

Der Weg zur funktionalen Softwarearchitektur

Michael Sperber

@sperbsen





- Individualsoftware
- branchenunabhängig
- funktionale Programmierung
- Schulungen, Coaching

www.active-group.de
funktionale-programmierung.de
bobkonf.de



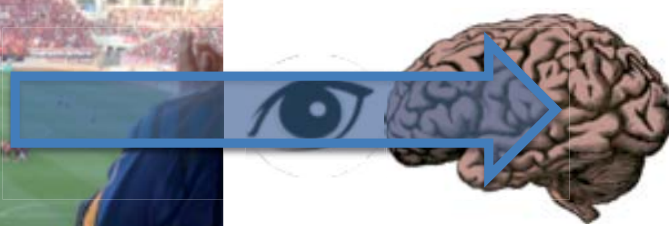
Eine Welt von Objekten



(© Trustees of the British Museum)

@active group

Realität und Schnappschuss





Haskell

```
data Animal =  
    Dillo Bool Double -- lebendig?, Gewicht  
| Rattlesnake Double Double -- Dicke, Länge
```

Daten und Information

```
d1 = Dillo True 10
    -- Gürteltier, lebendig, 10kg
d2 = Dillo False 12
    -- Gürteltier, tot, 12kg

r1 = Rattlesnake 10 150
    -- Klapperschlange, Dicke 10cm, Länge 150cm
r2 = Rattlesnake 0 170
    -- Klapperschlange, platt, Länge 170cm
```

Leben & Sterben auf dem Highway

```
runOverAnimal :: Animal -> Animal
runOverAnimal (Dillo _ weight) =
  Dillo False weight
runOverAnimal (Rattlesnake _ length) =
  Rattlesnake 0 length
```


Slalom

```
highway = [d1, r1, d2, r2]
```

```
runOverAnimals :: [Animal] -> [Animal]
```

```
runOverAnimals as = fmap runOverAnimal as
```

```
fmap :: (a -> b) -> [a] -> [b]
```

Kommunikation zwischen Programmteilen

```
runOverAnimal :: Animal -> Animal
```

vs.

```
abstract class Animal {  
    public void runOver();  
}
```



Effektive FP-Architektur



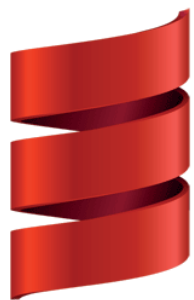
Von FP profitieren?



Technologien



F#



Scala

ABLE

EQUALS

Ergebnisorientierte Qualitätssicherung in sozialpädagogischen Einrichtungen



EQUALS ist ein Instrument, das neben der Abklärung der psychischen Gesundheit von jungen Menschen die Hilfen zur (Heim-)Erziehung pädagogisch dokumentiert, um den gesetzten Qualitätsstandards in der Jugendhilfe selbstverpflichtend zu entsprechen.



[Kontakt](#)

[Aktuelle Version](#)

EQUALS

- in Entwicklung seit 2010
- Fragebögen
- Datenhaltung
- Reports und Charts
- Standalone- und Client/Server-Versionen

Fragebogen in EQUALS - alt

FTNA – Nikotinabhängigkeit

Hast du jemals eine Zigarette geraucht?

nein ja

Wann nach dem Aufstehen rauchst du deine erste Zigarette?

innerhalb von 5 Minuten 6 bis 30 Minuten 31 bis 60 Minuten nach 60 Minuten

Findest du es schwierig, an Orten, wo das Rauchen verboten ist (z.B. Kirche, Bücherei, Kino usw.) das Rauchen zu unterlassen?

nein ja

Auf welche Zigarette würdest du nicht verzichten wollen?

die Erste am Morgen andere

Wie viele Zigaretten rauchst du im Allgemeinen pro Tag?

1 - 10 11 - 20 21 - 30 31 und mehr

Rauchst du am Morgen im Allgemeinen mehr als am Rest des Tages?

nein ja

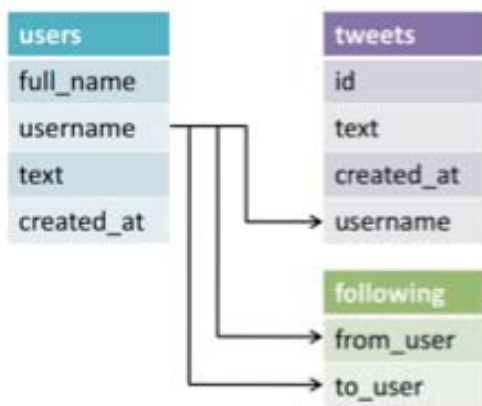
Kommt es vor, dass du rauchst, wenn du krank bist und tagsüber im Bett bleiben musst?

