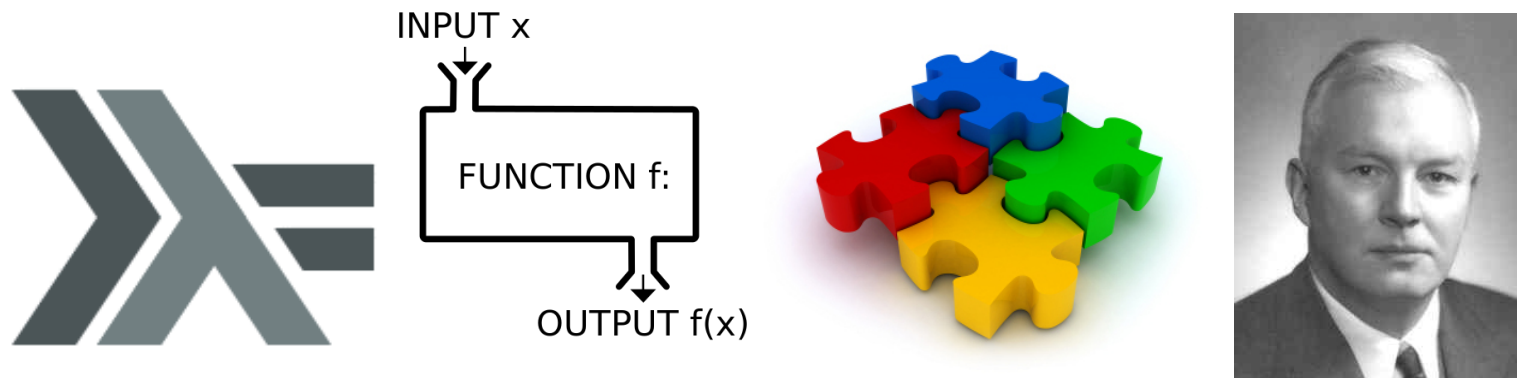


Functional Programming



Overview

Christoph Denzler / Daniel Kröni
University of Applied Sciences Northwestern Switzerland
Institute for Mobile and Distributed Systems

Functional Programming [fprog 1la]

Lecturer: Daniel Kröni daniel.kroeni@fhnw.ch

Time: 12:15-13:00 / 13:15-14:00 / 14:15-15:00

Room: 6.1H13

Office Hours: Tuesday: 15:15-16:15

Room: 5.2B16

Course Material: AD *E1862_Unterrichte_I/E1862_1la/fprog*

Examinations: 03.11.2015 12:15 - 13:15 5.0H02
12.01.2016 12:15 - 13:15 6.0D13

Functional Programming [fprog 1lb]

Lecturer: Edgar Lederer edgar.lederer@fhnw.ch

Time: 15:15-16:00 / 16:15-17:00 / 17:15-18:00

Room: 1.313

Course Material: AD *E1862_Unterrichte_I/E1862_1lb/fprog*

Examinations: 02.11.2015 15:15 - 16:15 5.0H06
11.01.2015 15:15 - 16:15 6.0D13

Functional Programming [fprog 1lbb]

Lecturer: Daniel Kröni daniel.kroeni@fhnw.ch

Time: 15:15-16:00 / 16:15-17:00 / 17:15-18:00

Room: 5.0H02

Office Hours: Tuesday: 15:15-16:15

Room: 5.2B16

Course Material: *AD E1862_Unterrichte_I/E1862_1lbb/fprog*

Examinations: 02.11.2015 15:15 - 16:15 5.0H06
11.01.2015 15:15 - 16:15 6.-1D13



A Tiny Book Library in Java (I)

```
public class Book {  
    private final String author, title;  
  
    public Book(String author, String title) {  
        this.author = author; this.title = title;  
    }  
  
    public String getAuthor() { return author; }  
  
    public String getTitle() { return title; }  
  
    public String toString() {  
        return "Book(" + author + "," + title + ")";  
    }  
}
```



A Tiny Book Library in Java (II)

```
public class JavaLibrary {  
    public final List<Book> shelve = new LinkedList<Book>();  
  
    public void addBook(Book book) {  
        shelve.add(book);  
    }  
  
    public List<Book> booksByAuthor(String author) {  
        List<Book> result = new LinkedList<>();  
        for (Book b : shelve) {  
            if (b.getAuthor().equals(author))  
                result.add(b);  
        }  
        return result;  
    }  
    ...  
}
```



A Tiny Book Library in Java (III)

...

```
public static void main(String[] args) {  
    JavaLibrary lib = new JavaLibrary();  
    lib.add(new Book("A", "A1")); lib.add(new Book("A", "A2"));  
    lib.add(new Book("B", "B1"));  
  
    List<Book> booksByA = lib.booksByAuthor("A");  
    Collections.sort(booksByA, new Comparator<Book>() {  
        public int compare(Book b1, Book b2) {  
            return b1.getTitle().compareTo(b2.getTitle());  
        }  
    });  
    System.out.print(booksByA);  
}
```



