

Abstrakte Datentypen

- Ein ADT besteht aus
 - Menge von Objekten
 - Operationen auf dieser Menge
 - Beschreibung der Semantik der Operationen (Regeln)
- Unabhängig von einer Programmiersprache, kann z.B. in natürlicher Sprache formuliert werden.
- Beschreibt *was* die Operationen tun, nicht *wie* sie es tun.
 - ⇒ *Kapselung* (Operationen bilden Schnittstelle),
Geheimnisprinzip (Implementierung bleibt verborgen)
- Bekannte Beispiele: Schlange, Prioritätsschlange, Keller,

ADT File

- Anordnung von Elementen des Typs t
- Elemente können in der Reihenfolge ihrer Anordnung von links nach rechts gelesen werden
- Beim Ersetzen eines Elementes werden die Nachfolgenden gestrichen
- Modell: Magnetband mit Schreib/Lesekopf, der von links nach rechts läuft.
- Signatur:

```
data File t = CreateFile | Write t | Read t
show  :: File t      -> t
skip  :: File t      -> File t
eof   :: File t      -> Bool
reset :: File t      -> File t
put   :: File t -> t -> File t
```

O–Kalkül, Rekurrenzen

- Rekurrenz: Wichtig bei Betrachtung rekursiver Funktionen.
- Funktionsaufwand setzt sich zusammen aus
 - Aufwand beim aktuellen Aufruf
 - Aufwand des rekursiven Aufrufs
- Schreibweise:

$$\underbrace{T(n)}_{\text{gesamt}} = \underbrace{O(f(n))}_{\text{aktueller Aufruf}} + \underbrace{k \cdot T(n')}_{\text{rekursiver Abstieg}}$$