nein ja

Ton

Abschließen Abbrechen

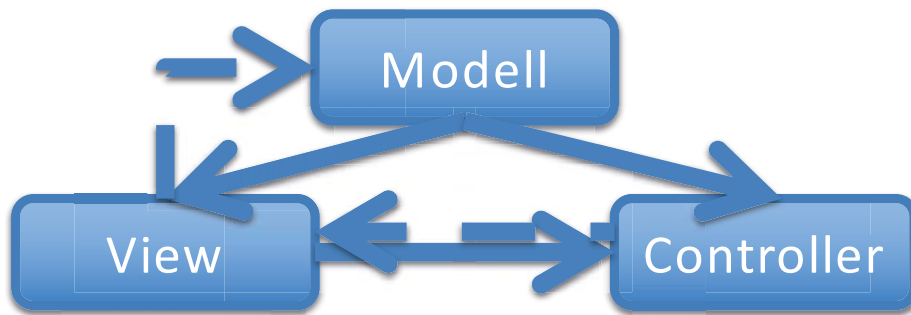
EQUALS 2013



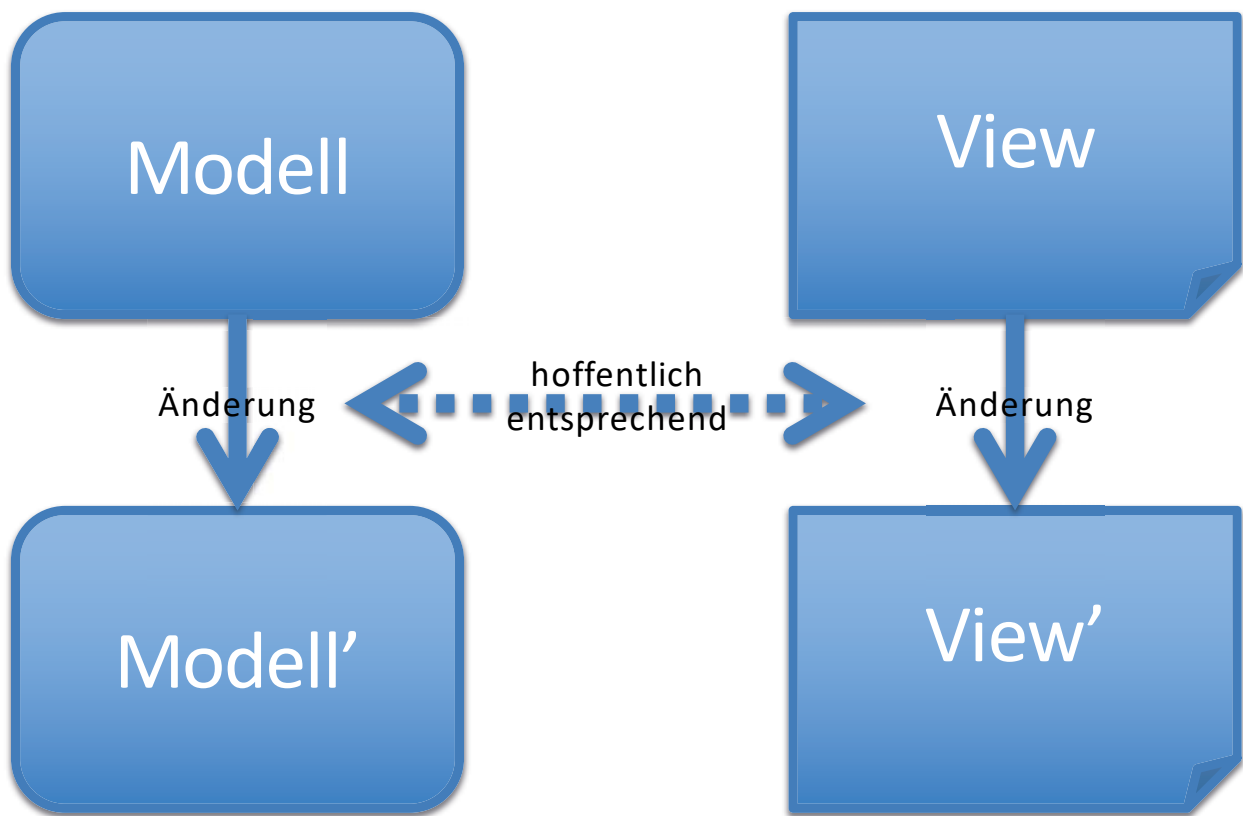
≈150,000locs Java



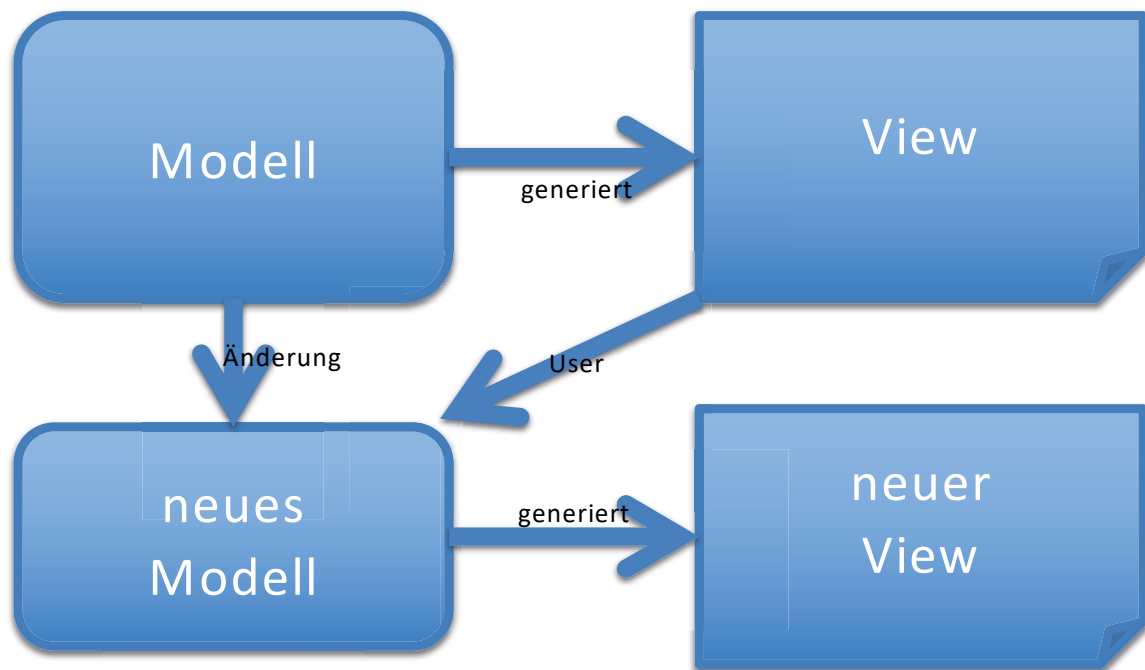
MVC



Problem



Reacl



Beispiel: Todo-Manager

TODO

Zap Make money

Zap retire


die Add #2

Applikations-Zustand

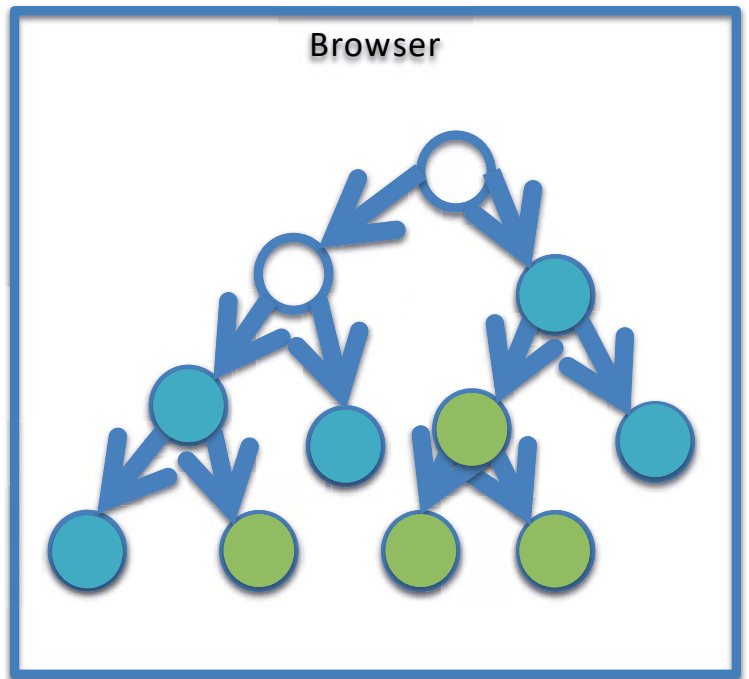
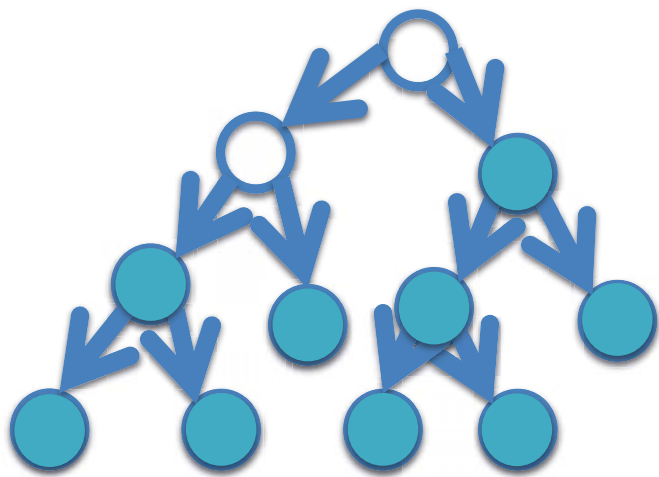
```
(defrecord Todo [id text done?])  
(def t1 (->Todo 0 "Make money" false))  
(def t2 (->Todo 1 "retire" false))  
  
(defrecord TodosApp [next-id todos])  
(def ts (->TodosApp 2 [t1 t2]))
```

