie die folgende Liste von Features von einem „Online Pizza Konfigurator“ in einen Heap umgewandelt werden kann. Zeichnen Sie dafür die Baumdarstellung und die Arraydarstellung der Features nach jeder Einfügeoperation.

Name	Beschreibung	Priorität
F1	Bestellung Aufnahme	32
F2	Email Bestätigungen	29
F3	Vorschau für Bestellungen	30
F4	Beratung bei der Auswahl von Zutaten	24
F5	Benutzer Registrierung	31
F6	Bonuspunktesystem für Stammkunden	12
F7	Bestellung Storno	32
F8	Berechnung des Kaloriengehalts	13
F9	Empfehlung der Zutaten	10

In der Zeichnung geben Sie den Namen und die Priorität des Features an.



Ändert sich die Heapstruktur, wenn die Priorität von allen existierenden Features um 1 erhöht wird? Begründen Sie ihre Antwort.