

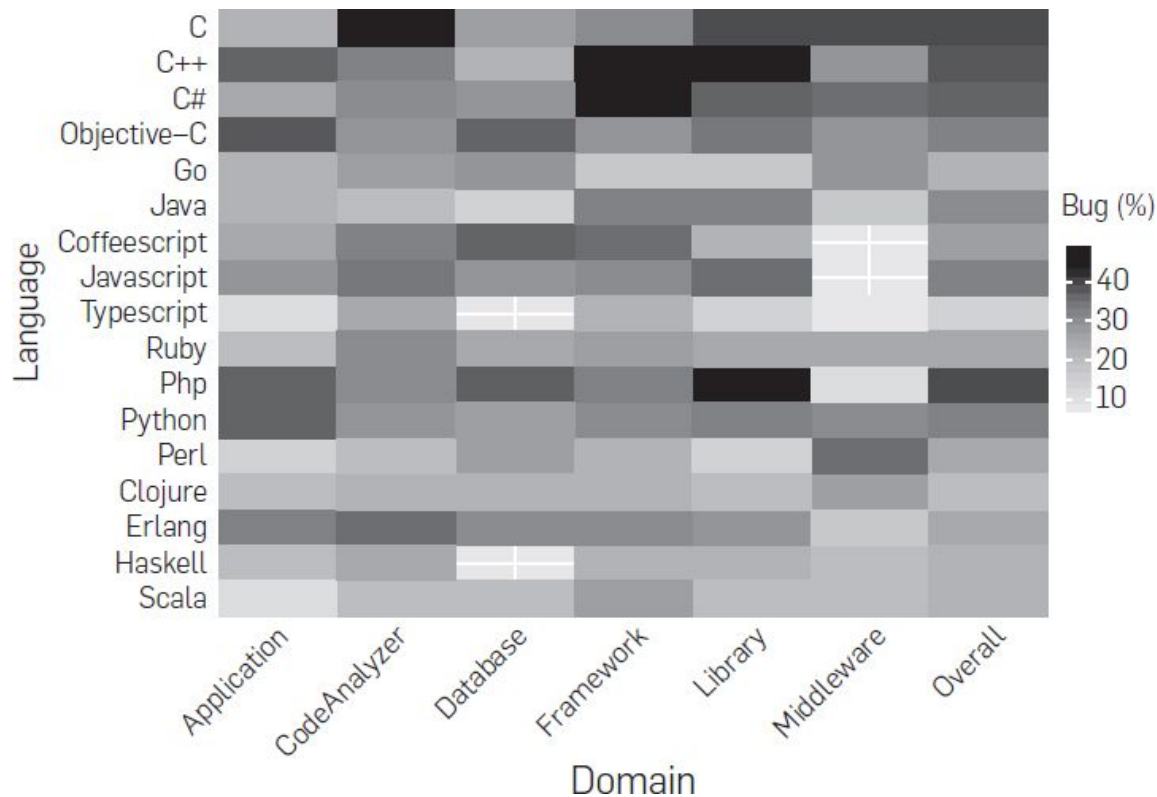
# Programação Funcional em Java

Daniela Morais

**\$ whoami**

Entusiasta de Java e de códigos “limpos”  
Ativista de software livre

# A Large-Scale Study of Programming Languages and Code Quality in Github



# java 8

JSR 335: Versão final em 2014

Releases de Java a cada 6 meses

Scala, Kotlin e Clojure em alta nos últimos anos



# conceitos

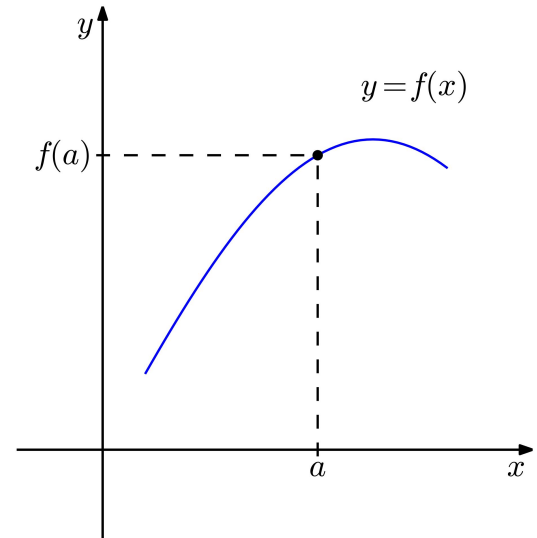
funções puras

imutável

funções anônimas

# funções puras

uma função liga um elemento do domínio com um segundo conjunto, o contradomínio de tal forma que a cada elemento do domínio está associado exatamente a um, e somente um, elemento do contradomínio.



# funções puras

Exemplo

# imutável

tendência em criar novas funções

transformar em novos objetos

mais fácil de testar



# funções anônimas

lambdas

código mais limpo e menos verbosidade

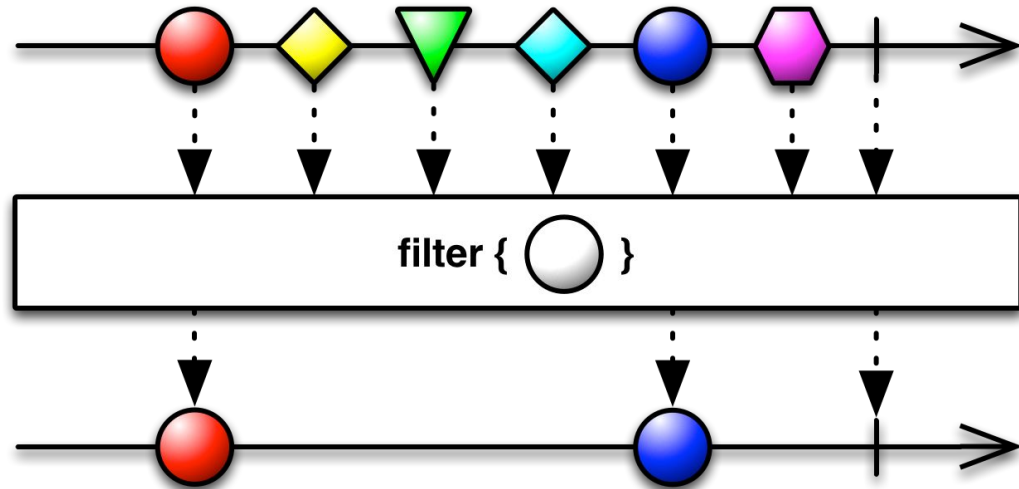
# funções anônimas

Exemplo

# streams, filters

estruturas básicas

realizam “transformações”



# streams, filters

Exemplo

# method reference

Exemplo

# map reduce

transformar entradas em outro tipo na saída

# map reduce

Exemplo

# map, filter, and reduce explained with emoji 🤔

map([🐮, 🍠, 🐔, 🌽], cook)  
=> [🍔, 🍟, 🍗, 🍿]

filter([🍔, 🍟, 🍗, 🍿], isVegetarian)  
=> [🍟, 🍿]

reduce([🍔, 🍟, 🍗, 🍿], eat)  
=> 🤑



# java 9

Novos incrementos

Criação de Collections imutáveis

*“Java 9: A versão que ninguém usou”*

# takeWhile / dropWhile

Exemplo

# duck typing

possibilitou a implementação

invokeDynamic de Java 7

gerar bytecode em runtime



# dicas

code style

quebra de linha depois de `stream()` e a cada operação

evitar funções anônimas com mais de 3 linhas

## that's all

*“A economia do século 21 não é sustentada por aço.  
A economia do século 21 é sustentada por software.*

*Software é elemento tão fundamental para o  
desenvolvimento econômico no século 21 quanto o foi a  
produção de aço no século 20. (...)*

*E a boa notícia é que ninguém o possui.”*

Software e Comunidade no começo do Século 21