Einzelnes Todo

```
(react/defclass to-do-item
  this todo []
  render
    (dom/div (dom/input
      {:type "checkbox"
       :value (:done? todo)})
    (dom/button "Zap")
    (:text todo)))
```



Rerendering



Fragebogen in EQUALS - neu

TigerKa

Allgemeine Kompetenzen Kompetenz einschätzung Schläsbeurteilung

TigerKa

Kommunikationsfähigkeit

Positiv-Pol **Negativ-Pol**

Der junge Erwachsene

- bringt sich in Gesprächen ein
- nimmt mit unterschiedlichen Menschen Kontakt auf
- bezieht in Gesprächen Stellung
- lässt andere Meinungen gelten und äußert sich ohne andere abzuwerten
- hört zu, berichtet auch über sich selbst / teilt sich mit
- nimmt Blickkontakt auf und hält ihn auch

Der junge Erwachsene

- redet in Kontakt- und Gesprächssituationen fast nicht
- verhält sich ablehnend und nimmt keinen Kontakt auf
- beginnt Gespräche nicht von sich aus und nimmt an ihnen auch nicht teil
- zieht sich übermäßig zurück und bleibt auf Ansprache auch wortkarg
- zeigt Schwierigkeiten sich verbal und mimisch mitzuteilen
- vermeidet Blickkontakt und hält ihn auch nicht

TigerKa zeigt das Zielverhalten, so wie im Positiv-Pol beschrieben

immer meistens häufig **öfter** manchmal selten nie

Ist TigerKa motiviert sich in dieser Kompetenz zu verbessern

ja, sehr ja **teils/teils** etwas nein

Seite 1 von 11

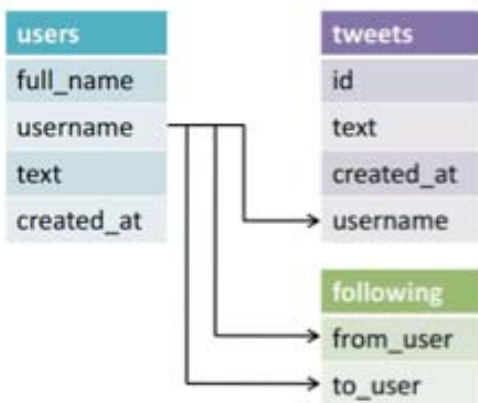
ⓧ ⏸ ⏹

Vorgehensweise

1. neue Funktionalität mit FP
2. reife Abstraktionen und Frameworks
3. Ersatz des alten Codes

EQUALS 2015

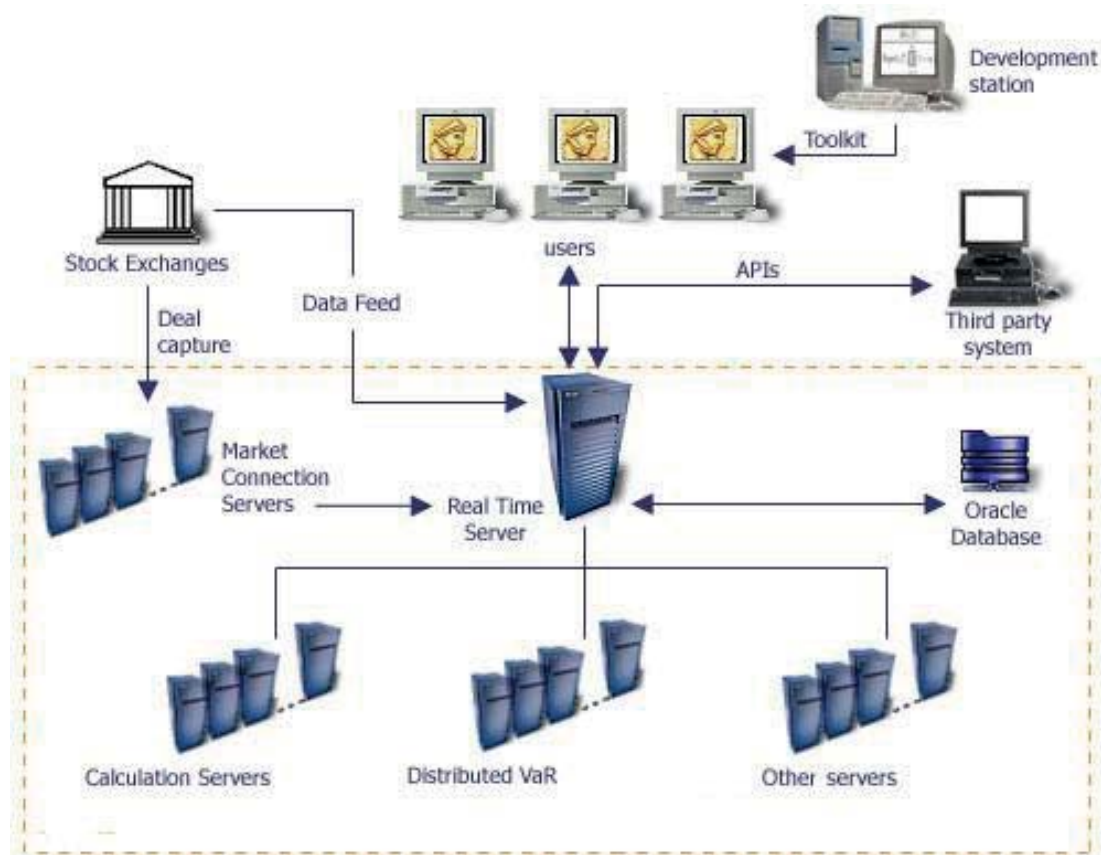
- 75,000loc Java
- 42,000loc Scala
- 75,000loc ClojureScript



