



Gemeinhin ist eine Liste eine Aufzählung von Dingen, seien es Zahlen (Altersangaben einer Klasse), Worte (Einkaufszettel) oder eine Liste von Personen.

- Eier
- Mehl
- Salz
- Karotten

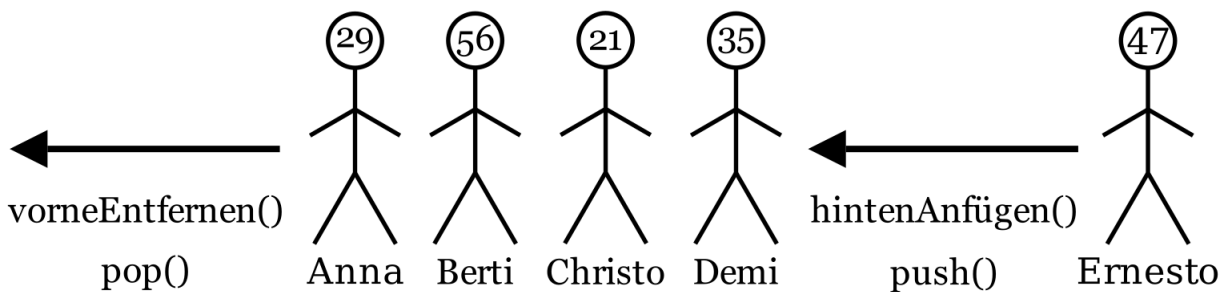
Listen sind so alltäglich, dass wir überlicherweise gar nicht aktiv über sie nachdenken.

Unser erstes Ziel in diesem Schuljahr ist die Übertragung des Konzepts der Liste in die Informatik. Hier benötigen wir noch viel öfter „**Zusammenstellungen**“ von Dingen. Das Zusammenfassen von Objekten zu Listen ist sogar so zentral, dass es Programmiersprachen gibt, die auf die Liste als Basiselement zurückgreifen, z.B. Lisp.

Traditionell ist die Liste ein umso wichtigeres Element der Sprache, je funktionaler diese angelegt ist. In objektorientierten Sprachen sind Listen in ihrer Funktionalität zwar ebenfalls vorhanden, allerdings nicht immer ohne einen gewissen Aufwand zu verwenden.

In den folgenden Kapitel werden wir selbst Implementierungen für Listen in Java entwickeln. Wie oft in der Programmierung gibt es dabei **nicht** einen Königsweg, sondern verschiedene Ansätze, die unterschiedliche Vor- und Nachteile bringen.

Als Prototyp einer Liste wird uns dabei eine **Warteschlange** dienen (ganz bildlich gedacht), z.B. die Schlange vor einer Kasse.



Hinweis: Um sprachlich stringenter zu bleiben und Denglisch möglichst zu vermeiden, werden die Methoden im Fließtext häufig auf Deutsch stehen bzw. beschrieben werden, mit der üblichen englischen Übersetzung bei der ersten Verwendung in Klammern. Auch die Abbildungen werden, wenn nötig, beide Schreibweisen enthalten, der Quellcode aktuell nur englische Bezeichnungen (Eine zweite Version des Skripts mit deutschem Quellcode ist geplant, wenn nötig). Für Prüfungen sind sowohl deutscher als auch englischer Quellcode in Ordnung. Begründungen müssen aber in jedem Fall auf Deutsch geschrieben werden. Im Abitur ist auch der Quelltext i.d.R. auf Deutsch.

Zurück zum obigen Bild. Wir werden Anna, Berti, Christo, Demi und Ernesto noch öfter sehen, da sie unsere Liste füllen und auch wieder verlassen werden. Sie stehen in einer **Warteschlange (Queue)**. Auch eine Warteschlange ist nur eine Form von Liste, allerdings mit speziellen Eigenschaften. Halten sich alle an die Regeln, so ist es nur möglich, dass sich jemand hinten anstellt oder vorne aus der Warteschlange entfernt wird, weil er dran ist und „abgearbeitet“ wird. In unserer zukünftigen Modellierung entspricht dies den Methoden `vorneEntfernen()` (`pop()`) bzw. `hintenAnfügen()` (`push()`).

Hinweis: Übergabeparameter werden im Text häufig weggelassen, natürlich müsste die zweite Methode etwa so aussehen: `hintenAnfügen(Mensch mensch)`. Die Warteschlange gehorcht dem sogenannten **FIFO-Prinzip** (First In, First Out). Neben diesem gibt es noch weitere Ansätze, siehe dazu auch Kapitel 6, das sich mit Typen von Listen beschäftigt.

Es bleibt abschließend die Frage, wie sich die Idee einer Warteschlange objektorientiert umsetzen lässt.