Adopting Java

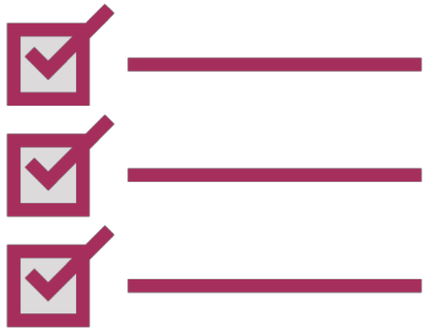
Sander Mak
FELLOW & SOFTWARE ARCHITECT
@Sander_Mak

Overview

Philosophy

When & why

Comparison



Philosophy of Java

Write

Once

Run

Anywhere

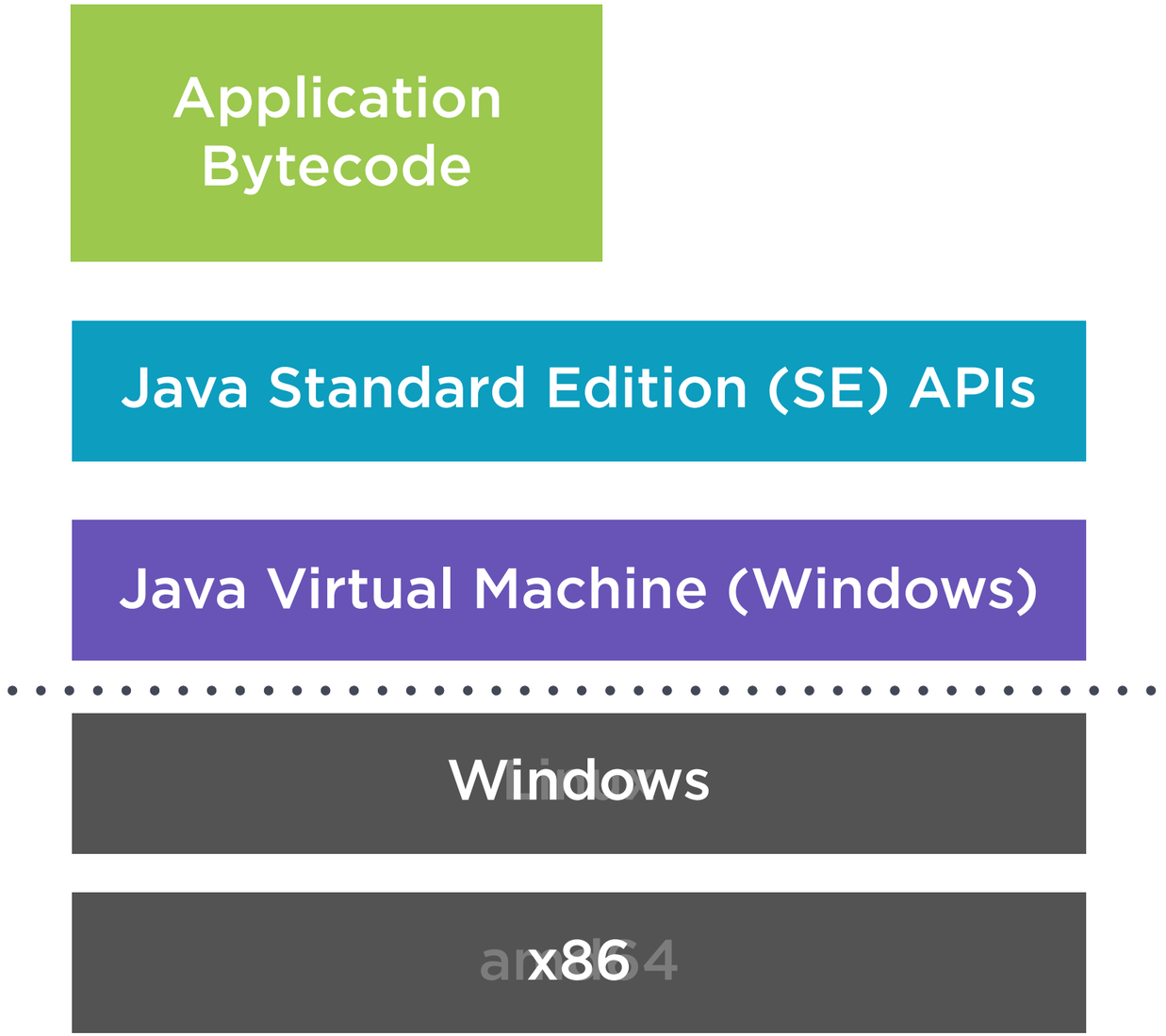
Portability

Portability

JVM for each OS/architecture

Bytecode is portable

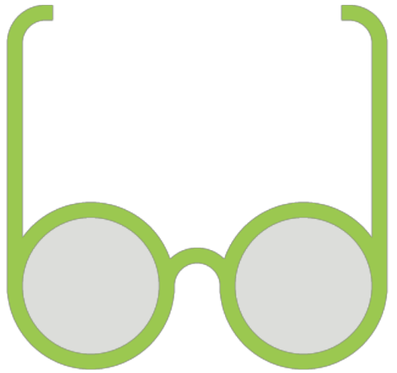
Java SE APIs platform-agnostic



Optimized for Readability

Reading code is more
important that **writing** code

Understand why to express ideas over code



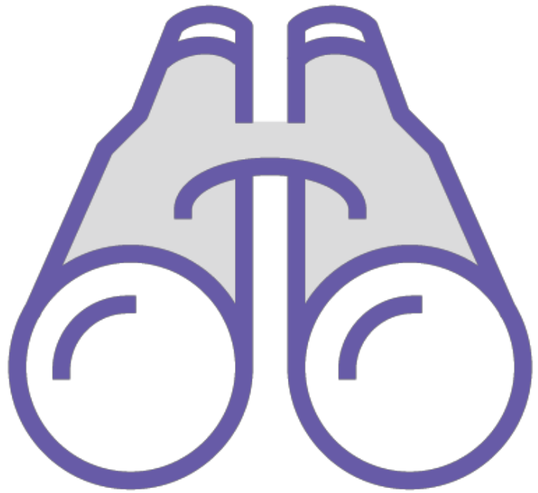
Conservative: New Features

Planning for the next 20 years of Java

'First do no harm'

Developer productivy

Maintaining simplicity



Conservative: Backward Compatibility

Existing code on new JVMs

Controlled deprecation

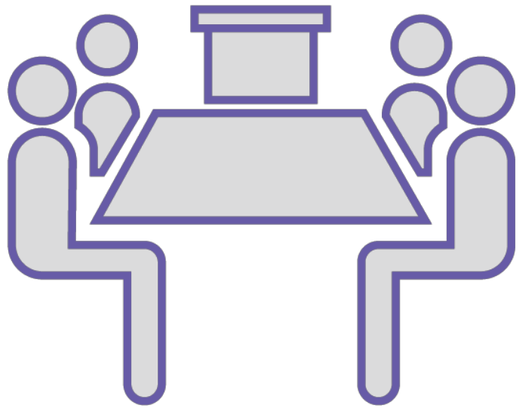
Maintaining simplicity

Open: Specification Process

Java Community Process (JCP)

Specifies the platform

Vendor & community participation



Many non-Oracle implementations: IBM, Eclipse

Open: Open-Source

OpenJDK project

GPL 2 licensed

Experimental subprojects

openjdk.java.net

Workshop
OpenJDK FAQ
Installing
Contributing
Sponsoring
Developers' Guide

Mailing lists
IRC · Wiki

Bylaws · Census
Legal

JEP Process

search

Source code
Mercurial
Bundles (6)

Groups
(overview)
2D Graphics
Adoption
AWT
Build
Compatibility & Specification Review
Compiler
Conformance
Core Libraries
Governing Board
HotSpot
Internationalization
JMX
Members
Networking
NetBeans Projects
Porters
Quality
Security
Serviceability
Sound
Swing
Vulnerability
Web

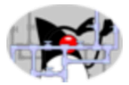
Projects
(overview)
Amber
Annotations Pipeline 2.0
Audio Engine
Build Infrastructure
Caciocavallo
Closures
Code Tools

OpenJDK

What is this? The place to collaborate on an open-source implementation of the Java Platform, Standard Edition, and related projects. (Learn more.)

