

Generics und Polymorphie

Carsten Gips (FH Bielefeld)

Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

Generische Polymorphie

`B<E> extends A<E>`

```
class Vector<E> { ... }  
class Stack<E> extends Vector<E> { ... }  
  
Vector<Double> vd = new Stack<Double>();  
Vector<String> vs = new Stack<String>();
```

=> Polymorphie bei Generics bezieht sich auf **Typ** (nicht Typ-Parameter)

Polymorphie bei Generics bezieht sich nur auf Typ!

“B extends A” bedeutet
nicht “C extends C<A>”

```
Stack<Number> s = new Stack<Integer>(); // DAS GEHT SO NICHT!
```

```
// Folgen (wenn obiges gehen wuerde):
```

```
s.push(new Integer(3)); // das ginge sowieso ...
```

```
// Folgen (wenn obiges gehen wuerde):
```

```
// Stack<Number> waere Oberklasse auch von Stack<Double>
```

```
s.push(new Double(2.0)); // waere dann auch erlaubt ...
```

```
// Das Objekt (Stack<Integer>) kann aber keine Double speichern!
```

```
// Zur Laufzeit keine Typ-Informationen mehr!
```

Abgrenzung: Polymorphie bei Arrays

Wenn "B extends A" dann "B[] extends A[]"

```
Object[] x = new String[] {"Hello", "World", ":--"};
x[0] = "Hallo";
x[0] = new Double(2.0); // Laufzeitfehler
String[] y = x; // String[] ist KEIN Object[]!!!
```

- Arrays besitzen Typinformationen über gespeicherte Elemente
- Prüfung auf Typ-Kompatibilität zur **Laufzeit** (nicht Kompilierzeit!)

Arrays vs. parametrisierte Klassen

=> Keine Arrays mit parametrisierten Klassen!

```
Foo<String>[] x = new Foo<String>[2];    // Compilerfehler
```

```
Foo<String[]> y = new Foo<String[]>();    // OK :)
```

Vererbung:

- IS-A-Beziehung
- Anwendung: Vererbungsbeziehung vorliegend, Eigenschaften verfeinern
- Beispiel: Ein Student *ist eine* Person

Generics:

- Schablone (Template) für viele Datentypen
- Anwendung: Identischer Code für unterschiedliche Typen
- Beispiel: Datenstrukturen, Algorithmen generisch realisieren

- Generics: Vererbung und Überladen möglich, aber:

Aus “`U extends O`” folgt **nicht “`A<U> extends A<O>`”**

- Achtung: Bei Arrays gilt aber: Wenn “`U extends O`” dann gilt auch “`U[] extends O[]`” ...

LICENSE



Unless otherwise noted, this work is licensed under CC BY-SA 4.0.