EQUALS 2018

- 50,000loc Java
- 45,000loc Scala
- 85,000loc ClojureScript

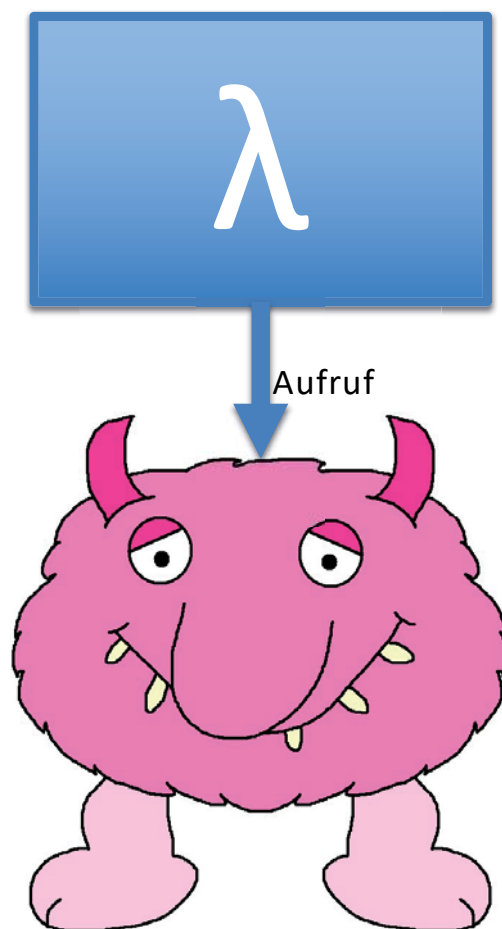
Management von Finanzderivaten



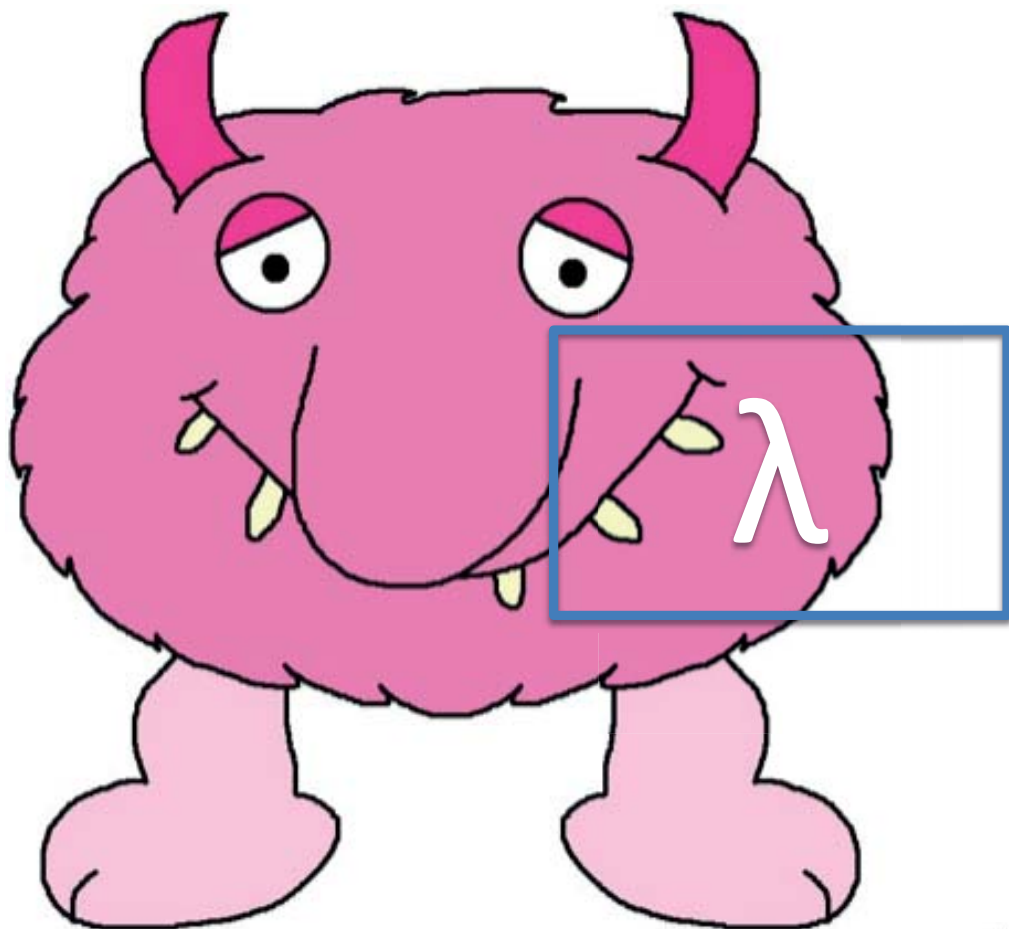
Verteilte Monte-Carlo-Simulationen mit Scheme 48



