Schlangen

Vor Kassen, an Schaltern, auf der Autobahn und an zahlreichen anderen Orten bilden sich häufig **Schlangen**.

Die Anzahl der Elemente (Personen, Autos) kann sich dabei ändern, jedoch sind gewisse Regeln einzuhalten: Neue Elemente werden stets **hinten** an der Schlange eingefügt, Elemente die die Schlange verlassen (oder von dieser entfernt werden sollen) werden stets **vorne** entfernt.

Dieses Prinzip nennt man auch "**FiFo**": First in, First Out.

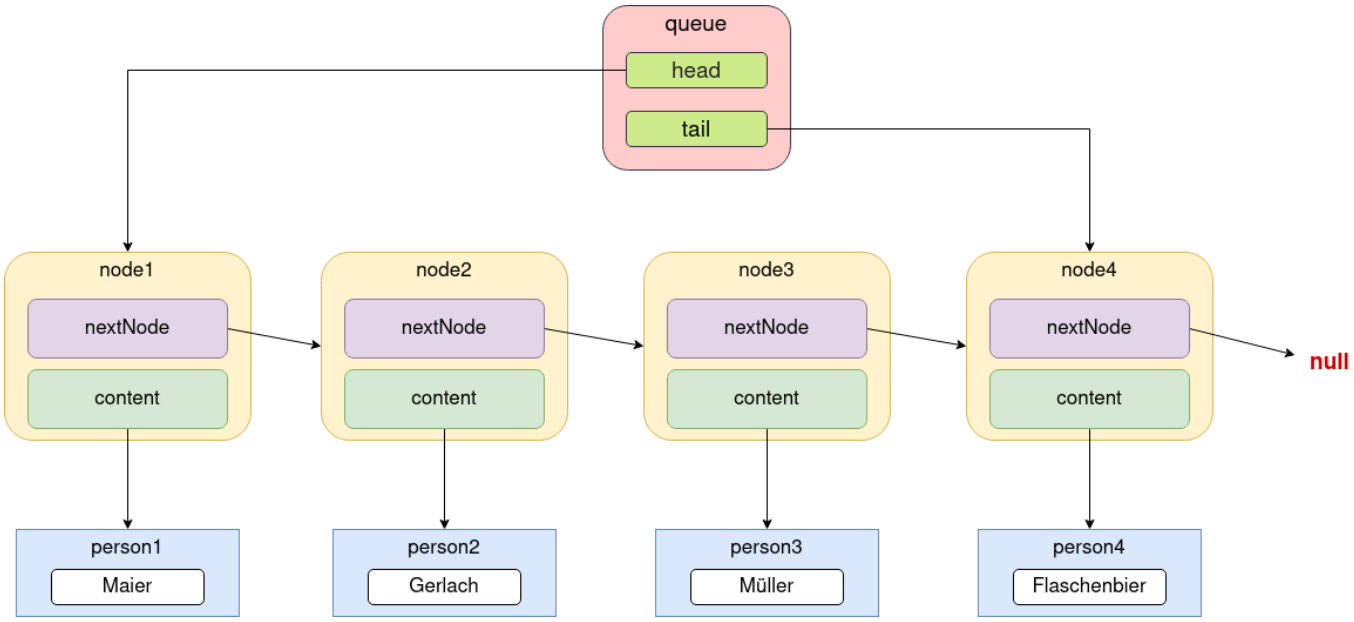
Schlangen finden in der Informatik häufig Anwendung, z.B. zur Verwaltung von Arbeitsaufträgen. Ein bekanntes Beispiel dafür ist z.B. eine Druckerwarteschlange: neue Druckaufträge werden stets hinten eingereiht, die zuerst eingereihten Aufträge werden zuerst abgearbeitet und dann aus der Schlange entfernt. [1)]

Struktur einer generischen Schlange

Objekte vom Typ `Node` (deutsch: Knoten) speichern jeweils einen Verweis auf den nächsten Knoten (`nextNode`) sowie einen Verweis auf ein beliebiges Inhaltsobjekt (`content`), z.B. vom Typ Person. Damit übernehmen die Knoten die Verwaltung der Struktur der Schlange, die eigentlichen Inhalte sind jedoch in die Inhaltsobjekte ausgelagert.

Jede Klasse hat also ihren eigenen Verantwortungsbereich: Die Klasse `Person` verwaltet alle Daten zur Person, die Klasse `Node` stellt Attribute und Methoden zur Verfügung, die für die Organisation der FIFO-Daten-Struktur nötig sind.

Da die Knoten lediglich Verweise auf Inhaltsobjekte verwalten, können Objekte eines beliebigen Typs in der Schlangenstruktur verwaltet werden. Man spricht von einer **generischen Schlange**.



Die Klasse `queue` stellt ein Schnittstelle bereit, um mit der Knotenstruktur arbeiten zu können.

Damit das sinnvoll möglich ist, muss die `queue`-Klasse Methoden zum Einfügen, Löschen und Auslesen von Elementen anbieten.

`queue` speichert sich eine Referenz auf den ersten Knoten der Schlange (`head`) als Attribut, damit man auf das erste Schlangenelement zugreifen kann, außerdem und auch eine Referenz auf den letzten Knoten (`tail`), damit man am Ende der Schlange neue Elemente einfügen kann.

Zur Funktionalität der Datenstruktur **Schlange** (**Queue**) gehören, neben den Methoden Einfügen (`enqueue`) und Entfernen (`dequeue`) die Ausgabe des ersten Elements (`front`) sowie die Abfrage, ob die Warteschlange leer ist (`isEmpty`).

Um diese Methoden umsetzen zu können, muss die Knotenklasse der Schlange `Node` folgende Methoden bereitstellen:

- Der Nachfolgeknoten muss gesetzt und abgefragt werden können (`setNext` bzw. `getNext`)
- das jeweilige Inhaltsobjekt muss abgefragt werden können (`getContent`)

Du findest hier eine BlueJ-Vorlage[2], in der du eine Schlange implementieren kannst.

Die folgenden Erklärungen zu den einzelnen Funktionalitäten der Schlange sollten dir helfen.

- Einfügen eines Elements am Ende (enqueue)
- Entfernen eines Elements (dequeue)

—-

Alternative Implementation

- Alternative Implementation einer Schlange als Array

1)

Photo by John Cameron on Unsplash

2)

git clone https://codeberg.org/qg-info-unterricht/bluej-linked-queue.git