Adopting Java SE 17

Sander Mak
Java Champion

@Sander_Mak

When and Why to Adopt Java?

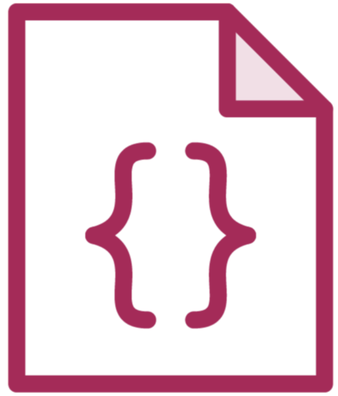
When and Why to Adopt Java?

Philosophy

When and Why to Adopt Java?

Philosophy

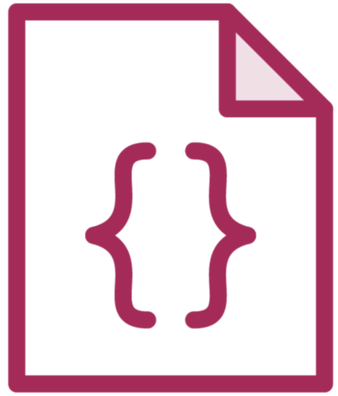
Java's characteristics



When and Why to Adopt Java?

Philosophy

Java's characteristics



Comparison

The Philosophy of Java: Readability

The Philosophy of Java: Readability

More than **10 million** Java developers

The Philosophy of Java: Readability

More than **10 million** Java developers

The Philosophy of Java: Readability

More than **10 million** Java developers

Reading code is more important
than **writing** code

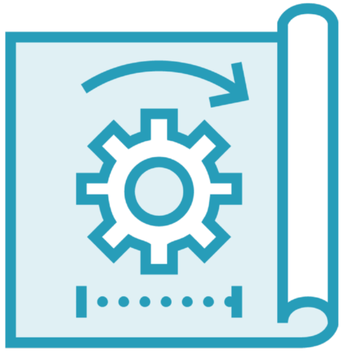
The Philosophy of Java: Readability

More than **10 million** Java developers

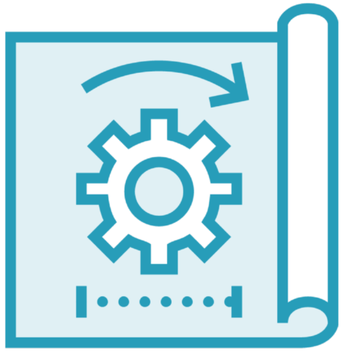
Reading code is more important
than **writing** code

Understandable code over **short** and
clever code

The Philosophy of Java: Stability

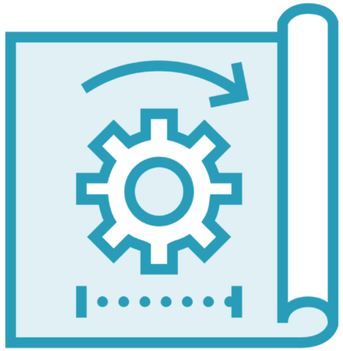


The Philosophy of Java: Stability



Conservative in adding new featurees

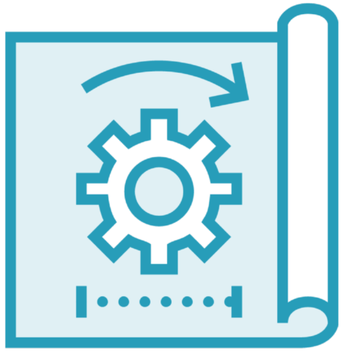
The Philosophy of Java: Stability



Conservative in adding new featurees

Java 1.0	Java 17	
1996	2021	2040?

The Philosophy of Java: Stability

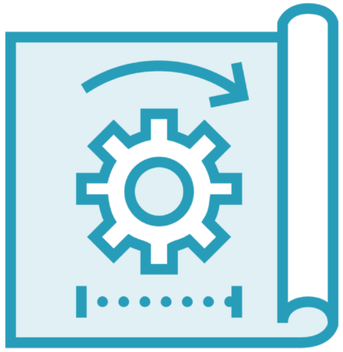


Conservative in adding new featurees

First, do no harm

Java 1.0	Java 17	
1996	2021	2040?