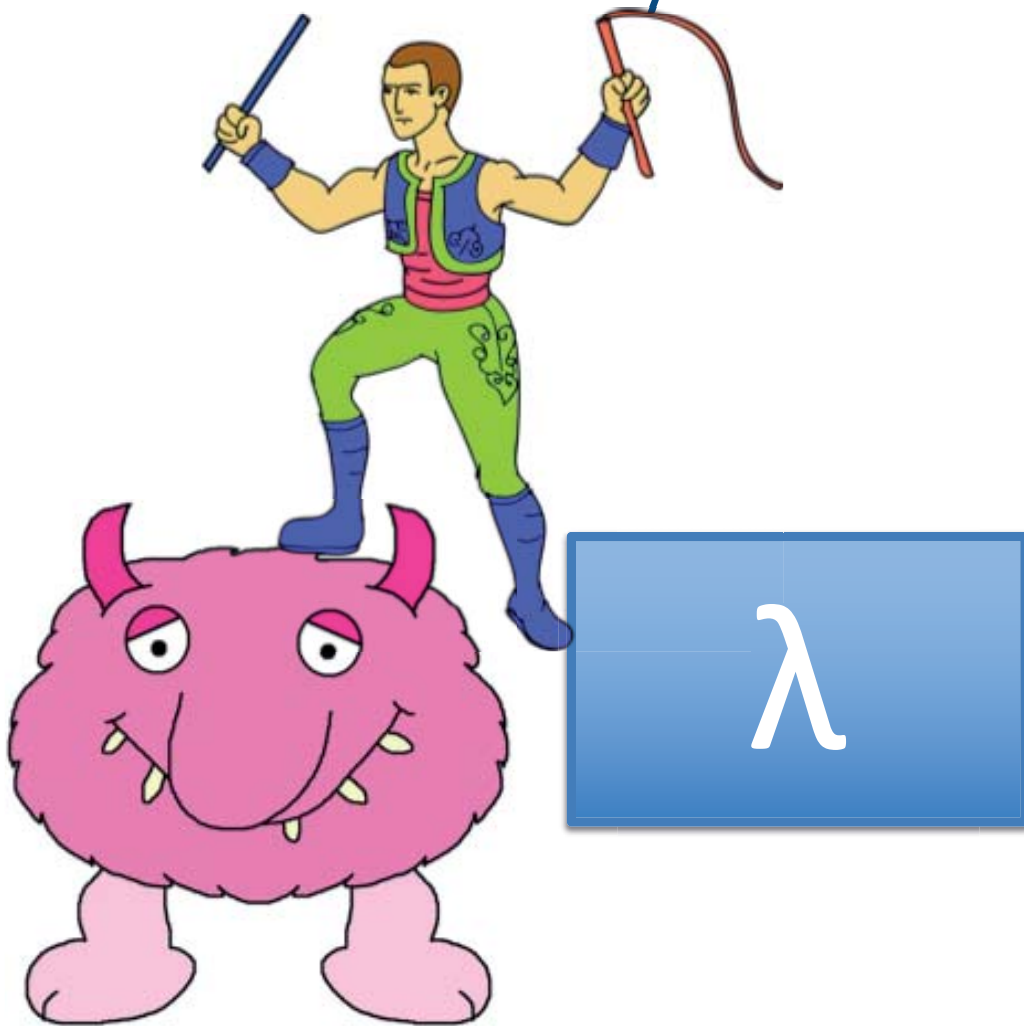
Libraries aus dem Ökosystem nutzen



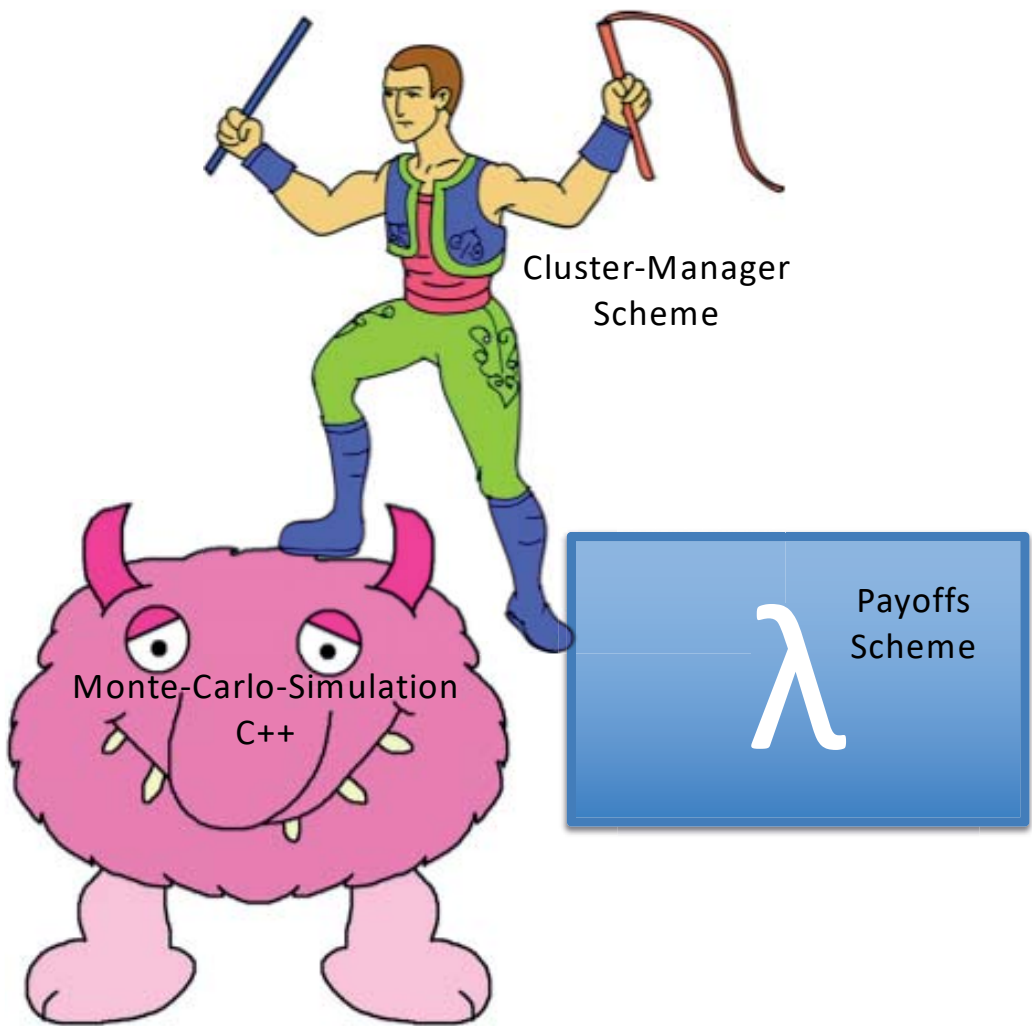
Libraries aus dem Ökosystem nutzen



Libraries aus dem Ökosystem nutzen



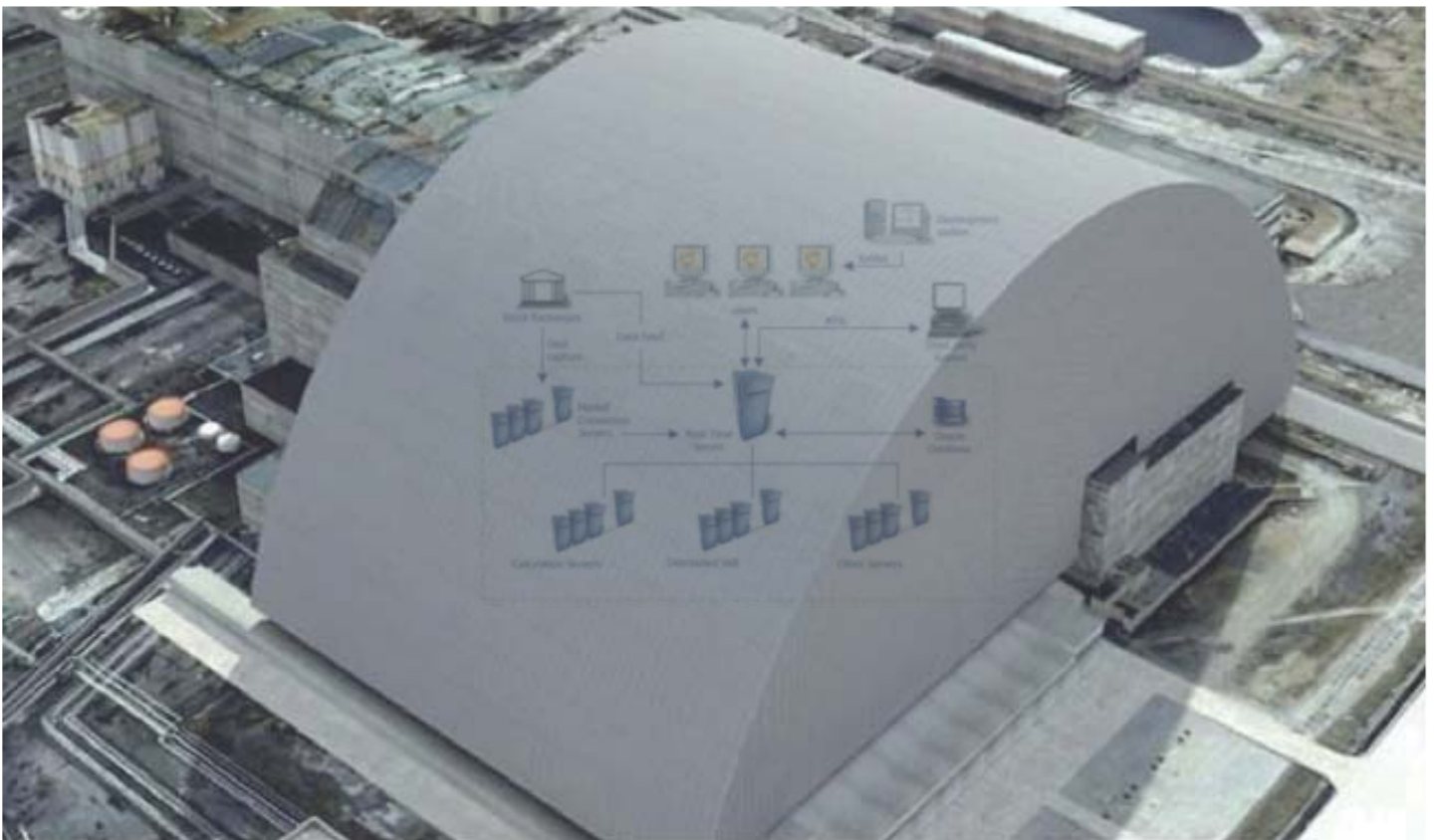
Und auch hier ...



