



Aufgabe 10.1 (Ü) Queues

Eine Schlange ist eine Datenstruktur zur Speicherung von Elementen nach dem FIFO-Prinzip (First-In-First-Out). Die zweite der in der Vorlesung vorgestellten Implementierungen von Schlangen (Klasse: `Queue`) verwendet zur Speicherung der Elemente ein Array. Zwei Indizes, `first` und `last`, geben dabei die Position des ersten bzw. letzten Wertes an. Bei Bedarf wird das Array beim Einfügen von Elementen durch `enqueue()` durch ein größeres Array ersetzt. Eine Verkleinerung erfolgt nicht.

Erweitern Sie die in der Vorlesung vorgestellte Methode `dequeue()` (zum Entnehmen des nächsten Elements), so dass das Array bei Bedarf auch verkleinert wird.

Hinweis: Die zu ergänzende Klasse `Queue` finden Sie auf der Internetseite zur Übung.

Aufgabe 10.2 (H) Queues II

Benutzen Sie nun Ihre Klasse `LinkedList` aus Blatt 9 (siehe Lösungsblatt und File auf der Übungshomepage), um String-Warteschlangen mit doppelt verketteten Listen zu realisieren. Die Einträge der Warteschlange sollen diesmal vom Typ `String` sein.

Ein Objekt der Klasse `QueueDLL` enthält nur noch einen Verweis, `header`, auf ein Objekt der Klasse `LinkedList`. Die Klasse `LinkedList` enthält bereits die Funktionalitäten aus der Hausaufgabe 9.2. Implementieren Sie die Klasse `QueueDLL`:

- a) Erstellen Sie einen Konstruktor, der eine leere `QueueDLL` erzeugt.
- b) Implementieren die Methoden:
 - `boolean isEmpty()` testet auf Leerheit
 - `String dequeue()` liefert das erste Element und entfernt dieses
 - `void enqueue(String x)` reiht x in die Warteschlange ein (am Ende)
 - `String toString()` liefert eine String-Darstellung der ganzen Schlange

Benutzen Sie dabei gegebenenfalls die Methoden der Klasse `LinkedList`.