The Philosophy of Java: Stability



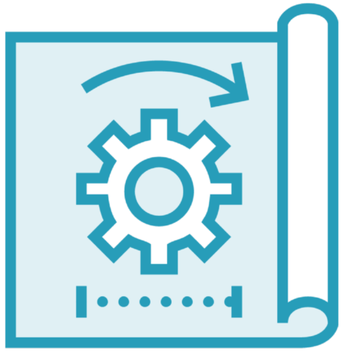
Conservative in adding new featurees

First, do no harm

Productivity

Java 1.0	Java 17	
1996	2021	2040?

The Philosophy of Java: Stability



Conservative in adding new featurees

First, do no harm

Productivity

Java 1.0 **Java 12** **Java 17**

1996

Preview Features

2021

2040?

Backward Compatibility

Backward Compatibility

Backward Compatibility

Backward Compatibility

Java 8 **Java 17**

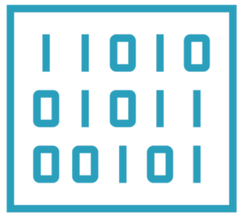
Backward Compatibility

Java 8	Java 17
 Source code	

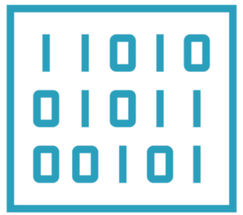
Backward Compatibility

Java 8	Java 17
 Source code	

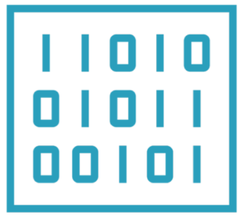
Backward Compatibility

Java 8	Java 17
 Source code	
 Byte code	

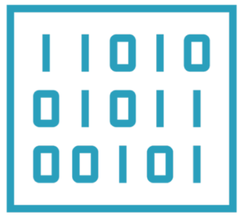
Backward Compatibility

Java 8	Java 17
 Source code	
 Byte code	

Backward Compatibility

Java 8	Java 17	Java N
 Source code	✓	✓
 Byte code	✓	✓

Backward Compatibility

Java 8	Java 17	Java N
 Source code		
 Byte code		

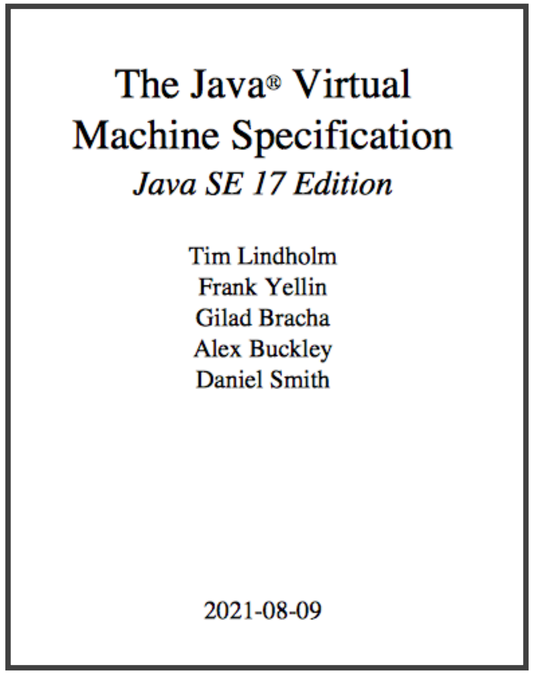
Philosophy of Java: Openness & Community

