

# Datenstrukturen

## ADT (Abstract Data Type)

Menge von Objekten zusammen mit einer Menge von Operationen auf diesen Objekten.  
 Was kann ich mit den Daten machen?

ADT	Methoden	Beschreibung
Primitive Datentypen: ganze Zahlen Fließkommazahlen boolean Text	Rechenoperatoren	
Stack, Stapel, LIFO	push(e) pop()	Zugriff auf <b>zuletzt</b> hinzugefügtes Element
Queue, Schlange, FIFO	enqueue(e) dequeue()	Zugriff auf <b>zuerst</b> hinzugefügtes Element
Priority Queue	enqueue(e, priority) dequeue()	Zugriff auf Element mit <b>höchster Priorität</b>
List, Sequence	insert(i, e) remove(i) get(i)	geordnet, Zugriff über <b>Index</b>
Sorted List	insert(i, e) remove(i) get(i)	immer sortiert
Map, Dictionary, Associative Array	set(key, value) delete(key) get(key)	Zugriff nur über <b>key</b>
Set, Menge	add(e) list() delete(e)	ungeordnet, unique
Graph	node (Knoten) edge (Kante)	Kanten können gerichtet und gewichtet sein
Tree, Baum	root (Wurzel) child (Kind)	Jedes Kind hat genau eine Mutter

## Datenstruktur

Konkrete Realisierung eines ADT mittels einer Programmiersprache.  
 Wie sind die Daten im Speicher organisiert?

- Array
- Linked List
- Double Linked List
- Tree
- Hash Table