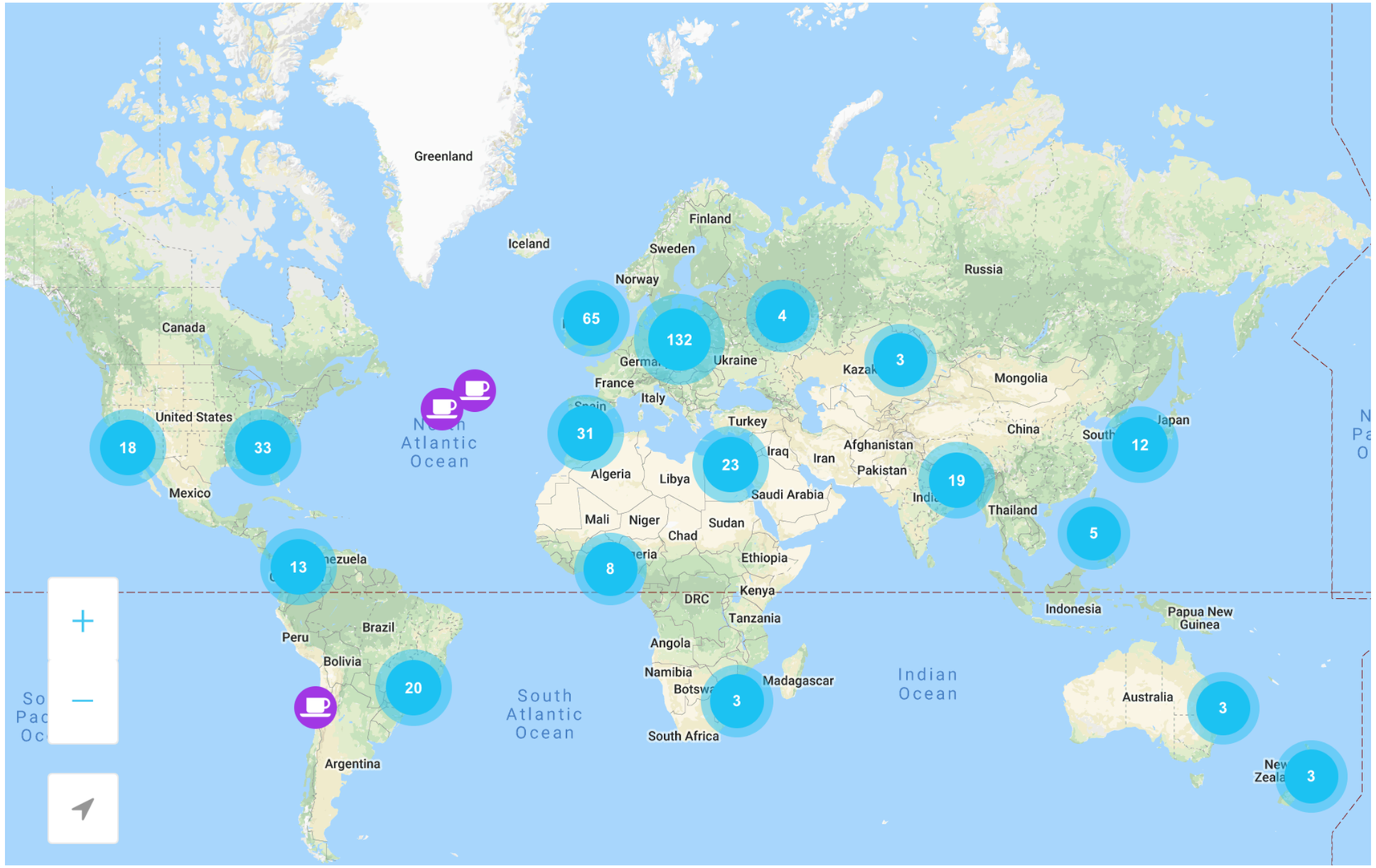
Download and install the open-source JDK for most popular Linux distributions. Oracle's OpenJDK JDK 10 binaries are at **jdk.java.net/10**; Oracle's JDK 10 product binaries for Linux, macOS, and Windows, based largely on the same code, are here.

Learn how to use the JDK to write applications for a wide range of environments.



