



# Universität Karlsruhe (TH)

## Lehrstuhl für Programmierparadigmen

Fortgeschr. Objektorientierung SS 2009      <http://pp.info.uni-karlsruhe.de/>  
Dozent: Prof. Dr.-Ing. G. Snelting      [snelting@ipd.info.uni-karlsruhe.de](mailto:snelting@ipd.info.uni-karlsruhe.de)  
Übungsleiter: Andreas Lochbihler      [lochbihl@ipd.info.uni-karlsruhe.de](mailto:lochbihl@ipd.info.uni-karlsruhe.de)  
Dennis Giffhorn      [giffhorn@ipd.info.uni-karlsruhe.de](mailto:giffhorn@ipd.info.uni-karlsruhe.de)

Blatt 10      Ausgabe: 25.06.2009      Besprechung: 01.07.2009

### 1. Visitor Pattern und Multimethoden

Sie haben eine Hierarchie mit folgenden Java-Klassen:

---

```
1 abstract class Expression { }
2
3 public class IntExpression extends Expression {
4     int value;
5 }
6
7 public class AddExpression extends Expression {
8     Expression e1, e2;
9 }
10
11 public class MultExpression extends Expression {
12     Expression e1, e2;
13 }
```

---

Nun möchten Sie nachträglich eine *prettyPrint()*-Methode implementieren, welche eine Expression korrekt formatiert ausgibt, ohne sie direkt in die Klassen zu implementieren. Welche Probleme entstehen dabei? Lösen Sie dieses Problem einmal durch Anwendung des *Visitor Patterns* und einmal mit *Multimethoden*. Welche Vor- und Nachteile haben beide Ansätze?

## 2. Mixins in Scala

- (a) Schreiben Sie einen `trait Ordered`, der die Vergleichsmethoden `<`, `<=`, `>` und `>=` für eine totale Ordnung bereitstellt und diese auch, soweit möglich, bereits implementiert.
- (b) Implementieren Sie nun eine Datumsklasse `Date`, welche diesen `trait` verwendet.
- (c) Überlegen Sie sich, wie man dies in Java realisieren würde und diskutieren Sie beide Ansätze.

## 3. Iteratoren durch Mixins in Scala

Betrachten Sie folgenden `trait` für einen abstrakten Iterator:

---

```
1      trait AbsIterator {  
2          type T;  
3          def hasNext: boolean;  
4          def next: T;  
5      }
```

---

- (a) Definieren Sie jetzt einen davon abgeleiteten `trait RichIterator`, der eine `foreach`-Schleife des Iterators implementiert, welcher eine Funktion übergeben werden kann, die `unit` (Scalas `void`) zurückliefert. Welche Eigenschaft von `traits` nützen Sie dabei aus?
- (b) Definieren Sie einen weiteren, von `AbsIterator` abgeleiteten `trait SyncIterator`, der jeden Aufruf von `hasNext` und `next` synchronisiert. Was ist hierbei die Besonderheit?
- (c) Definieren Sie nun noch eine konkrete `StringIterator` Klasse, welche unter Verwendung des `trait AbsIterator` die einzelnen Zeichen eines Strings zurückgibt. Was müssen Sie dabei beachten?
- (d) Zum Schluss benötigen wir noch ein konkretes Testprogramm, das synchronisiert die Zeichen eines Strings mittels einer `foreach`-Schleife als Spalte untereinander ausgibt.