A Tiny Book Library in Haskell

```
import GHC.Exts (sortWith)

data Book = Book { author :: String, title :: String } deriving Show

shelve :: [Book]
shelve = [Book "A" "A1", Book "A" "A2", Book "B" "B1"]

booksByAuthor :: String -> [Book] -> [Book]
booksByAuthor name books = filter (\b -> author b == name) books

sortByTitle :: [Book] -> [Book]
sortByTitle books = sortWith title books

main :: IO ()
main = putStr (show booksByA)
  where booksByA = sortByTitle (booksByAuthor "A" shelve)
```



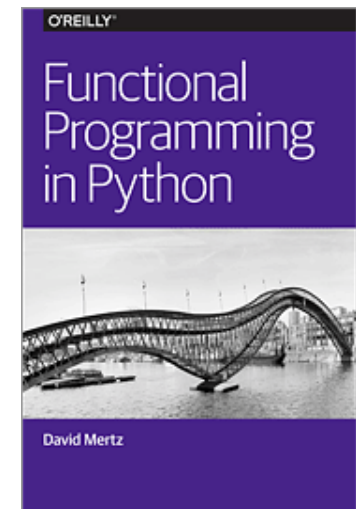
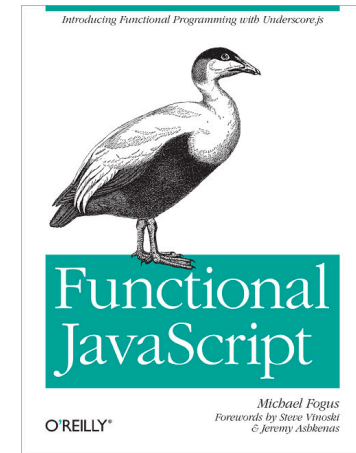
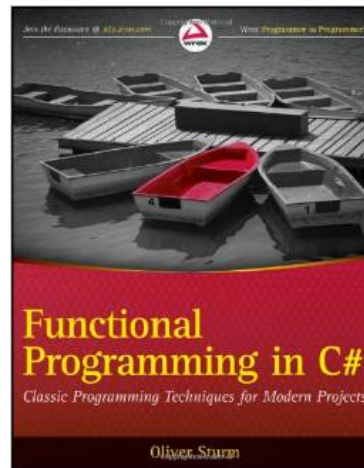
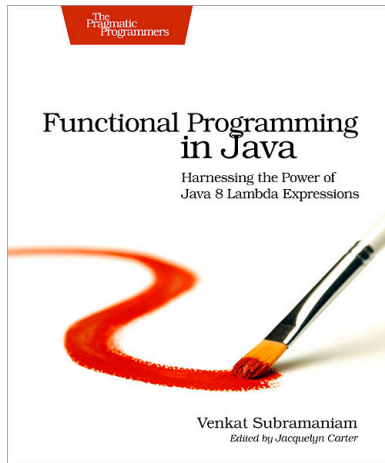

Same in Java 8

```
public class JavaLibrary {  
    public static final List<Book> books =  
        Arrays.asList(new Book("A", "A1"), new Book("A", "A2"), new Book("B", "B1"));  
  
    public static List<Book> booksByAuthor(String author) {  
        return books.stream()  
            .filter((b) -> b.getAuthor().equals(author))  
            .collect(Collectors.toList());  
    }  
  
    public static void main(String[] args) {  
        List<Book> booksByA = booksByAuthor("A").stream()  
            .sorted((b1, b2) -> b1.getAuthor().compareTo(b2.getAuthor()))  
            .collect(Collectors.toList());  
        System.out.print(booksByA);  
    }  
}
```

Why Functional Programming

FP	OOP
Declarative (what to do) Combining expressions	Imperative (how to do it) Step-by-step statements
Data Functions	Classes Interfaces Inheritance Hierarchies Methods Objects (this, super) Dynamic Dispatch
Evaluating expressions	Changing state
Reduce movable parts (things do no change)	Encapsulate movable parts (you can't see what changes)
Simple: Big advantage when building complex systems.	Complex: Seldom the right choice to build a system.

Many Languages Adopt Functional Ideas



Why Haskell?