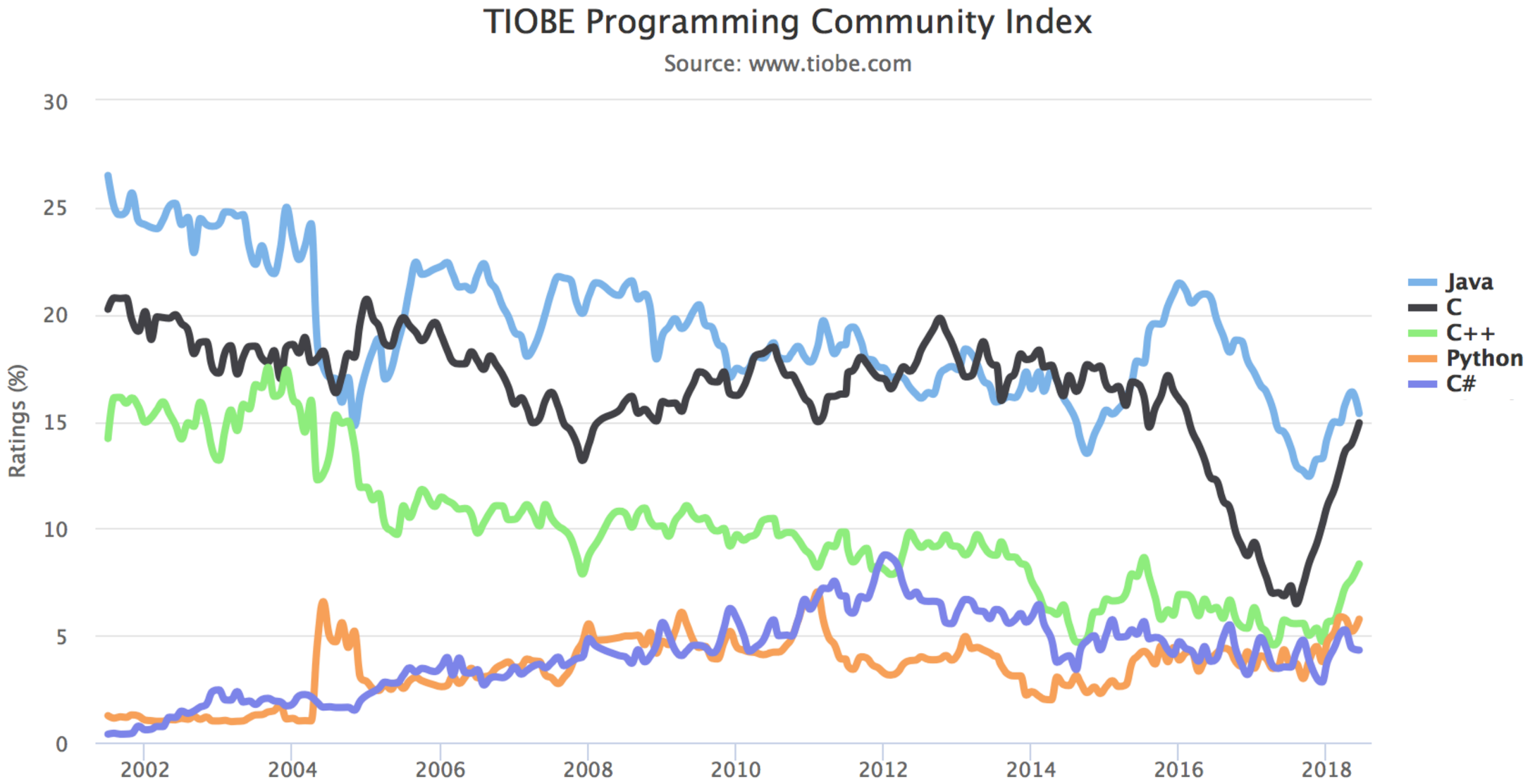
Hack on the JDK itself, right here in the OpenJDK Community: Browse the code on the web, clone a Mercurial repository to make a local copy, and contribute a patch to fix a bug, enhance an existing component, or define a new feature.

