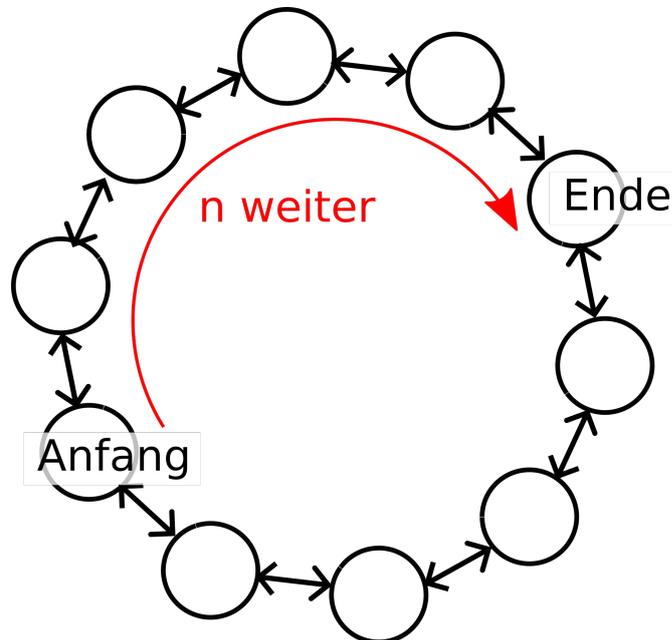


## Einführung in die Programmierung für Studierende der Naturwissenschaften

Blatt 8 – 27.06.2022

Abgabe: bis Montag, 04. Juli, 16 Uhr per e-Mail an Ihre:n Tutor:in.

**Aufgabe 1** (5 Punkte). Das “auszählen” soll in einem Computerprogramm realisiert werden. Dabei stehen alle in einem Kreis. Beginnend bei einer Zufälligen Position, “Anfang”, wird im Uhrzeigersinn mit einem Auszählsspruch  $n$  Personen weiter gezählt und die Person am “Ende”, die “getroffen” wird, ist raus und muss den Kreis verlassen.



Welche Vorteile hat es, (i) beim Finden der ausgezählten Person und (ii) beim Entfernen der ausgezählten Person, die Liste der Personen in einem `std::vector` oder einer `std::list` zu speichern? Füllen Sie die folgende Tabelle aus!

		Datenstruktur	
		<code>std::vector</code>	<code>std::list</code>
Aktion	Finden der Person	Vorteile/Nachteile	Vorteile/Nachteile
	Entfernen der Person	Vorteile/Nachteile	Vorteile/Nachteile

*Hinweis:* Sie müssen hier nichts implementieren sondern sollen lediglich begründen, welche Datenstruktur beim Implementieren welche Vor- und Nachteile hätte.

**Aufgabe 2** (2 + 3 Punkte). Auf der Vorlesungswebseite finden Sie die Datei `dynamic_vector.cc`, in dem die Klasse `DynamicVector` implementiert wurde. Laden Sie die Datei herunter und fügen Sie folgende Methoden hinzu

- (a) Eine Methode `unsigned long size() const`, um die aktuelle logische Größe des `DynamicVector` abzurufen
- (b) Eine Methode `void pop_back()`, um das letzte Element aus dem `DynamicVector` logisch zu entfernen

**Aufgabe 3** ((2+4+4) Punkte). Auf der Vorlesungswebseite finden Sie eine Datei `zahlen.txt`, in der eine Folge von Zahlen gespeichert ist. Diese sollen in einer `std::list<int>` gespeichert werden.

- (a) Lesen Sie die Datei zeilenweise ein. Speichern Sie jede neue Zahl am Anfang der Liste, falls sie negativ ist und am Ende der Liste, falls sie nicht negativ ist.
- (b) Lesen Sie die Datei zeilenweise ein. Die Zahlen sollen jetzt *sortiert* in der Liste gespeichert werden. Speichern Sie sich dazu nach jedem Einfügen die Position, an der Sie die letzte Zahl eingefügt haben und suchen Sie im nächsten Schritt die Position der nächsten Zahl in der Nähe der letzten Position.
- (c) Löschen Sie alle Duplikate aus der sortierten Liste, indem Sie an jeder Position `it` das nächste Element löschen, falls dieses den gleichen Wert hat wie das an der Position `it`.

*Hinweis:* die Teilaufgaben (a) und (b) bauen nicht aufeinander auf. Wenn Sie (c) lösen wollen ohne (b) implementiert zu haben, generieren Sie selber ein passendes Minimalbeispiel.

**Aufgabe 4** (optional). Schildern Sie in einer Datei `erfahrung.txt` kurz Ihre Erfahrung mit dem aktuellen Übungsblatt. Berichten Sie darin wieder in Stichpunkten bzw. ein bis zwei kurzen Sätzen über Ihre Erfahrungen mit Kursinhalt und Übungsaufgaben. Was fiel Ihnen leicht? Was ist noch unklar? Wie viel Zeit haben Sie für die Bearbeitung der Hausaufgaben benötigt und welche Probleme traten dabei auf?