Concepts of Functional Programming

Peter Kofler, ‘Code Cop’
JavaAbend, October 2010
... for Java Brain\textsuperscript{(dead)s}

UML

OOA

OOP

SOAP
Peter Kofler

- dev since 11 years
- “fanatic about code quality”
- “Code Cop”
f(x)
History

- 193x Lambda
- 1958 LISP
- 197x Scheme
- 198x Haskell
- 199x Common Lisp
- 200x Clojure
Ideas...

Think different.
evaluation of mathematical functions
def f(x) x+2 end

def f(x:Int) = x+2

public int f(int x) {
    return x+2;
}

</code>
Lambda Calculus
\[ \lambda x. \ x+2 \]

\[ f(x) := x+2 \]
Definition $f(x)$

$\lambda x. x + 2$

application $f(3)$

$(\lambda x. x + 2) 3$
No Side Effects!
no side effects
pure
Immutable Data
final is the new private

(Joshua Bloch, Effective Java)
avoids state and mutable data
RECURSION
It recurs.

RECURSION
It recurs.

RECURSION
It recurs.

RECURSION
It recurs.

RECURSION
It recurs.

RECURSION
It recurs.

RECURSION
It recurs.

RECURSION
It recurs.
\[ n! := n \times (n-1)! \]
\[ 0! := 1 \]
\[ f(n) := n \times f(n-1) \]
\[ f(0) := 1 \]
def fact(n: Int): Int =
    if (n === 0) 1 else n * fact(n-1)
Higher Order
\[
\frac{dx}{x} f(x) \rightarrow f'(x)
\]
Anonymous Functions
array = [3, 5, 7, 9]
array.find do |v|
  v*v > 30
end

<ruby>
Closures
function derivative(f, dx) {
    return function(x) {
        return (f(x+dx)-f(x))/dx;
    };
}
first-class & anonymous functions
Currying
\[ f(x,y) := x+y \rightarrow \]
\[ \text{def } f(x:\text{Int}, y:\text{Int}) = x+y \]

\[ f_1(x) \text{ returns } f_2(y) \rightarrow \]
\[ \text{def } f_1(x:\text{Int}) = (i:\text{Int}) \Rightarrow x+i \]
\[ \text{def } f_2(y) = X_{\text{now constant}} +y \]
“normal”
\( \lambda x \ y. \ x + y \)
curried
\( \lambda x. \ \lambda y. \ x + y \)
Evaluation

strict

lazy
length([2+1, 1/0])

→ Error

→ "2"
everything is a function
def mywhile(cond: =>Boolean) (body: =>Unit) {
    if (cond) {
        body; mywhile(cond)(body)
    }
}

... mywhile( i > 0 ) { i -= 1 }
The almighty List

by Jonas Bonér
Iterating
(foeach/each)
Folding (foldLeft/inject)
Reducing (reduceLeft)
def factorial(n: Int) =
(((1 to n) \ 1) ( _ * _ ))
Mapping (map/collect)
Binding (flatMap)
a = [ "a", "b", "c"]
a.collect { |x| x+"!" }  
=> # ["a!", "b!", "c!"]
def list(folder: File) = {
  folder.listFiles.
    filter(_.isDirectory).
    flatMap(list(_)) ++
  folder.listFiles.
    filter(_.isFile)
}

</scala>
Infinite Lists
Continuations
\[(\ast \ (\ + \ 1 \ 2) \ 3)\]  
\[=\]  
\[(+ \ 1 \ 2)\]  
with continuation  
\[(\ast \ [\] \ 3)\]
def loop(interrupt)
    for i in 1..10
        puts "Value of i: #{i}"
        if i == interrupt
            callcc{|c| return c}
        end
    end
end
end

</ruby by Bruce Tate>
irb(main):007> cont = loop 5
Value of i: 1
...
Value of i: 5
=> #<Continuation:0x2b5a358>
irb(main):008> cont.call
Value of i: 6
...
Value of i: 10
Uhhh ?!
Monads
Skipped...

- Monads
- Category Theory
- Pattern Matching
- Algebraic Data Types
- Type Inference
- Hindley Miller
- ...

56
Thank You
Acknowledgements

• This presentation was supported by System One (as Research Day)
  (http://www.systemone.net/en/) ... 20%

• This presentation was supported by sIT Solutions (for Developer Round Table)
  (http://www.s-itsolutions.at/) ... 50%
www.code-cop.org
Links

- http://www.defmacro.org/ramblings/fp.html
CC Images #1

- zombie: http://www.flickr.com/photos/vincetemplement/333325657/
- spray face: http://www.flickr.com/photos/iangallagher/4115047191/
- other worlds: http://www.flickr.com/photos/tohoscope/43818252/
- new ideas: http://www.flickr.com/photos/jpovey/2051196149/
- history: http://www.flickr.com/photos/shadowgate/2679760160/
- functions: http://www.flickr.com/photos/stefz/2159280574/
- lambda: http://www.flickr.com/photos/itsgreg/419031515/
- effects: http://www.flickr.com/photos/delgrossodotcom/3094902951/
- pure: http://www.flickr.com/photos/10451396@N00/429388973/
- dices: http://www.flickr.com/photos/dicemanic/19743895/
- lock: http://www.flickr.com/photos/bpc009/3328427457/
- states: http://www.flickr.com/photos/krazydad/2986774792/
- recursion: http://www.flickr.com/photos/torley/2361164281/
CC Images #2

- higher: http://www.flickr.com/photos/brent_nashville/2204427649/
- unknown: http://www.flickr.com/photos/laughingsquid/2443128847/
- closed: http://www.flickr.com/photos/functoruser/244208110/
- currying: http://www.flickr.com/photos/dumbeast/315869395/
- strict: http://www.flickr.com/photos/williac/99551756/
- lazy: http://www.flickr.com/photos/ucumari/2813269535/
- möbius: http://www.flickr.com/photos/serafa/2590342868/
- uhh: http://www.flickr.com/photos/jswaby/1178547691/
- seaside: http://www.flickr.com/photos/rgtmum/2280206308/
- monads: http://www.flickr.com/photos/adamrice/3266888223/
- questions: http://www.flickr.com/photos/seandreilinger/2326448445/