



# Praktische Informatik 3: Funktionale Programmierung

## Vorlesung 13 (24.01.2023): Übungen

Christoph Lüth



Deutsches  
Forschungszentrum  
für Künstliche  
Intelligenz GmbH



Universität  
Bremen

Wintersemester 2022/23

# Your Turn

## Übung 13.1: Scala die Erste

Ladet Euch die Quellen für die Vorlesung von unser Webseite, und den Scala-Interpreter und Compiler von

`https://www.scala-lang.org/`

herunter. Startet den Interpreter (`scala`), ladet das Beispiel oben mit

```
scala> :load 02-Rational-2.scala
```

Was passiert, wenn ihr ein `Rational`-Objekt konstruiert, bei dem die Vorbedingung verletzt ist?

# Your Turn

## Übung 13.1: Scala die Erste

Ladet Euch die Quellen für die Vorlesung von unser Webseite, und den Scala-Interpreter und Compiler von

<https://www.scala-lang.org/>

herunter. Startet den Interpreter (`scala`), ladet das Beispiel oben mit

```
scala> :load 02-Rational-2.scala
```

Was passiert, wenn ihr ein `Rational`-Objekt konstruiert, bei dem die Vorbedingung verletzt ist?

**Lösung:** Es gibt (wenig überraschend) eine Exception:

```
scala> new Rational(6,0)
java.lang.IllegalArgumentException: requirement failed
    at scala.Predef$.require(Predef.scala:327)
    ... 29 elided
```

# Your Turn

## Übung 13.2: Scala die Zweite

Öffnen Sie die Datei `02-Expr-fold.scala`, und vervollständigen Sie die Definition der `fold`-Funktion für den Datentyp `Expr`.

# Your Turn

## Übung 13.2: Scala die Zweite

Öffnen Sie die Datei 02-Expr-fold.scala, und vervollständigen Sie die Definition der fold-Funktion für den Datentyp Expr.

Lösung:

```
def fold[A]( v: String ⇒ A
            , n: Double ⇒ A
            , p: (A, A) ⇒ A
            , m: (A, A) ⇒ A
            , t: (A, A) ⇒ A
            , d: (A, A) ⇒ A): A = this match
{
  case Var(variable) ⇒ v(variable)
  case Num(num) ⇒ n(num)
  case Plus(e1, e2) ⇒ p( e1.fold(v, n, p, m, t, d)
                        , e1.fold(v, n, p, m, t, d))
  case Minus(e1, e2) ⇒ m( e1.fold(v, n, p, m, t, d)
                        , e1.fold(v, n, p, m, t, d))
  case Times(e1, e2) ⇒ t( e1.fold(v, n, p, m, t, d)
                        , e1.fold(v, n, p, m, t, d))
  case Div(e1, e2) ⇒ d( e1.fold(v, n, p, m, t, d)
                      , e1.fold(v, n, p, m, t, d))
}
```

# Your Turn

## Übung 13.3: Scala die Dritte

Betrachten Sie folgendes Beispiel:

```
class Function[S, T] {  
  def apply(x:S) : T  
}
```

Wie müssen hier die Varianz-Annotation für die Typvariablen **S** und **T** lauten?

# Your Turn

## Übung 13.3: Scala die Dritte

Betrachten Sie folgendes Beispiel:

```
class Function[S, T] {  
  def apply(x:S) : T  
}
```

Wie müssen hier die Varianz-Annotation für die Typvariablen **S** und **T** lauten?

Lösung:

```
class Function[-S, +T] {  
  def apply(x:S) : T  
}
```