Philosophy of Java: Openness & Community

Rigorous specification

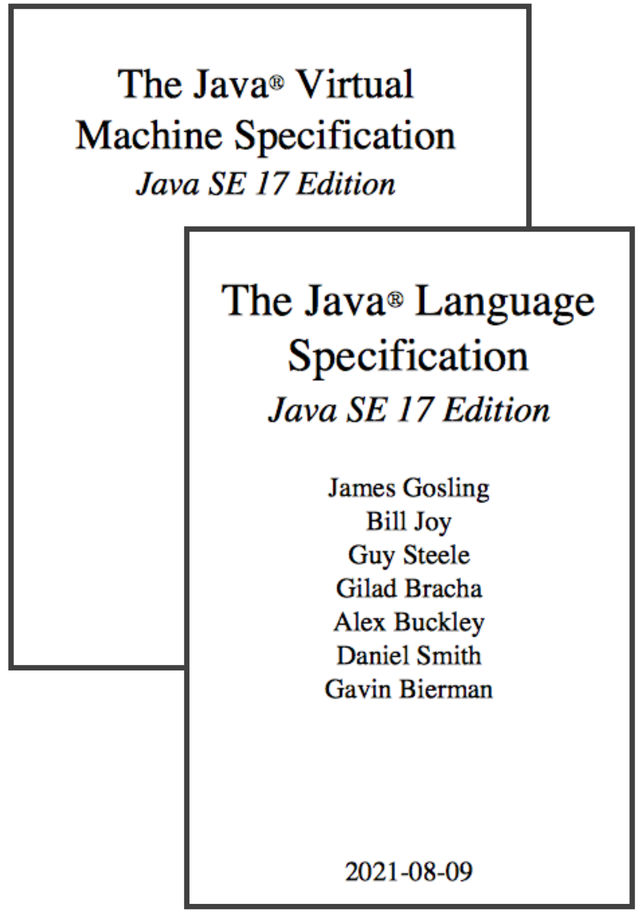
Philosophy of Java: Openness & Community

