

Generisches Programmieren

am Beispiel von Java 5

- + **Warum generische Programmierung?**
- + **Implementierung einer generischen Maximumsfunktion**
 - + Anwendungsbeispiel
- + **Varianzfälle**
 - + Kovarianz
 - + Invarianz
- + **Konkrete Anwendungsfälle**
- + **Probleme bei der praktischen Umsetzung**
- + **Quellen**

Generische Programmierung Paradigma



Ziele:

- ✚ Nur eine einzige Implementierung eines Algorithmus, der auf vielen verschiedenen Datenstrukturen operiert
- ✚ Verminderung des Wartungsaufwands
- ✚ Typprüfung während der Kompilation

Implementierung einer generischen Maximumsfunktion

Definition Typparameter T

```
1 public class Compare {  
2   public static <T extends Comparable<T>> T max (T a, T b) {  
3     if (a.compareTo(b) >= 1) {  
4       return a;  
5     } else  
6       return b;  
7   }  
8 }
```

Grenze / Schranke (Bound)

Vergleich beider Argumente

Anwendungsbeispiel

```
Integer zahl = Compare.max(10, 30);  
System.out.println(zahl);
```

Ausgabe: 30

// ---

```
String string = Compare.max("a", "b");  
System.out.println(string);
```

Ausgabe: b

Kovarianz

Typparameter muss einem bestimmtem Datentyp entsprechen oder von ihm abgeleitet sein

Invarianz

Typparameter wird durch genau einen bestimmten Datentyp ersetzt

Anwendungsbeispiel

Parametrisierung des generischen Typs

```
Keller<Integer> keller = new Keller<Integer>();
keller.push(10);
Integer zahl = keller.top();
```

Implementierung des Kellers

```
public class Keller<T> {
    public void push(T wert) {
        // [...]
    }
    public T top() {
        // [...]
    }
}
```

Generischer Typ
Typparameter

Java Klassenbibliothek

Beispiele:

1. Hashtable
2. Bäume
3. Listen beliebiger Objekte

Probleme bei der praktischen Umsetzung

- ✚ Programmiersprache (und zugehöriger Compiler) muss generisches Programmieren ermöglichen (Sprachunterstützung)
- ✚ Sprache wird komplexer
- ✚ Klassenbibliothek muss angepasst sein
- ✚ Alter (nicht generischer) Code muss umgeschrieben werden, meist nicht praktikabel

Zwischenlösung: Hüllschicht zwischen neuem, generischen und altem Code (Wrapper)

- ✚ Jazayeri, Loos, Barke: Generic Programming '98, Springer Verlag 2000
- ✚ Schneider, Werner: Taschenbuch der Informatik, Fachbuchverlag Leipzig 2001
- ✚ Langer, Kreft: Java Generics - Parametrisierte Typen und Methoden, JavaMagazin April 2004
- ✚ Wikipedia (DE) Stichwort: Generische Programmierung Stand vom 21.06.2005

Verwendete Software:

- ✚ J2SE™ Development Kit 5.0 Update 3
- ✚ Eclipse IDE 3.1 RC 3

Vielen Dank für Ihre Aufmerksamkeit!