

Der Datentyp Map¹

Schaut man bei Wikipedia nach, findet man die Bezeichnung *assoziatives Array*. Auch an dieser Stelle zeigt sich die traditionelle Wurzel *Arrays* der Sammlungstypen. Eine Map ist also eine echte Sammlungsstruktur bei Java² und bietet also viel mehr als nur das, was ein Array bietet. Trotzdem ist es sinnvoll, ihre Grundkonstruktion am Unterschied zum normalen Array zu beschreiben.

Map

Maps sind so organisiert, dass man die Objekte als Paar ablegt. Die Elemente sind also Schlüssel – Wert Paare [key-value] und man kann direkt auf einen Wert über die Angabe des Schlüssels zugreifen.

Bei einem Array hat der Programmierer allein die Möglichkeit, die Größe des Arrays zu steuern. Nennen wir die Größe des Arrays einmal `anzahl`, dann erfolgt der Zugriff über die vom System so vorgegebenen Indizes `0`, `1`, ..., `anzahl`. Sie stellen also die einzigen "Schlüssel" zum Zugriff dar. Bei einer Map definiert der Programmierer oder der Anwender selbst die Schlüsselwerte, über die auf die zugeordneten Werte zugegriffen wird. Wie in einem Wörterbuch kann man dann beim `key` nachschlagen und erhält dazu den zugehörigen `value`.

Im mathematischen Sinn stellt diese Zuordnung eine Funktion dar, weil sie eindeutig sein muss. Das bedeutet also, dass man zu einem Schlüsselwert genau einen zugeordneten Wert haben kann.

HashMap

Eine interessante Variante der Datenspeicherung nutzt intern eine solche Paarbildung. Dabei wird zu jedem Objekt ein Hashcode berechnet, der die Position des Objektes in der Datenstruktur beschreibt. Der Zugriff auf so abgespeicherte Daten ist sehr schnell, so dass man sie zu nutzen versucht, wenn es auf solche Schnelligkeit ankommt. Hier ist allerdings das Problem der Kollision zu lösen. Die Werte können sich wiederholen und man muss sich Gedanken machen, wie man dann vorgeht.

Dieses Nachdenken nimmt einem die von Java bereit gestellte Klasse `HashMap` ab, die eine solche `Hashtable` intern zur effizienten Speicherung der key-value-Paare nutzt.

TreeMap

Eine weitere sehr effiziente Möglichkeit zur Speicherung bieten Bäume an. Wenn man eine eindeutige Ordnung der key-Daten³ hat, kann man die Speicherung in einem binären Baum organisieren. Der Baum hat eine Wurzel, also ein Element, auf das als erstes zugegriffen werden kann und einen Vorgängerzweig, der alle Elemente enthält, die vor ihm kommen, sowie einen Nachfolgerzweig mit allen Elementen, die nach ihm kommen. Die Theorie der Bäume ist sehr umfangreich. So ist z.B. ein wichtiges Problem, wie man beim Aufbau und der Verwaltung einer solchen Datenstruktur vermeidet, dass der Baum unausgewogen ist, also sehr viel mehr Elemente auf der einen Seite hat als auf der anderen.

1 Der Datentyp Map wird bei anderen Programmiersprachen auch Dictionary bezeichnet, in Scheme spricht man selbstverständlich nicht von einem assoziativen Array, sondern von einer Assoziationsliste.

2 Beachten Sie aber, dass Map eine interface ist. Dazu gibt es mit der `AbstractMap` eine Implementierung der wesentlichen Grundfunktionen, die man durch Erweitern zu einem eigenen Map-Typ verwenden kann oder deren von ihr ererbende Klassen, die hier teilweise beschrieben sind.

3 Sie müssen das interface `Comparable` implementieren.