

Nachteile von Object-Listen:

- Werte verschiedener Typen durcheinander
- Um Werte aus der Liste wieder als z.B. Bruch zu verarbeiten, muss man von Object auf Bruch casten.

Lösung: Generische Klassen/
Generische Typen

1 Klasse definiert viele Typen

Liste < Bruch >

Liste < Integer >

⋮

Parametrischer Polymorphismus:

Typparameter T erlaubt es,
die gleiche Implementierung
für verschiedene Typen zu
benutzen. (Kammt aus der
fkt. Prog.)

Ad hoc Polymorphismus:

Verschiedene Implementierungen
der gleichen Methode. Zur Lauf-
zeit wird aufgrund des Typs
des Objekts entschieden, welche

Implementierung ausgeführt.

(Kommt aus OO-Präg.)

Weitere Möglichkeiten für

Typparameter/Typebounds:

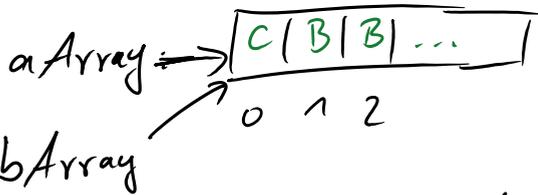
- Es gibt Einschränkungen nach oben/unten in Typhierarchie (extends/super)
- Wildcards: ?

:

⇒ Kombination von parametr.

Polymorphismus mit Klassenhierarchie.

Array-Bsp:



⇒ Fehler beim Entwurf von Java, bei gener. Typen korrigiert.