Open: Java Community



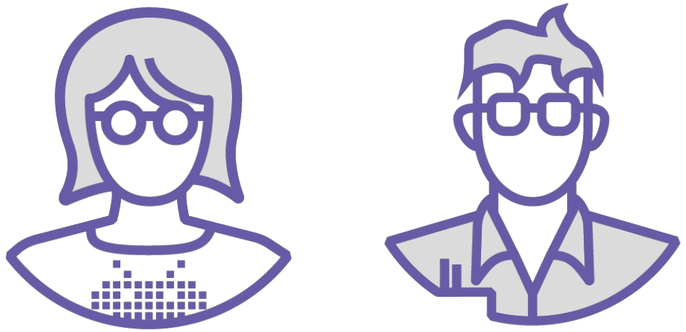
Choosing Java: When and Why

Popularity


TIOBE Programming Community Index
Source: www.tiobe.com
Ratings (%)
30
25
20
15
10
5
0
2002
2004
2006
2008
2010
2012
2014
2016
2018
Java
C
C++
Python
C#

Popularity

Estimated **10 million** Java developers



Web-applications

Backend services

Data-intensive applications

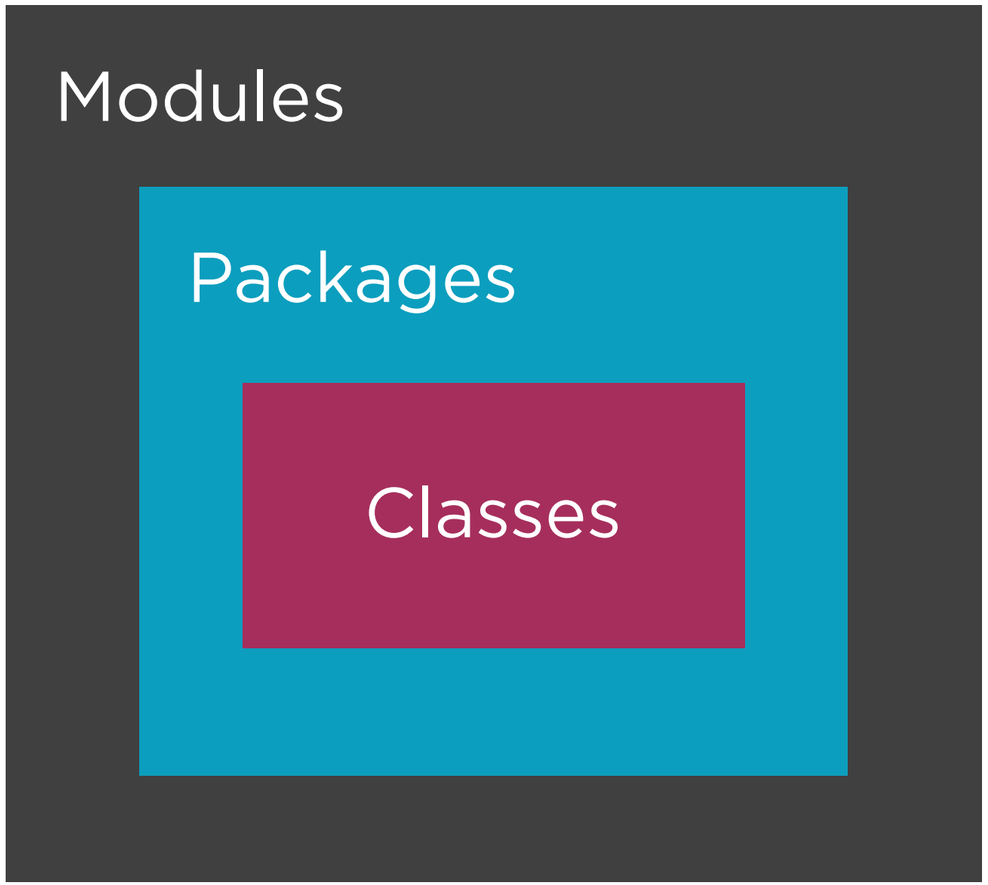
Scalable Development

Hierarchical & structured codebases

Established coding practices

Strong tooling

Wealth of libraries



Productivity: Type System

Catch bugs early

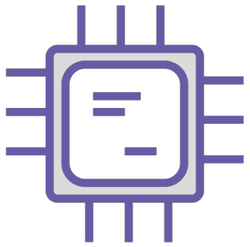
```
public class Hello {

    public static void main(String[] args) {
        int message = "Hello PluralSight!";
        System.out.println(message);
    }

}
```

```
Hello.java:4: error: incompatible types: String
cannot be converted to int
      int message = "Hello PluralSight!";
                    ^
```

