Alternative Implementation einer Schlange als Array

Da es sich bei einer Schlange um einen asbtrakten Datentyp handelt, spielt es für die Funktionalität keine Rolle, wie die Elemente im Hintergrund verwaltet werden. Die Elemente der Queue können auch in einem Array gespeichert werden. Dabei merkt man sich mit Hilfe zweier Attribute `first` und `last`, welcher Bereich des Arrays die Queue repräsentiert.

- Das Attribut `first` ist dabei sinnvollerweise inklusive gemeint, `last` sollte eine exklusive Grenze sein (Beispiel 1).
- Beim Einreihen eines neuen Elements wird dieses am Index `last` ins Array geschrieben und `last` wird um 1 erhöht (Beispiel 2).
- Wenn ein Index so erhöht wird, dass er die Länge des Arrays erreicht, wird er auf 0 zurückgesetzt (Beispiel 3).
- Um das vorderste Element der Queue zu entfernen, nimmt man den Wert am Index `first` und erhöht `first` um 1 (Beispiel 4). Es stört nicht, dass der alte Wert weiterhin im Array steht – wir greifen nie wieder lesend auf ihn zu. Irgendwann wird der Wert wieder überschrieben.
- Wenn nach dem Einfügen eines neuen Wertes `first` und `last` gleich sind, ist das Array *„voll“* (Beispiel 5). Damit später weiterhin Elemente eingefügt werden können, muss das Array vergrößert werden. Man erzeugt ein neues Array, das z.B. doppelt so groß ist und kopiert die alten Werte ins neue Array. Dann passt man die Indizes `first` und `last` passend an.

Implementieren die Klasse `ArrayQueue` nach diesem Prinzip und führe die Tests der zugehörigen Testklasse aus.

Hinweis: In Java kann man kein Array eines generischen Typs erzeugen. Stattdessen erzeugt man ein Object-Array und führt beim Auslesen einen Cast zum generischen Typ T aus:

```
Object[] daten = new Object[10];
...
T front = (T)daten[5];
```

From:
https://wiki.qg-moessingen.de/ - **QG Wiki**

Permanent link:
https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:adt:queue:array_queue:start?rev=1636614708

Last update: **11.11.2021 08:11**