Abstrakte Datentypen

- **Idee**: Datenstrukturen + Algorithmen gehören zusammen
- Ein **abstrakter Datentyp** (*abstract data type*) bündelt die Trägermenge von **Objekten** mit den zugehörigen Operationen (**Methoden**)
- Beispiel:

  <N; 0, 1, +, -, *>

Abstrakte Datentypen

- Datenrepräsentation und Implementierung der Methoden bleiben hinter einer abstrakten **Schnittstelle** (*interface*) verborgen
- Daten in den Objekten können nur über die Methoden manipuliert werden
- **Geheimnisprinzip** (*information hiding*)
- Nur der Programmierer des abstrakten Datentyps muss die Datenrepräsentation und die Programmierung der Funktionen verstehen
- Benutzer braucht nur die Funktionsweise der Schnittstelle zu verstehen

Abstrakte Datentypen

- Java unterstützt Information Hiding nur teilweise
- Ein abstrakter Datentyp wird durch eine (Objekt-) **Klasse** implementiert (realisiert)
- Ein Objekt in einem abstrakten Datentyp ist im allgemeinen ein Verbund-Objekt, das Daten als Objekt-**Zustand** speichert
- Auf den Objekten operieren Methoden, die in der Klasse implementiert sind und über ein Objekt aufgerufen werden können.

Abstrakte Datentypen

- Ein **Objekt** ist ein (Daten-)Verbund zusammen mit den Algorithmen (Funktionen, Methoden), die auf Verbunden dieses Typs operieren können
  - jedes Objekt einer Klasse hat eigenen Zustand
  - alle Objekte einer Klasse haben die gleichen Methoden
  - Daten gleich Objekt-**Zustand**
  - Algorithmen (Methoden) gleich Objekt-**Verhalten**

- **Information Hiding**
  - Manche Methoden können von außen auf dem Objekt aufgerufen werden
  - Andere Methoden sind von außen unsichtbar
  - Sichtbare Methoden definieren die Schnittstelle (*interface*) des Objekts

Abstrakte Datentypen

- **Klasse** fasst gleichartige Objekte zusammen mit
  - gleichen Methoden und
  - Daten gleichen Typs
- **Klassendeklaration**
  - definiert den Typ des Datenverbundes
  - listet die Implementierung der Methoden

Abstrakte Datentypen

- Objekt ist eine konkrete **Instanz** einer Klasse
  - ein Exemplar eines Verbundes mit separaten Daten
  - gebündelt mit den für die Klasse definierten Methoden
  - jede Methode wird auf einem konkreten Objekt aktiviert
- Beispiele
  - Rational r = new Rational(2,4);
  - r.normalize();
  - bool b = r.isZero();

Abstrakte Datentypen

- **Objektorientierte Sprache**
  - erlaubt es, die Kopplung mit Sprachkonstrukten zu fixieren
  - statt sie der Selbstdisziplin des Programmierers zu überlassen
- Außerdem höhere objektorientierte Konzept wie z.B. **Vererbung**