- Because Haskell is
 - Designed from the ground up to be a purely functional language
 - No compromise / no cheating allowed
 - Statically typed (the compiler is your friend)
 - One of the coolest languages by now
- But what you learn in this class is applicable to:
 - JavaScript
 - Java, C# (and other languages which adopt functional concepts)
 - F# (Microsoft's functional language)
 - Erlang (Drives Whatsapp with 600M users)
 - Scala (FP-OO Hybrid on JVM, used heavily by Twitter)
 - Clojure (Lisp on the JVM)

Goals

- **The Students**
 - ... learn a new way of thinking about programming
 - ... appreciate the clear and concise style of functional programs
 - ... can write functional programs in Haskell
 - ... can apply functional concepts to other programming languages such as Java or C#

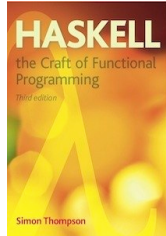
References

- **Books**



Graham Hutton

<http://www.cs.nott.ac.uk/~gmh/book.html>



Simon Thompson

<http://www.haskellcraft.com/craft3e/Home.html>



Miran Lipovača

<http://learnyouahaskell.com/>

Schedule

Week	Date Mo.	Date Tue.	Topic
1	14.09.15	15.09.15	Introduction / Motivation
2	21.09.15	22.09.15	Types
3	28.09.15	29.09.15	Lists
4	05.10.15	06.10.15	Defining Functions
5	12.10.15	13.10.15	More Functions
6	19.10.15	20.10.15	Recursive Functions
7	26.10.15	27.10.15	Lab
8	02.11.15	03.11.15	Midterm Exam
9	09.11.15	10.11.15	Algebraic Datatypes
10	16.11.15	17.11.15	Typeclasses
	23.11.15	24.11.15	Projektwoche
11	30.11.15	01.12.15	Higherorder Functions
12	07.12.15	08.12.15	IO
13	14.12.15	15.12.15	Case Study
	21.12.15	22.12.15	Christmas
	28.12.15	29.12.15	Holidays
14	04.01.16	05.01.16	Functors and Monads
15	11.01.16	12.01.16	Final Exam

Facebook's New Spam-Killer Hints at the Future of Coding

LOUIS BRANDY PAUSES before answering, needing some extra time to choose his words. “I’m going to get in so much trouble,” he says. The question, you see, touches on an eternally controversial topic: the future of computer programming languages.

Brandy is a software engineer at Facebook, and alongside a team of other Facebookers, he spent the last two years rebuilding the system that removes spam—malicious, offensive, or otherwise unwanted messages—from the world’s largest social network. That’s no small task—Facebook juggles messages from more than 1.5 billion people worldwide—and to tackle the problem, Brandy and team made an unusual choice: they used a programming language called Haskell.

<http://www.wired.com/2015/09/facebooks-new-anti-spam-system-hints-future-coding/>

Want to get rich?

★ What kind of jobs do software engineers who earn \$500k per year do?

Recently I read an article about a programmer who gets paid \$3 million / year at Google :

www.businessinsider.com/a-google-programmer-blew-off-a-500000-salary-at-startup--because-hes-already-making-3-million-every-year-2014-1 ↗

What kind of tasks (or skills maybe) make them so valuable?

Follow Question

1.7k

Comment

1

Share

41

Downvote

...

<http://www.quora.com/What-kind-of-jobs-do-software-engineers-who-earn-500k-per-year-do>

Learn Haskell!



Michael O. Church

322 upvotes by Jeff Nelson (Invented Chromebook, Former Googler), Si Yin, Justin Chu, (more)

First, \$500,000 per year and \$3 million require very different strategies.

Decent financial quants get \$500,000 per year, including bonus. You have to be strong, but you don't need to be a "rock star". I know smart people you've never heard of who are earning \$300k-1.5M as quants. Unlike in the Valley, where you need to be a 10x self-seller and put up with VCs and become a celebrity to make any money in code, you can get moderately rich just by being a good programmer in finance. (Some people say that programmers are second-class citizens in finance, compared to traders banking millions, and that's true; but they're *third*-class citizens compared to founders and VCs in the Valley.) As a pure programmer, you need to find a niche that finance demands (e.g. C++, low-latency programming, and perhaps Haskell in a few years). Or you can study the mathematics and become a quant. This shouldn't

More Motivation

jaxenter

FOLLOW US



SEARCH

GO!

NEWS

VIDEOS

BOOKS

JAX MAGAZINE

HOME » NEWS » JAVA

EDITORS PICK

ECLIPSE

ANDROID

ARCHITECTURE

CLOUD

AGILE

WEB

SERIES



Waiting on nine

Java 9 features announced!

New APIs, performance features and modular source code are the first features to be confirmed for Oracle's next major Java release



Functional programming

5 reasons you should care about functional programming

Even if you don't think you need to care about functional programming, you should. There are plenty of reasons for Javaheads to know more about them: W-Jax speaker Neal Ford...

jaxmagazine

Targeting the future
Inside Open API

▼ Download Free PDF Now!



VIDEOS