



INFORMATIK I

Tutorium 29.11.2016

BESPRECHUNG

Blatt 5



WIEDERHOLUNG

.....
Vorlesung & Für Blatt 6



WIEDERHOLUNG: LET

- Mit let können wir Ausdrücke an Variablen binden **innerhalb** einer Prozedur
- Syntax: (let ((<id1> <e1>)) ... ((<id_n> <e_n>)) <e>)
- Wann wird es verwendet? —> Z.b. wenn wir nicht mehrmals die selbe Prozedur mit den selben Parametern aufrufen wollen (-> Ressourcen sparen, Zeit sparen)

```
(: test (%a -> %a))
(check-expect (test 5) 5)
(check-expect (test "hi") "hi")
(define test
  (lambda (x)
    (let ((y x))
      y)))
```

WIEDERHOLUNG: VIOLATION

- Manchmal wollen wir ein Programm mit einer Fehlermeldung abbrechen
- Syntax: (violation <string>)

```
(: test2 (number -> number))  
(check-expect (test2 3) 3)  
(check-error (test2 -3) "Zahl muss größer 0 sein")  
(define test2  
  (lambda (n)  
    (if (> n 0) n  
        (violation "Zahl muss größer 0 sein")))))
```

WIEDERHOLUNG: REKURSION

- Endlich Rekursion :)
- **Grundgedanke:** Funktion ruft sich selbst mit veränderten Parametern auf bis Abbruchbedingung eintrifft
- Beispiel: Fibonacci zahlen. $\text{Fib}(0) = 0$, $\text{Fib}(1) = 1$, $\text{Fib}(2) = 1$, $\text{Fib}(3) = 3$, ..., $\text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2)$

```
(: fib (natural -> natural))
(check-expect (fib 0) 0)
(check-expect (fib 1) 1)
(check-expect (fib 3) 2)
(define fib
  (lambda (n)
    (if (<= n 1)
        n
        (+ (fib (- n 2)) (fib (- n 1))))))
```

WIEDERHOLUNG: REKURSION AUF LISTEN

- **Grundprinzip:** Ruf Prozedur mit (rest ls) auf, bis leere Liste entsteht —> Dann Abbruch, arbeite zuvor mit (first ls)
- Beispiel: Summe einer Liste

```
(: list-sum ((list-of number) -> number))  
(check-expect (list-sum (make-pair 1 (make-pair 3 empty))) 4)  
(check-expect (list-sum empty) 0)  
(define list-sum  
  (lambda (ls)  
    (if (empty? ls)  
        0  
        (+ (first ls) (list-sum (rest ls))))))
```

ÜBUNGSAUFGABEN

.....

- Schreibe eine Prozedur, die alle geraden Zahlen einer Liste addiert
- Schreibe eine Prozedur, die jedes 2. Element einer Liste addiert
- Schreibe ein Prozedur, die die Länge einer Liste berechnet
- Schreibe eine Prozedur, die den Mittelwert einer Liste mit Zahlen berechnet
- Schreibe eine Prozedur, die alle Elemente einer Liste quadriert und diese anschließend addiert
- Schreibe eine Prozedur, die die Varianz einer Liste aus Zahlen berechnet
- Schreibe eine Prozedur, die alle Zahlen einer Liste durch n teilt und diese erneut als Liste zurückgibt