**Projekt Wartezimmer (Klasse Queue)**

**Beispiel: (FIFO-Prinzip)**

Das FIFO-Prinzip ist ein grundlegendes Prinzip der Organisation von Produkten oder Abläufen. FIFO ist die Abkürzung für „First In, First Out“.

Warenwirtschaft

Ältere (zuerst gelagerte) Bestände werden nach Möglichkeit zuerst verbraucht. Dieses hat insbesondere eine Bedeutung, wenn die Mindesthaltbarkeit eine Rolle spielt. Bei der Lagerung von Schüttgütern wird dieses durch den Einsatz von Silos organisiert.

Informatik

Datenströme (Pipe oder Pipeline genannt) zwischen zwei Prozessen, die durch einen Puffer organisiert werden, arbeiten nach dem FIFO-Prinzip. Beispiele sind Tastaturpuffer, Druckerwarteschlagen, Serielle Schnittstelle (RS 232-Port).

Alltag

Warten an Einkaufskassen, Tankstellen, Aufrufsysteme, Arztpraxen.

Mathematik

Die Warteschlangentheorie (oder Bedienungstheorie) ist ein Teilgebiet der Wahrscheinlichkeitstheorie bzw. der Unternehmensforschung und somit ein Beispiel für angewandte Mathematik. Sie beschäftigt sich mit der mathematischen Analyse von Systemen, in denen Aufträge von Bedienungsstationen bearbeitet werden, und gibt Antwort auf die Fragen nach den charakteristischen Größen wie der Stabilität des Wartesystems, der Anzahl der Kunden im System, ihrer Wartezeit usw. Sie unterstützt unter anderem Managemententscheidungen über den Personaleinsatz und den Abfertigungsprozess und hilft, ein System zur Leistungsmessung auszubauen. Ihre Anwendung reicht von Computern, Telekommunikationssystemen, Verkehrssystemen über Logistik bis zu Fertigungssystemen. (Wikipedia, <https://de.wikipedia.org/wiki/Warteschlangentheorie>, 17.07.2015)

**Modell einer Schlange (Queue)**

Als Modell einer Queue können Sie sich ein Rohr vorstellen, das an beiden Enden offen ist. An einem Ende werden neue Elemente hineingeschoben. Das Prinzip der Abarbeitung ist bei allen Beispielen das gleiche. Es unterscheidet sich nur dadurch, dass unterschiedliche Objekte derselben Klasse verarbeitet werden. Im Folgenden wird eine Klasse entworfen, die nach dem oben beschriebenen Abarbeitungsprinzip arbeitet.

Ein generischer Typ erlaubt es, Datentypen zu erzeugen, die von den zugrunde liegenden Typen abstrahieren.

**Klasse Queue**

**150717_Queue**

**Die generische Klasse Queue<ContentType>**

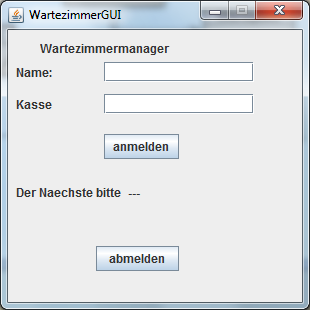
Objekte der generischen Klasse **Queue** (Warteschlange) verwalten beliebige Objekte vom Typ **ContentType** nach dem First-In-First-Out-Prinzip, d.h., das zuerst abgelegte Objekt wird als erstes wieder entnommen. Alle Methoden haben eine konstante Laufzeit, unabhängig von der Anzahl der verwalteten Objekte.

**Dokumentation der generischen Klasse Queue<ContentType>**

|  |  |
| --- | --- |
| **Konstruktor** | **Queue<ContentType>()** |
|  | Eine leere Schlange wird erzeugt. Objekte, die in dieser Schlange verwaltet werden, müssen vom Typ **ContentType** sein. |
| **Anfrage** | **boolean isEmpty()** |
|  | Die Anfrage liefert den Wert **true**, wenn die Schlange keine Objekte  enthält, sonst liefert sie den Wert **false**. |
| **Auftrag** | **void enqueue(ContentType pContent)** |
|  | Das Objekt **pContent** wird an die Schlange angehängt. Falls **pContent** gleich null ist, bleibt die Schlange unverändert. |
| **Auftrag** | **void dequeue()** |
|  | Das erste Objekt wird aus der Schlange entfernt. Falls die Schlange leer ist, wird sie nicht verändert. |
| **Anfrage** | **ContentType front()** |
|  | Die Anfrage liefert das erste Objekt der Schlange. Die Schlange bleibt  unverändert. Falls die Schlange leer ist, wird **null** zurückgegeben. |

**Aufgabe (s. Q1-II.1)**

Im Laufe eines Tages kommen zu zufälligen Zeiten Patienten zur Behandlung in eine Arztpraxis. Sobald ein Patient eintrifft werden sein Name und seine Krankenkasse erfasst. Die Patienten werden in der Reihenfolge ihres Eintreffens in das Wartezimmer gesetzt und in dieser Reihenfolge behandelt.

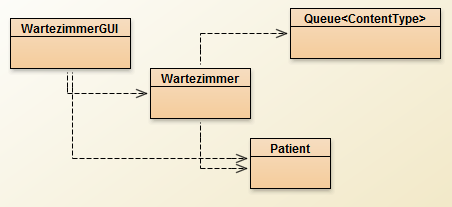


Entwickeln Sie ein Simulationsprogramm für diese Patientenverwaltung. Nutzen Sie dazu die Klasse **Queue**. Zur Vereinfachung werden nur der Patientenname und die Bezeichnung der Krankenkasse des Patienten verwaltet.

Lösungsskizze:

150728_Wartezimmer_Queue

Darstellung in BlueJ



Implementation mit Hilfe der Klasse **Queue**

