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.

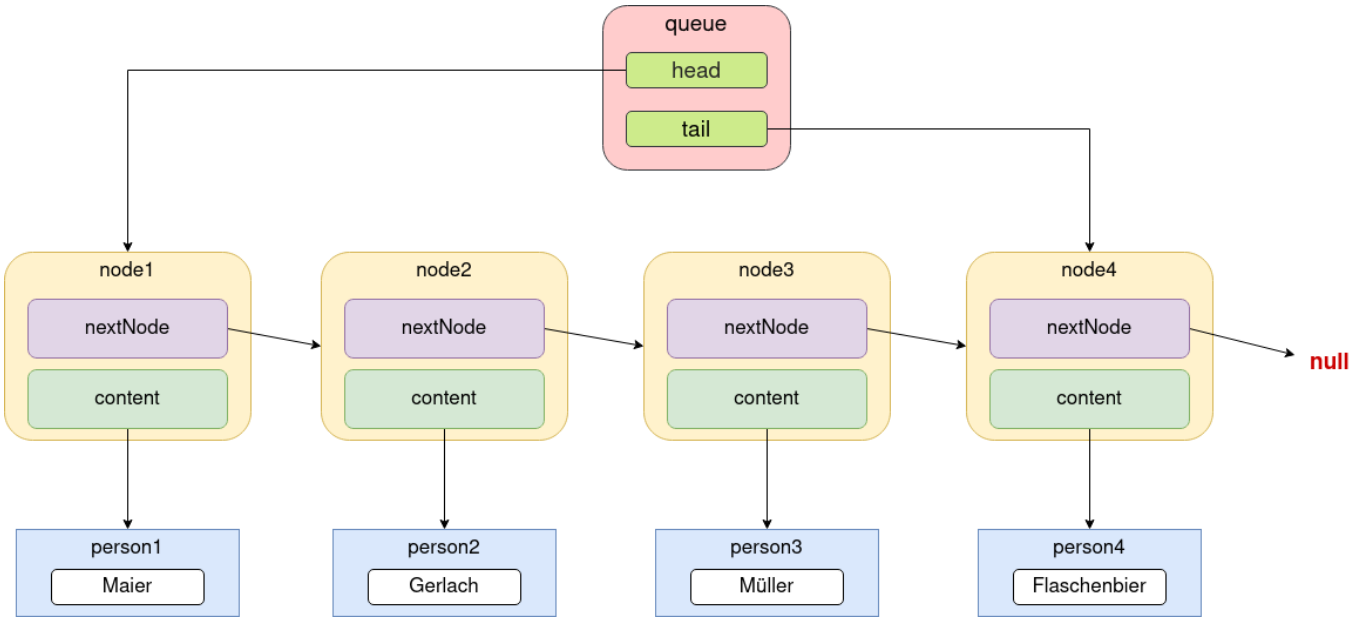
Schlangen finden in der Informatik häufig Anwendung, z.B. zur Verwaltung von Arbeitsaufträgen. Ein bekanntes Beispiel dafür ist z.B. eine Druckerwarteschlange: neue Druckaufträge werden stets hinten eingereiht, die zuerst eingereihten Aufträge werden zuerst abgearbeitet und dann aus der Schlange entfernt.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

Die Knoten der Datenstruktur werden als innere Klasse deklariert, da einzig die Klasse Queue Zugriff auf die Knoten haben soll und benötigt. Queue und QueueNode bilden so in ihrem Zusammenhalt eine logische Einheit mit klar abgegrenzten Aufgaben gegenüber anderen Klassen.

1)

Photo by John Cameron on Unsplash

From:
https://www.info-bw.de/ -

Permanent link:
https://www.info-bw.de/faecher:informatik:oberstufe:adt:queue:start?rev=1633972997

Last update: **11.10.2021 17:23**