

Aufgabenblatt 05: Generische Queue als Ringpuffer

Sie werden im Verlauf dieses Studiums noch viele Möglichkeiten haben, generische Klassen zu schreiben, vielleicht sogar im Rahmen dieses Praktikums. Aber diese eine Aufgabe passt jetzt ganz gut zum Stoff dieser und der letzten Woche.

Eine *Queue* ist eine Struktur, bei der am Ende eingefügt wird und am Anfang entnommen wird. Es handelt sich also um einen Lifo-Speicher (First-in, First out), wie an einer Supermarktkasse etc. Die minimal zu implementierenden Operationen sind:

1. *enqueue*: Einfügen eines Elements am Ende.
2. *dequeue*: Entfernen des Elements am Anfang.
3. *peek*: Liefert das Element am Anfang zurück.
4. *isEmpty*: Liefert *true*, wenn die Queue leer ist, andernfalls *false*.
5. *isFull*: Liefert *true*, wenn die Queue voll ist, andernfalls *false*.

Eine gängige Variante zur Implementierung einer *Queue* ist ein *Ringpuffer*: Dies ist ein Array, in dem zunächst mittels *enqueue* an den Stellen 0, 1, ... eingefügt und mittels *dequeue* entsprechend bei 0, 1, ... entnommen wird. Ist mittels *enqueue* das Ende des Arrays erreicht, so wird dann wieder bei 0 beginnend eingefügt, so dort (inzwischen) Platz ist.

1. Entwerfen Sie bitte diese Struktur und die Methoden zum Einfügen! Entscheiden Sie sich dabei für eine systematische Fehlerbehandlung (Exceptions oder Rückgabewerte)!
2. Schreiben Sie bitte ein Interface, das diese Entscheidungen widerspiegelt!
3. Schreiben Sie bitte eine Klasse, die Ihr Interface implementiert!
4. Demonstrieren Sie bitte an sinnvollen JUnit-Testfällen mit primitiven Typen und mit Klassen, dass Ihre Implementierung funktioniert!

Abgabe: **Mittwoch, 01.11.2010, 12:30**

per Email an bernd.kahlbrandt@informatik.haw-hamburg.de

Viel Spaß mit Generics!