Rigorous specification



Philosophy of Java: Openness & Community

Rigorous specification

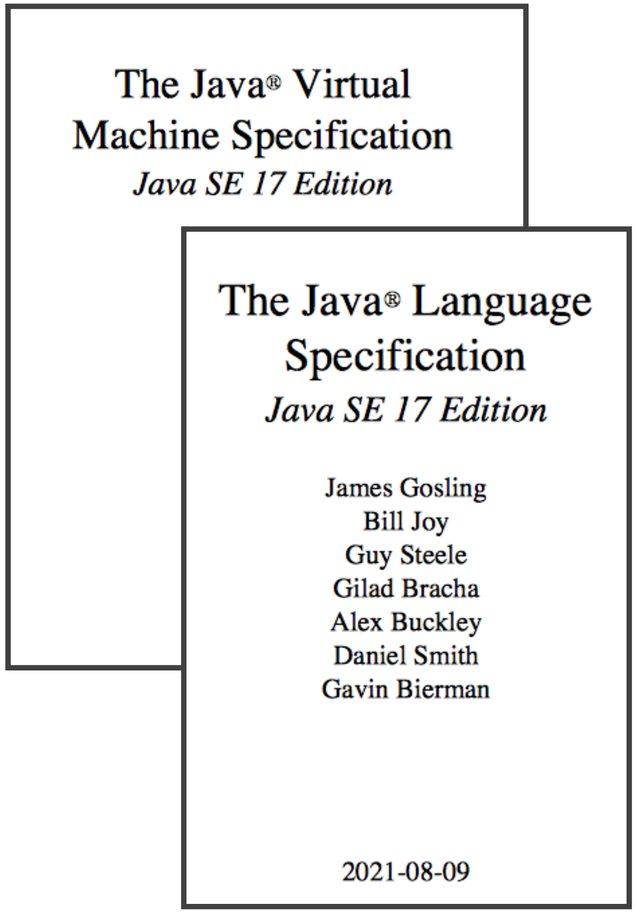


Philosophy of Java: Openness & Community

Rigorous specification

Java Community Process

https://jcp.org



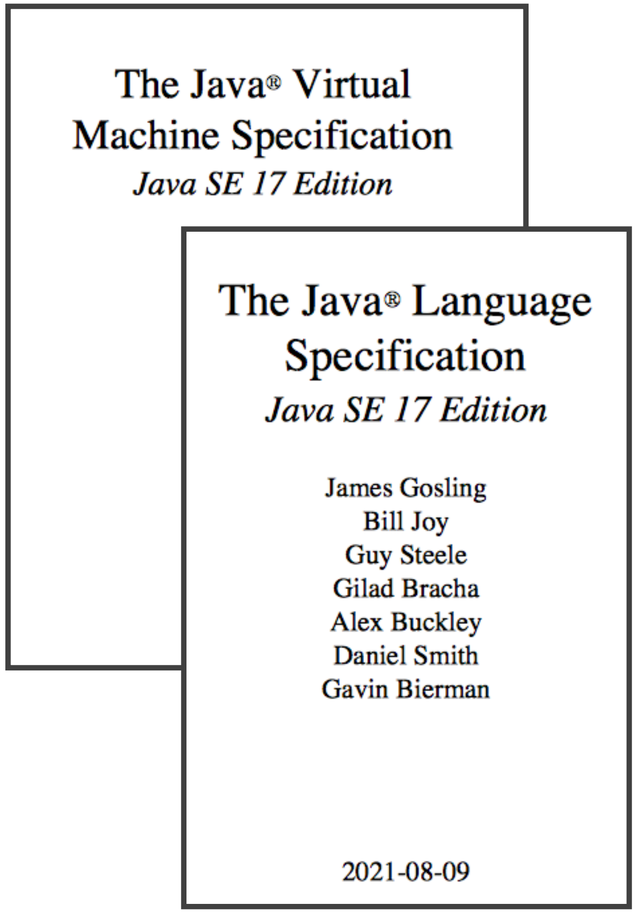
Philosophy of Java: Openness & Community

Rigorous specification

Java Community Process

https://jcp.org

Multiple vendors & community leaders



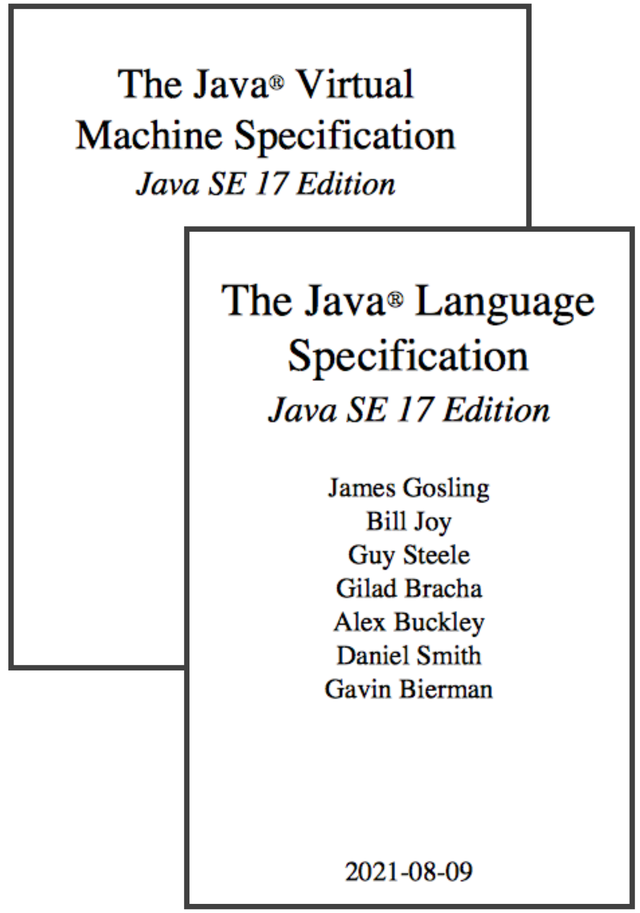
Philosophy of Java: Openness & Community

