

# Functional programming style

in Groovy, Java 8, and other languages

Michael van Vliet

Freek de Bruijn

DTL Programmers Meeting

December 12th 2014

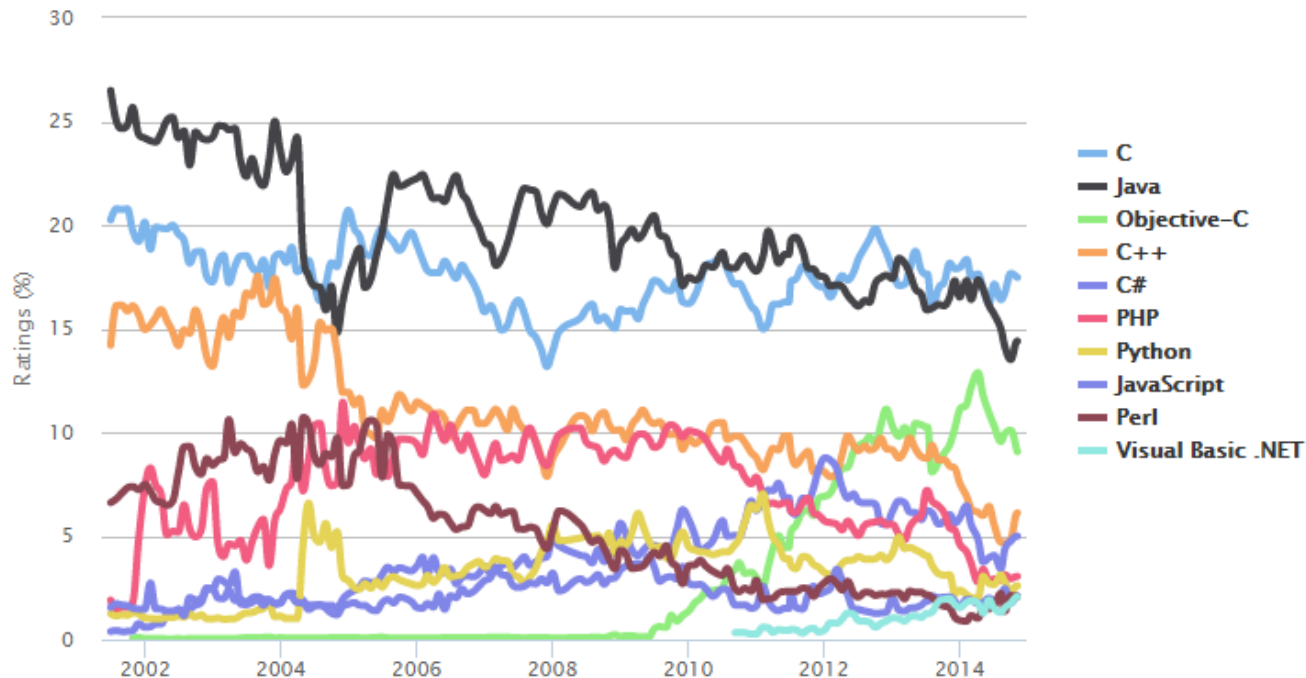
# Overview

- First 5 minutes
  - short Java history
  - functional programming
- Hands on
  - introduction
  - movie queries

# Language popularity?!?

TIOBE Programming Community Index

Source: [www.tiobe.com](http://www.tiobe.com)



# Java versions with main features

- 0: first version (was called Oak)
- 1: Java Database Connectivity
- 2: **collections framework**, Swing
- 3: Java sound ;-)
- 4: new I/O, logging API, regular expressions, XML/XSLT
- 5: **generics**, for each, annotations, autoboxing, varargs
- 6: (performance improvements)
- 7: diamond operator <>; no lambda functions & modules
- 8: **lambda functions, streams, pipelines, default methods**

# Java versions with dates

- 0: JDK 1.0 in January 1996
- 1: JDK 1.1 in February 1997 (13 months)
- 2: **J2SE 1.2** in December 1998 (22 months)
- 3: J2SE 1.3 in May 2000 (17 months)
- 4: J2SE 1.4 in February 2002 (21 months)
- 5: **J2SE 5.0** in September 2004 (31 months)
- 6: Java SE 6 in December 2006 (27 months)
- ----- **Sun acquired by Oracle** in January 2010 -----
- 7: Java SE 7 in July 2011 (55 months)
- 8: **Java SE 8** in March 2014 (32 months)

# Groovy

- **dynamic language for JVM**
- Java + **power features** inspired by (Python, Ruby and Smalltalk)
- **almost-zero learning curve (for Java developers)**
- optional **statically type check / compile**
- **Domain-Specific Languages -> easy to read and maintain**
- **reducing boilerplate/scaffolding code**
- **simplifies testing**, unit testing and mocking out-of-the-box
- seamless **integration with Java classes and libraries**
- compiled Groovy === Java bytecode

# Challenges: get ready

1. Copy/download **movies-mpaa.txt** to local machine ()  
([introcs.cs.princeton.edu/java/data/movies-mpaa.txt](http://introcs.cs.princeton.edu/java/data/movies-mpaa.txt))
2. Open IDE/editor to start solving some challenges
3. Example solutions available in Java 7/8 and Groovy
4. Last 15 minutes we will discuss solutions

# Challenges

1. How many movies are listed in movies-mpaa.txt?
2. How many unique movie titles are there?
3. Show movies with actor 'Clint Eastwood'.
4. Year(s) without movies?
5. Show number of movies by year.
6. Most active actor?
7. Most active actor by year?

Please note: movies-mpaa.txt contains movies from 1916 till 2004, limit all results within that time frame!



# References

Here are some references we used and/or that have additional information:

- [blog.paumard.org/en](http://blog.paumard.org/en)
- [en.wikipedia.org/wiki/Functional\\_programming](http://en.wikipedia.org/wiki/Functional_programming)
- [en.wikipedia.org/wiki/Imperative\\_programming](http://en.wikipedia.org/wiki/Imperative_programming)
- [blog.paumard.org/en](http://blog.paumard.org/en)
- [Functional Programming in Java - Harnessing the Power of Java 8 Lambda Expressions](#)
- [en.wikipedia.org/wiki/Java\\_version\\_history](http://en.wikipedia.org/wiki/Java_version_history)
- [programmers.stackexchange.com/questions/193630/summary-of-differences-between-java-versions](http://programmers.stackexchange.com/questions/193630/summary-of-differences-between-java-versions)
- [javapapers.com/core-java/java-features-and-history](http://javapapers.com/core-java/java-features-and-history)