Shell um alten Kern

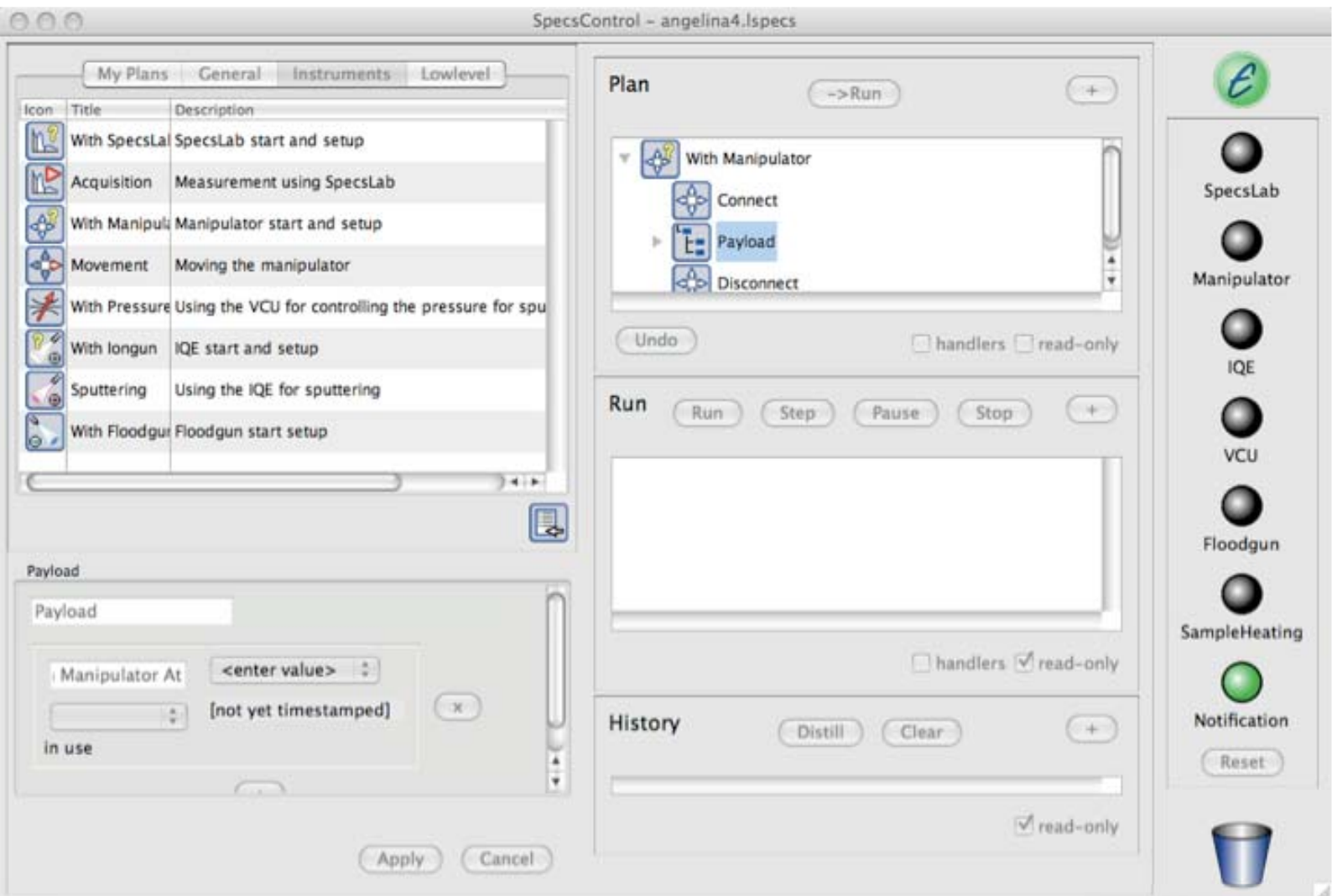


Außen hübsch ...



SpecsControl





DSL-Semantik in PLT Redex

```
specs.scm - DrRacket
specs.scm (define ...) Check Syntax Debug Macro Stepper Run Stop

(define begin? (test-match Lspecs (begin comm ...)))

(define Rspecs
  (reduction-relation
   Lspecs
   (--> (<> (begin (in-hole LE_1 (while var_1 (from) comm_1)) comm_2 ...) log_1)
        (<> (begin comm_2 ...) log_1)
        "while/end")
   (--> (<> (begin (in-hole LE_1 (while var_1 (from number_1 number_2 ...) comm_1))
                 comm_2 ...)
        log_1)
        (<> (begin (in-hole LE_1 (let var_1 number_1 comm_1))
                 (in-hole LE_1 (while var_1 (from number_2 ...) comm_1))
                 comm_2 ...)
        log_1)
        "while/1")
   (--> (<> (begin (in-hole LE_1 (let var_1 number_1 (in-hole LE_2 (in-hole CE_1 var_1)))
                 comm_2 ...)
        log_1)
        (<> (begin (in-hole LE_1 (let var_1 number_1 (in-hole LE_2 (in-hole CE_1
number_1))))
                 comm_2 ...)
        log_1)
        "var")
   (--> (<> (begin (in-hole CE_1 (+ number_1 number_2))
                 comm_2 ...)
        log_1)
        (<> (begin (in-hole CE_1 ,(+ (term number_1) (term number_2)))
                 comm_2 ...)
        log_1)
        "+")
   (--> (<> (begin (in-hole LE_1 (command string_1 number_1))
                 comm_2 ...)
        (item_1 ...))
        (begin comm_2 ...))
   ))))

Determine language from source Background expansion is disabled 137:0 268.28 MB
```

