

## **ADT**

Beim Arbeiten mit Daten treten immer wieder dieselben Strukturen auf, für die sich daher eine grundlegende Analyse zu ihrer Modellierung anbietet. Dabei muss vom speziellen Anwendungskontext und der gerade verwendeten Programmiersprache abstrahiert werden. Man beschreibt einen

### Abstrakten Datentyp

Die Eigenschaften eines ADT sind:

*Quelle: Rollke, LK Informatik*

#### **Universalität**

implementation independance

ein einmal entworfener und implementierter ADT kann von jedem anderen Programm benutzt werden

#### **präzise Schnittstellenbeschreibung**

precise specifications

die Schnittstelle zwischen ADT und Anwendung muss genau beschrieben sein. Schon bei ihrem Entwurf kommt es auf die größte Sorgfalt an, da spätere Änderungen der inneren Implementation zwar belanglos sind, eine geänderte Schnittstellendefinition hat aber katastrophale Auswirkungen auf die Anwender eines ADT.

#### **Einfachheit**

simplicity

versuche den Entwurf eines ADT so zu machen, dass sie für den Anwender so einfach wie möglich erscheint.

#### **Verdecken von Informationen**

information hiding

der Anwender eines ADT bekommt keine Informationen über den inneren Aufbau. Er kennt nur die Schnittstelle.

#### **Geschütztheit**

integrity

der Anwender eines ADT kann in den inneren Aufbau nicht eingreifen. Insbesondere kann er keine Daten ändern. Daten werden vom ADT grundsätzlich nur über die Schnittstelle zur Verfügung gestellt, können also nicht einmal direkt gelesen werden.

#### **Modularität**

modularity

ist das Ziel bei der Anwendung und sollte auch das Prinzip bei der Entwicklung von ADT sein.

#### **Kapselung**

encapsulating

ist der gemeinsame Oberbegriff fast aller o.a. Kriterien.