Productivity: Managed Runtime

Automatic memory management

Garbage collection



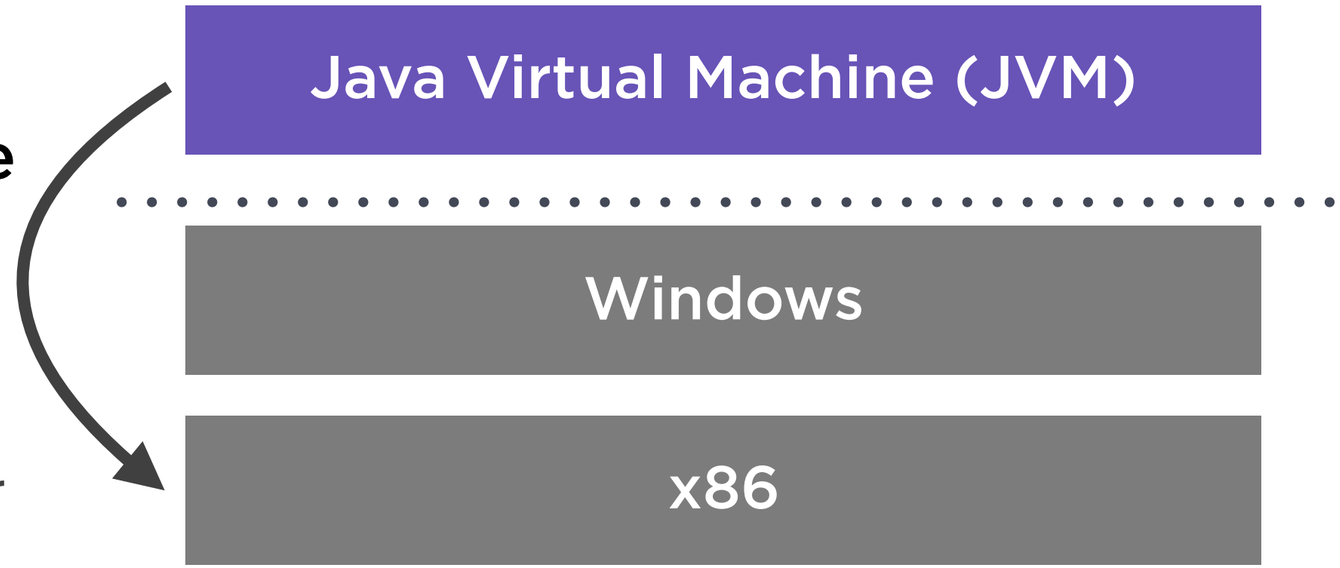
Multi-threading

Performance

Just-in-time compilation

Specialized to underlying hardware

Based on actual execution of code



“When web companies grow up, they turn into Java shops.”

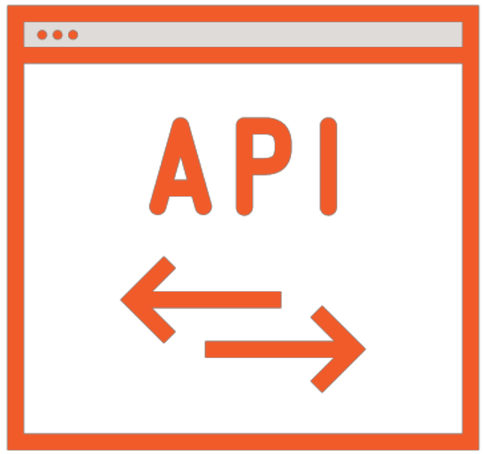
James Governor, RedMonk analyst & co-founder

When Is Java Not the Right Choice?

When Is Java Not the Right Choice?

Real-time systems

When Is Java Not the Right Choice?



Tight operating system integration

When Is Java Not the Right Choice?

Quick prototyping

When Is Java Not the Right Choice?

For developers who want
cutting-edge languages

Comparing Java

C#/.Net C/C++ Python JavaScript

C#/.Net

Common Language Runtime

Managed language

Open-source

Faster moving language

Only recently cross-platform

Ecosystem dominated by Microsoft

C/C++

Java syntax inspired by C

C++ also Object-oriented

Unmanaged language

More language features

Compiled to native code

Python

High-level managed language

Open ecosystem

Interpreted language

Not statically typed

Python 2/3 split

JavaScript

Managed runtime: NodeJS

Somewhat syntactically similar

Interpreted language

Not statically typed

Single-threaded

Summary

Portability, simplicity & openness

Productive, managed language

Comparing Java