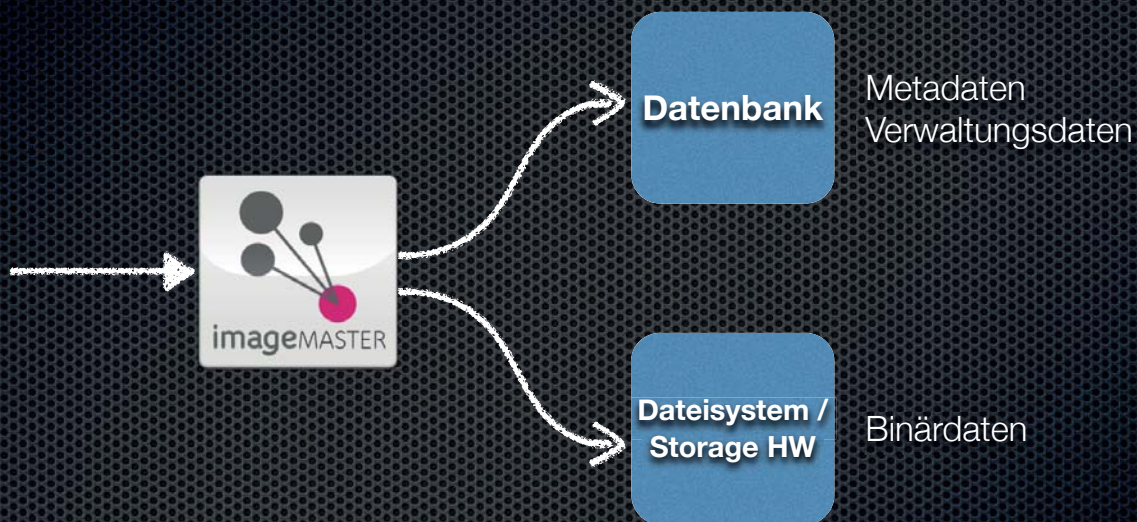
Implementierung in Java

```
public ReductionResult reduce(Environment env) {  
  
    if (this.bindings.forall  
        (new de.specs.util.Mapping<Binding, Boolean>() {  
            public Boolean apply(Binding binding) {  
                return binding.exp.isValue();  
            }  
        }  
    )) {  
  
        ISequence residual;  
        boolean failed;  
        de.specs.specscontrol.IInstruction done = null;  
        ISequence log;  
  
        if (this.body instanceof Skip)  
            return new ReductionResult(new Skip(), false);  
        else if (this.body instanceof Fail) {  
            if (this.handler instanceof Fail)  
                return new ReductionResult(new Fail(), true);  
            else if (this.handler instanceof Skip)  
                return new ReductionResult(new Fail(), true);  
            else {  
                ReductionResult handlerResult = this.handler.reduce(bind(env, this.bindings));  
                residual  
                    = new Begin(new Let(this.id, this.bindings,  
                                        this.body,  
                                        handlerResult.getResidualSequence()),  
                                new Skip());  
                failed = true;  
                done = handlerResult.getDone();  
                log = handlerResult.getHistory();  
            }  
        } else {  
            ReductionResult bodyResult  
                = this.body.reduce(bind(env, this.bindings));  
            residual  
                = new Begin(new Let(this.id, this.bindings,  
                                    bodyResult.getResidualSequence(),  
                                    this.handler),  
                            new Skip());  
            failed = bodyResult.getFailed();  
            done = bodyResult.getDone();  
            log = bodyResult.getHistory();  
        }  
    }  
  
    if (done != null) {
```

FP in Java

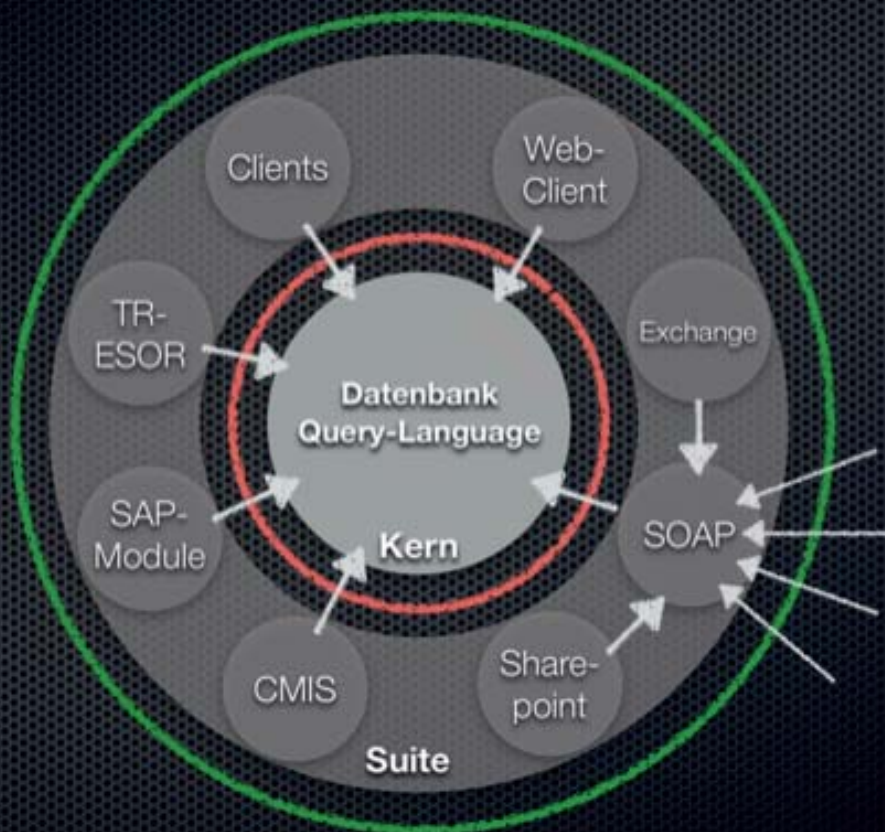


ImageMaster: Enterprise-Content-Management-Suite




- Produkt seit > 20 Jahren
- Zahlreiche Kunden, Milliarden von Dokumenten
- Ca. 200 Mitarbeiter in Entwicklung & Service
- Version 8: C++

Zentralisierung des Wissens über Datenbank und Datenmodell



5

(Folie: Eric Knauel, BOB 2015) 

Wormloch-Transformation für Galaxien

IQuery<ImaUniverse>

galaxy(batchItems)

BatchItem

Wormloch

IQuery<SqlUniverse>

```
project(["DOCUMENT_ID", ref("LHS_DOC_ID")],  
        ["BATCH_ID", ref("BATCH_ID")],
```

```
    restrict(=(ref("LHS_DOC_ID"), ref("RHS_DOC_ID"))  
    combine(product,  
            project(["LHS_DOC_ID", ref("DOCUMENT_ID")], ...,  
                    galaxy("LOCKS")  
            project(["RHS_DOC_ID", ref("DOCUMENT_ID")],  
                    restrict(not(=(bit_and(ref("TENANT_MASK"), const(1)),  
                                     const(0)))  
                    galaxy("DOCUMENTS")
```

