

# Übungsblatt 7

Abgabe: nein

---

## Aufgabe 1 Ein Ring, sie zu testen...

In Aufgabe 2 sollt ihr das Verfolgen eurer Spielfigur durch einen NPC implementieren. Eine geeignete Datenstruktur, um die dazu notwendige Zwischenspeicherung von Schritten zu realisieren, ist ein *Ringpuffer*. Da die Funktionsweise eines Ringpuffers nicht ganz einfach ist, soll er vorab getrennt vom Spiel getestet werden. Öffnet dazu das Projekt *RingBuffer* mit BlueJ. Die Klasse *RingBuffer* enthält bereits eine vermutlich korrekte Implementierung eines solchen Puffers.

### Aufgabe 1.1 Testfälle definieren

Überlegt euch eine Reihe von Tests, mit der die Implementierung des Ringpuffers getestet werden soll, damit ihr sicherstellt, dass sie sich auch so verhält, wie es die Kommentare im Quelltext beschreiben.<sup>1</sup> Erzeugt mit BlueJ eine Testklasse für *RingBuffer* und zeichnet die Tests auf oder implementiert sie von Hand.

### Aufgabe 1.2 Teststärke prüfen

Da die Implementierung vermutlich richtig ist, sollten alle Tests erfolgreich durchlaufen. Aber hätten eure Tests eigentlich Fehler gefunden? Um dies zu prüfen, nehmt mindestens 10 sinnvoll gewählte Änderungen an der Implementierung vor (immer nur eine zur Zeit), lasst die Tests jeweils durchlaufen und haltet fest, welche Tests nun fehlschlagen. Wenn eine Änderung unbemerkt bleibt, solltet ihr Aufgabe 1.1 noch einmal überarbeiten oder euch überlegen, warum das richtig sein kann. Änderungen könnten u.a. sein:

- Bedingungen auf *true* oder *false* setzen.
- Operatoren durch ähnliche ersetzen, z.B.  $>$  durch  $>=$ .
- Teile von Ausdrücken oder ganze Anweisungen weglassen.

## Aufgabe 2 Spürhund

Benutzt den Ringpuffer aus Aufgabe 1, um das Verfolgen eurer Spielfigur durch einen NPC zu realisieren. Erweitert die Implementierung aus Übungsblatt 3 so (oder erstellt eine neue Klasse), dass der NPC in einen Verfolgermodus umschaltet, wenn die Spielfigur in einer bestimmten Entfernung vor ihm platziert wird. Ein Ringpuffer wird genutzt, um die Bewegungen der Spielfigur aufzuzeichnen. Da diese nur Einerschritte macht und dabei in Richtung ihrer Schritte blickt, reicht es, ihre Rotationen zu speichern. Tragt erst die Rotation des NPC so oft in den Ringpuffer ein, wie es nötig ist, um durch Schritte die jetzige Position der Spielfigur zu erreichen. Tragt danach mit jedem weiteren Schritt der Spielfigur deren Rotation nach dem Schritt in den Ringpuffer ein. Zusätzlich entnehmt ihr immer eine Rotation aus dem Ringpuffer und bewegt euren NPC in diese Richtung. Dadurch führt der NPC zeitverzögert dieselben Bewegungen wie die Spielfigur aus.

---

<sup>1</sup>Verhalten, dass als *undefiniert* angegeben ist, kann nicht getestet werden.