

## Aufgabe 4 (20 Punkte) zum 20.6.2013

### Implementierung eines abstrakten Datentyps

Implementieren Sie die folgende Schnittstelle, die den abstrakten Datentyp Deque (Doule-Ended QUEue) repräsentiert, zwei Mal:

```
package de.uni_potsdam.hpi;
public interface Deque{
    int capacity();
    int size();
    void clear();
    void addFirst(Object e) throws DequeFull;
    void addLast(Object e) throws DequeFull;
    Object removeFirst() throws DequeEmpty;
    Object removeLast() throws DequeEmpty;
}
```

Eine Klasse, die die Schnittstelle Deque implementiert, soll dabei folgende Eigenschaften besitzen:

- Der Konstruktor der Klasse erwartet die maximale Kapazität des Deque-Exemplars. Die Methode `capacity()` gibt diese Größe zurück.
- Mit der Methode `addLast(Object)` wird ein Element in die Warteschlange am Ende eingefügt. Falls die Kapazität bereits erreicht ist, löst `.addLast` die Ausnahme `de.uni_potsdam.hpi.DequeFull` aus. Gleichermaßen fügt `addFirst` am Anfang ein.
- Die Methode `removeFirst()` liefert das erste Elemente der Warteschlange und entfernt es aus dieser. Sollte die Warteschlange leer sein, so löst `.removeFirst` die Ausnahme `de.uni_potsdam.hpi.DequeEmpty` aus. Gleichermaßen löscht `removeLast` ein Element am Ende.
- Die Methode `size()` liefert die aktuelle Zahl von Elementen in der Warteschlange.
- Die Methode `clear()` löscht alle Elemente aus der Warteschlange.

Achten Sie bei Ihrer Implementierung darauf, dass alle Methoden die Komplexität  $O(1)$  besitzen (vorausgesetzt, dass alle Elementaroperationen, einschließlich der Speicherallozierung, in konstanter Zeit erfolgen).

#### Teilaufgaben:

- a. Implementieren Sie Deque zum einen als doppelt verkettete Liste. Zur Erreichung konstanter Zeit für alle Operationen empfiehlt es sich, eine Referenz nicht nur auf den Anfang der Liste, sondern auch auf das

Ende der Liste zu führen. Alternativ ist es auch erlaubt, erstes und letztes Element der List miteinander zu verbinden, ggf. unter Einsatz eines Wächterelements.

- b. Implementieren Sie Deque zum anderen als Ringpuffer auf Basis eines Arrays. Für einen solchen Ringpuffer sind neben dem Array selbst zwei Integer-Variablen (etwa first und last) erforderlich, die den ersten und den letzten belegten Index anzeigen.
- c. Vergleichen Sie die Laufzeit beider Implementierungen für folgendes Anwendungsszenario: In eine Warteschlange der Kapazität 1000 werden 500 Elemente eingefügt, danach wird oft abwechselnd ein Element entfernt und der Leerstring ("") eingefügt; vergleichen Sie insbesondere die Zeit für dieses Einfügen und Entfernen.