Häufig gestellte Fragen

Michael Sperber

sperber@deinprogramm.de

Erkenntnisse

- Programmierausbildung ist möglich
  - sicher ab 5. Klasse
- erfordert grundlegendes Umdenken:
  - Didaktik
  - Programmiersprache
  - Programmierumgebung

Didaktik

- Konstruktionsanleitungen
- systematischer Zugang
- datengesteuerter Programmentwurf

Konstruktionsanleitungen

- Problemanalyse
- Funktionsdefinition
- Korrektheitsprüfung

Problemanalyse

- Verständnis
- Vertrag
- Beispiele/Testfälle

Funktionsdefinition

- Gerüst
- Schablone
- Rumpf

Korrektheitsprüfung

- Korrekturlesen
- Syntaxüberprüfung
- Test

Rekursion über Zahlen

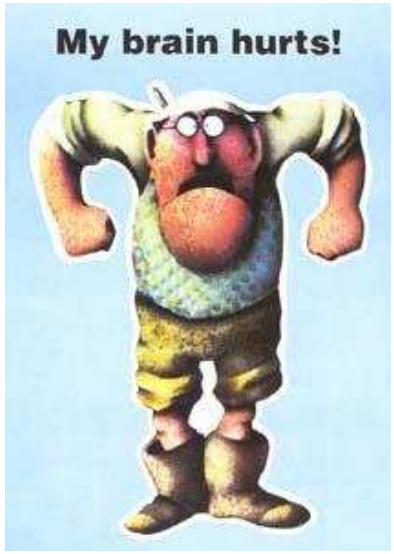
"Es ist ganz einfach!"

```
(define (factorial n)
  (cond
    ((= n 1) 1)
    (else (* n
             (factorial (- n 1))))))
```

Rekursion über Zahlen

"Es ist ganz einfach!"

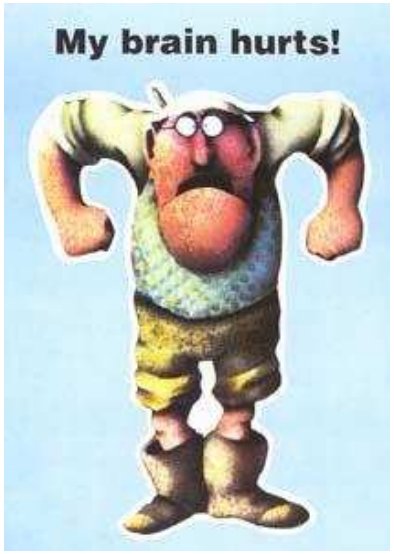
```
(define (fibonacci n)
  (cond
    ((= n 0) 0)
    ((= n 1) 1)
    (else (+ (fibonacci (- n 1))
             (fibonacci (- n 2))))))
```



Rekursion über Zahlen

"Es ist ganz einfach!"

```
(define (gcd a b)
  (cond
    ((= b 0) a)
    (else (gcd b (modulo a b)))))
```



Pascal

f(x) := x + 5

```
PROGRAM f(INPUT, OUTPUT);

FUNCTION f(x : INTEGER) : INTEGER;
BEGIN
  f := x + 5
END;

VAR x :  INTEGER;

BEGIN
   ReadLn(x) ;
   WriteLn(f(x))
END.
```

Ketchup vs. Kaviar

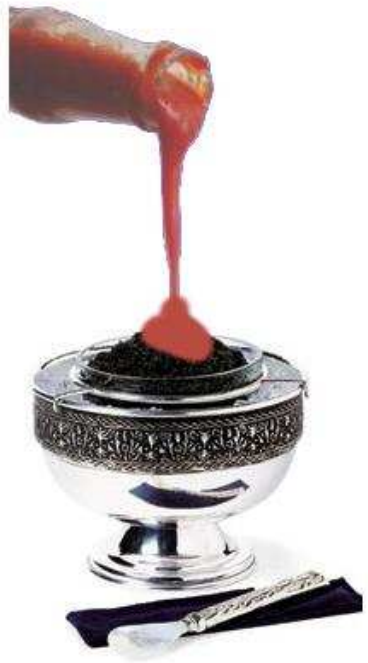
f(x) := x + 5

```
PROGRAM f(INPUT, OUTPUT);

FUNCTION f(x : INTEGER) : INTEGER;
BEGIN
  f := x + 5
END;

VAR x :  INTEGER;

BEGIN
   ReadLn(x) ;
   WriteLn(f(x))
END.
```



Delphi / Eclipse / JBuilder / Emacs

```
...
y + 1 := x;
...
```

warning: expression used as a statement - value is ignored
parse error before `+'

Ursache:

Entwicklungsumgebungen für Profis gedacht, nicht Anfänger

Curriculum

- einfache Daten
- zusammengesetzte Daten
- gemischte Daten
- Syntax und Semantik
- Listen und strukturelle Rekursion

Ausblick Curriculum

- selbstreferentielle Datenstrukturen
- Rekursion über Zahlen
- iterative Verfeinerung
- Abstraktion als Konstruktionsprinzip
- Prozeduren als Werte
- generative Rekursion
- Aufwand von Algorithmen
- Vektoren und Mutation
- Zuweisungen
- verschiedene Begriffe von Gleichheit

Objects First

- Methoden
- Instanzvariablen
- Konstruktoren

Objektorientierte Programmierung

- zusammengesetzte Daten
- Zustand
- Polymorphie
- Subtyping
- dynamische Bindung
- Message-Passing Style

Von Scheme zu Java

```
(define-struct book (title year))
```

```
class Book {
  String title;
  int year;

  Book(String title, int year) {
    this.title = title;
    this.year = year;
  }
}
```

Übergang zu OOP

- Konstruktion von Klassen
- Interfaces
- rein funktionale Methoden
- Gemeinsamkeiten -> Oberklassen
- Kapselung

OOP

- Mutation
- Datenstrukturen
- Traversierungen abstrahieren
- Collections
- Models und Views

EPAs Informatik

Einsatz verschiedener grundlegender Betrachtungsweisen im Rahmen von Problemlösungen, Kenntnisse der folgenden Modellierungstechniken, mindestens zwei im Grundkursfach und mindestens drei im Leistungskursfach:

- Objektorientierte Modellierung
- Datenmodellierung
- Zustandsorientierte Modellierung
- Modellierung von Abläufen mit Algorithmen
- Funktionale Modellierung
- Regelbasierte Modellierung

Warum nicht Haskell?

```
queens_kill :: Int -> [Int]  -> Bool
queens_kill locn places =
   helper places 1
       where
       helper [] _ = False
       helper (hd:tl) offset = locn == hd           |
                               locn == hd + offset  |
                               locn == hd - offset  |
                               helper tl offset+1
```

Warum nicht Haskell?

```
Prelude> :load nq.hs
Reading file "nq.hs":
Type checking
ERROR "nq.hs":2 - Instance of Num Bool
 required for definition of queens_kill
```

TeachScheme!

- Matthias Felleisen, Northeastern University
- Brown University, University of Chicago, University of Utah, Adelphi University ...
- Informatik-Unterricht an High Schools
- Programmierumgebung DrScheme
- How to Design Programs
- demnächst: How to Design Class Hierarchies
- viele Universitäten
- > 250 ausgebildete Lehrer