LOCKS X DOCUMENTS

**Das Wormloch zentralisiert das Wissen
über das Datenmodell!**

18

„Es ist vermutlich der einzige Compiler, der auch
Dokumente archivieren kann.“

– *Unbekannt*

Konzeption

```
(define (query-scheme* q env fail)

  (define (to-env scheme)
    (compose-environments (rel-scheme->environment scheme)
                          env))

  (let recur ((q q))
    (cond
      ((empty? q) the-empty-rel-scheme)
      ((base-relation? q) (base-relation-scheme q))
      ((project? q)
       (let ((base-scheme (recur (project-query q)))
             (alist (project-alist q)))
         (if fail
             (for-each (lambda (p)
                         (if (aggregate? (cdr p))
                             (fail "non-aggregate" (cdr p))))
                       alist))
         (make-rel-scheme
          (map (lambda (p)
                 (let ((type (expression-type* (to-env base-scheme) (cdr p))))
```

Implementierung

```
@Override
public List<List<Object>> visitProject(Scheme<U> scheme,
                                       List<Assoc<U>> associations,
                                       IQuery<U> query) {
    List<List<Object>> query_result = query.visitFrom(this);
    Scheme<U> query_scheme = query.getScheme();
    ArrayList<List<Object>> result = new ArrayList<List<Object>>();
    for (List<Object> row : query_result) {
        List<Object> row_result = new ArrayList<Object>();
        for (Assoc<U> assoc : associations) {
            Object expr_result =
                evaluate(assoc.expr,
                        new Environment<U>(query_scheme, row));
            row_result.add(expr_result);
        }

        result.add(row_result);
    }
    return result;
}
```

Technik

- alles neu
- tiefe Integration, schrittweise ersetzen
- Dompteur zwischen OOP und FP
- Schale um OO-Kern
- Entwurf in FP, Implementierung OOP



*Die Grenzen
meiner
Sprache
bedeuten die
Grenzen
meiner Welt.*

FP ist ganz toll ...

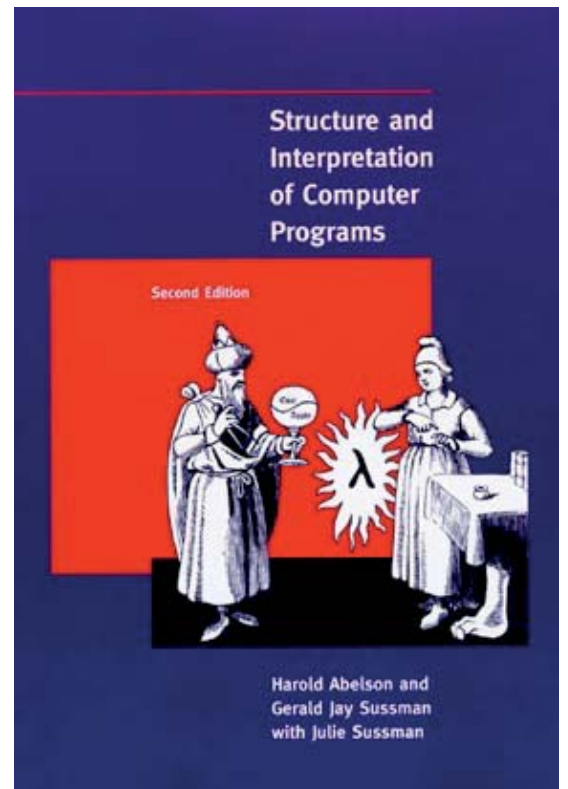


FP ist ganz leicht ...



„FP ist super für die Ausbildung“

- Syntax
- Größe
- funktional
- einfacher Übergang zu X



Using Scala as a “better Java”

By Alvin Alexander. Last updated: September 10 2016

A lot of people want to sell Scala as a functional programming (FP) language. That's fine, it certainly is, and FP has many benefits. But when I first started working with Scala I was just looking for a "better Java".

Personally, I grew tired of Java's verbosity. Ruby showed us a different way, and I loved it, but ... after a while I felt like many of my unit tests in Ruby wouldn't be necessary if the language had static types. I wished for a concise Ruby-like language with static types

Table of Contents

- [Examples](#)
- [A class to launch your application](#)
- [Types](#)
- [Collections and loops](#)
- [Defining a class](#)
- [Method to double an int](#)
- [Method to double every element in a collection](#)
- [Loops and filters](#)

Program by Design

Program by Design is an innovative project for computing education that combines motivation with principles. On the surface we use engaging contexts—our “hello world” program is an animation, and students have the opportunity to program games, mashups, phones, etc.—while teaching a principled and scalable approach to computing. We have spent over fifteen years developing this curriculum, and have offerings at the middle-school, high-school and university levels (roughly, ages 10 and above). Our material is even used for in-house corporate training. As we discuss [in more depth](#), we set out to address fundamental problems that students face. This has led to a series of innovations in programming environments, programming methodology, and programming languages.

Our widely-attended workshop series has been suspended because we now invest our energy in our sibling project, [Bootstrap](#), which is designed for younger students (starting around age 12). That material is a perfect setup for ours. Meanwhile, you can still get access to all our [materials](#), which remain in constant development.

Our Project comprises faculty and graduate students at several universities including Adelphi, Brown, Cal Poly, Northeastern, Northwestern, Utah and WPI, with support from several other universities as well as industrial organizations.

The Project is primarily funded by the US National Science Foundation and Google.

Overview What the Project is about

Events Our free workshops

Materials Books, software, etc.

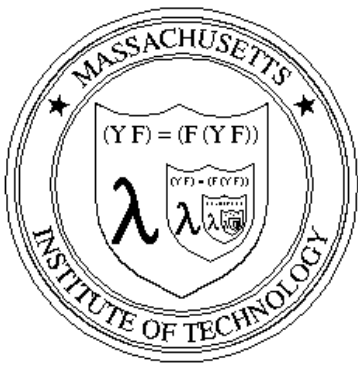
Dual Credit How high-school students can earn college credit

Who The people behind the project

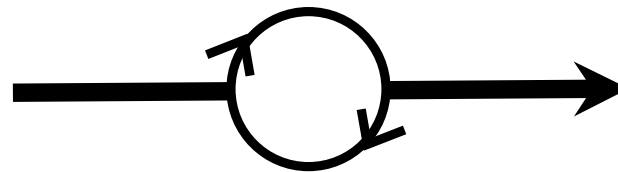
Sponsors Thanks to the funding sources who keep this project active!

Lehrsprachen

beobachten
verbessern
von vorn



willkürlicher Startpunkt



Lehrsprachen



Lehr-Programmierungsumgebung

```
(define inc
  (lambda (x)
    (set! x (+ x 1))
    x))
```

Welcome to [DrRacket](#), version 5.1.3 [3m].
Language: Die Macht der Abstraktion; memory limit: 128 MB.
lambda: Lambda-Abstraktion hat mehr als einen Ausdruck als Rumpf
>

Die Macht der Abstraktion ▾ 2:2

Material



DeinProgramm ist ein Projekt zur Anfängerausbildung im Programmieren, das seit 1999 an der Universität Tübingen und anderswo entwickelt wird. Die entstandenen didaktischen Konzepte, Software und Materialien wurden in zahlreichen Anfängerausbildungen erprobt und kontinuierlich verbessert.

- FP besser als OO
- viele Wege für existierende Projekte
- langfristig große Veränderung
- erfordert ~~gute~~ richtige Ausbildung

