Vorgehensweise und Entscheidungsfindung

Interface-Entwurf und Fehlerbehandlung:

Zunächst haben wir das Interface IringBuffer erzeugt. Neben den fünf geforderten Operationen haben wir als zusätzliche Operation size eingebunden, damit die aktuelle Größe des RingBuffer abgefragt werden kann. Fehler können bei den Operationen enqueue, dequeue oder peek auftreten. Wird versucht ein weiteres Element in einen bereits vollen RingBuffer zu werfen, führt dies zu einer RunTimeException (Overflow). Bei dequeue und peek fliegt die RunTimeException (underflow), falls versucht wird, aus einer leeren Liste etwas herauszuholen.

Klassenentwurf:

In der Klasse gibt es die folgende Attribute:

- private T[] elements → Zum Speichern der Elemente
- private int size → speichert die Anzahl der Elemente
- private int head → Zeiger auf das erste Element
- private int last → Zeiger auf die nächste freie Speicherzelle

Der Konstruktor bekommt als Parameter die Länge des RingBuffers übergeben. Danach wird auf den Typ gecastet, mit dem initialisiert wurde und ein entsprechendes Array mit der übergebenen Länge erstellt. Die Methoden isEmpty, isFull und peek sind trivial. Interessant sind die Methoden enqueue und dequeue, die allerdings recht analog zueinander sind (außer size-- und size++).

Zu Beginn (leerer Buffer) zeigen die beiden Zeiger head und last auf das nullte Feld. Das erste Element wird an der Stelle last eingefügt. Danach wird der Zeiger last aktualisiert, er springt um einen weiter (+1) auf das Feld 1. Die Modulo Anwendung auf die Arraylänge ergibt in diesem Falle ebenfalls 1. Interessant wird es erst, wenn z.B. das vorletzte Feld im Array belegt ist und z.B. alle weiteren frei. Dann zeigt der head auf das vorletzte Feld und last auf das letzte Feld. Wenn man jetzt ein Element z.B. reinwirft, ergibt die Erhöhung von last um 1 modulo auf die Arraylänge 0. Damit findet ein „wrap arround“ statt und last zeigt auf das erste Feld im Array.

Testfälle:

Bei den Testfällen prüfen wir nur mit Objekttypen. Reinwerfen können wir später, trotz Initialisierung mit der z.B. Integer oder Character, auch primitive Typen (int, char), die durch autoboxing und autounboxing eh gewandelt werden.

Martin Slowikowski
Matrikelnummer: 199 91 66

Jan-Tristan Rudat
Matrikelnummer: 200 78 52

Teamname: Bernie und Ert