

# Funktionale und objektorientierte Programmierung

Mitschrift

Fabian Damken

25. Oktober 2016

# Inhaltsverzeichnis

<b>1</b>	<b>Programmierung</b>	<b>3</b>
1.1	Strukturmechanismen . . . . .	3
1.2	Programmentwicklung . . . . .	3
<b>2</b>	<b>DrRacket</b>	<b>4</b>
2.1	Strukturmechanismen . . . . .	4
2.1.1	Primitive Sprachelemente . . . . .	4
2.1.2	Kombinationsmittel . . . . .	4
2.1.3	Abstraktionsmittel . . . . .	4
2.1.4	Auswertungsregeln . . . . .	5
2.1.5	define - Regel . . . . .	5
2.1.6	Prozeduren . . . . .	5
2.2	Programmentwicklung . . . . .	6
2.2.1	Zweck verstehen . . . . .	6
2.2.2	Beispiel ausdenken . . . . .	6
2.2.3	Testen . . . . .	6

# 1 Programmierung

## 1.1 Strukturmechanismen

- Primitive Ausdrücke
  - Zahlen (selbstausswertend)
  - Boolesche Werte (selbstausswertend)
  - Eingebaute Prozeduren (bspw. +, -, \*, /, ...)
- Kombinationsmittel
- Abstraktionsmittel

## 1.2 Programmentwicklung

1. Zweck verstehen.
2. Beispiel ausdenken.
3. Implementieren.
4. Testen.

## 2 DrRacket

### 2.1 Strukturmechanismen

#### 2.1.1 Primitive Sprachelemente

- Ganze Zahlen [bspw. 3]
- Endliche Dezimalzahlen (Punkt statt Komma) [bspw. 2.3]
- Brücke (ohne Leerzeichen) [bspw. 7/3]
- Unendliche Zahlen ohne Periode (beginnend mit #i) [bspw. #i2.718]

#### 2.1.2 Kombinationsmittel

Kombinationen werden in DrRacket mit Hilfe der Präfixdarstellung dargestellt. Das heißt dass der gesamte Ausdruck in Klammern gefasst wird, erst der Operator und dann mögliche Operanden getrennt mit einem Leerzeichen.

Beispiele:

- (+ 2 3)
- (+ 4 (\* 2 3))

Kombination  $\iff$  Prozedur

#### 2.1.3 Abstraktionsmittel

1. Erstellen einer komplexen Sache.
2. Benennung der komplexen Sache.
3. Nutzung als primitive Sache.

Beispiele:

- (define score (+ 27 3))
- (define PI 3.1415)

## 2.1.4 Auswertungsregeln

- Selbstauswertend (self-rule) [bspw. 2, true]
- Eingebauter Operator [bspw. +, \*]
- Name/Variablen (name-rule)
- Kombination [bspw. (+ 2 3)]
  1. Werte Unserausdrücke aus.
  2. Wende die Prozedur an. [Beispiel: (+ 4 (\* 2 3)) = (+ 4 6) = 10]
- Sonderformen

## 2.1.5 define - Regel

- Operand 1 := Name
- Operand 2 := Wert
- Wertet nur den Wert aus.
- Der Wert wird an den Namen gebunden.
- Kein Rückgabewert.

## 2.1.6 Prozeduren

### 2.1.6.1 Kreisfläche

Berechnet die Fläche eines Rings.

**Definition:** (define (area-of-disk r) (\* 3.14 (\* r r)))

**Nutzung:** (area-of-disk 5) = 78.5

### 2.1.6.2 Ringfläche

Berechnet die Fläche eines Rings durch die Subtraktion der Fläche des Innenrings von der Fläche des Außenrings.

**Definition:** (define (area-of-ring outer inner) (- (area-of-disk outer) (area-of-disk inner)))

**Nutzung:** (area-of-ring 5 3) = 50.24

## 2.2 Programmentwicklung

### 2.2.1 Zweck verstehen

```
;; Funktionsname :: Parametertypen -> Ergebnistyp  
;; Kurze Beschreibung.  
(define ...)
```

### 2.2.2 Beispiel ausdenken

```
;; Beispiel: (Funktionsname Beispielparameter) = Beispielergebnis
```

### 2.2.3 Testen

- `(check-expect test expected)` - Prüfung auf exakte Gleichheit.
- `(check-within test expected delta)` - Prüfung auf ungefähre Gleichheit in dem gegebenen Toleranzrahmen (`delta`).
- `(check-error test message)` - Prüfung auf das Auslösen eines Fehlers.