Rigorous specification

https://jcp.org

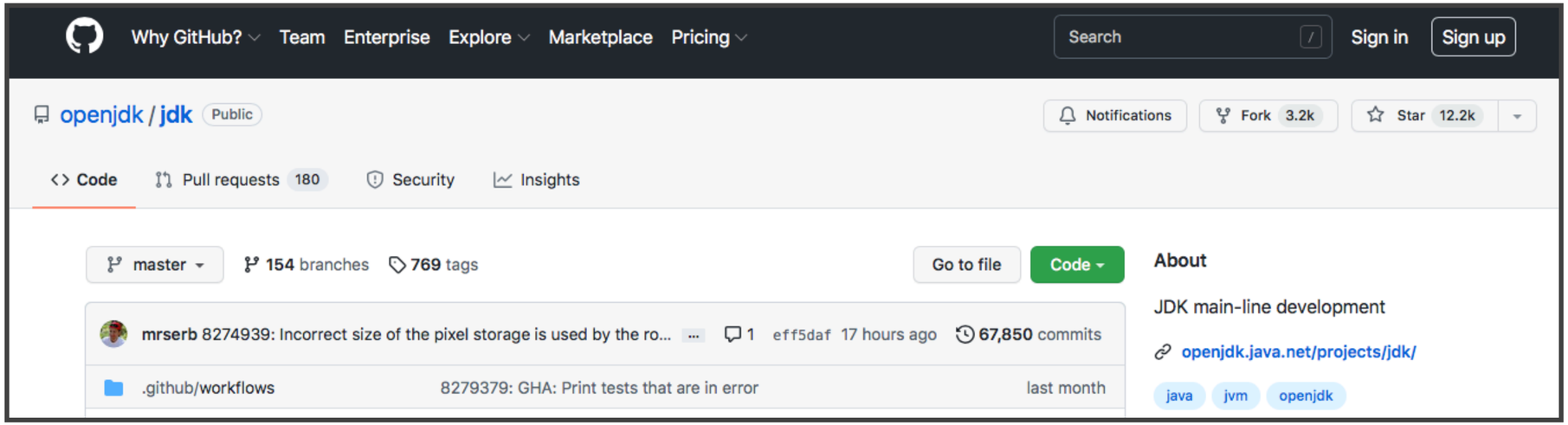
Multiple vendors & community leaders

IBM, Red Hat, Azul, Microsoft, Amazon, ...



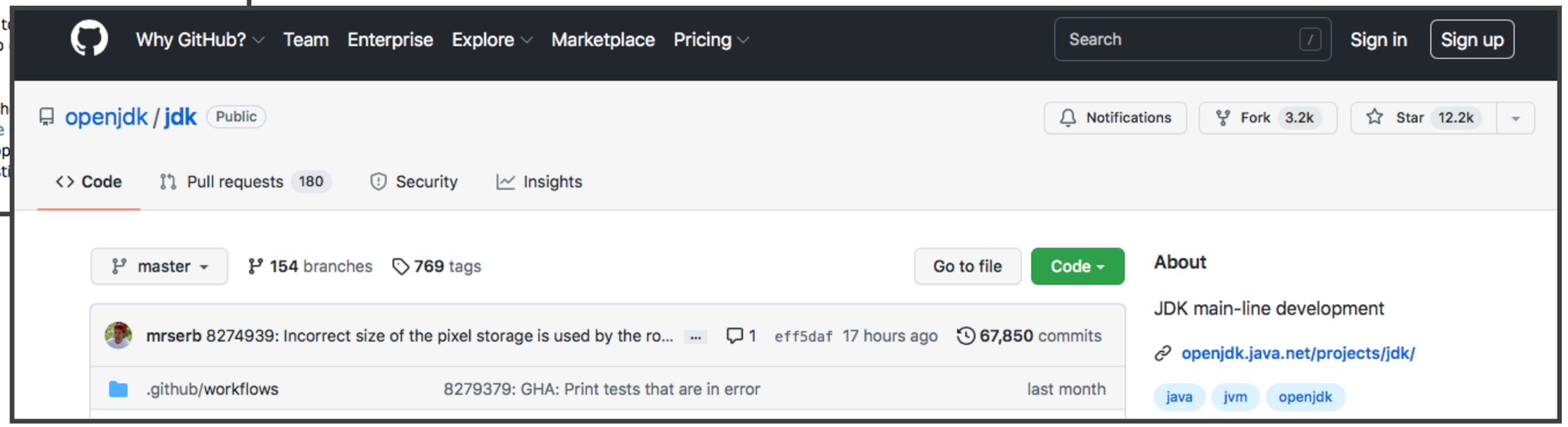
OpenJDK

OpenJDK


Why GitHub?
Team
Enterprise
Explore
Marketplace
Pricing
Search
Sign in
Sign up
openjdk / jdk
Public
Notifications
Fork 3.2k
Star 12.2k
Code
Pull requests 180
Security
Insights
master
154 branches
769 tags
Go to file
Code
About
JDK main-line development
mrserb 8274939: Incorrect size of the pixel storage is used by the ro...
1
eff5daf 17 hours ago
67,850 commits
openjdk.java.net/projects/jdk/
.github/workflows
8279379: GHA: Print tests that are in error
last month
java
jvm
openjdk

OpenJDK

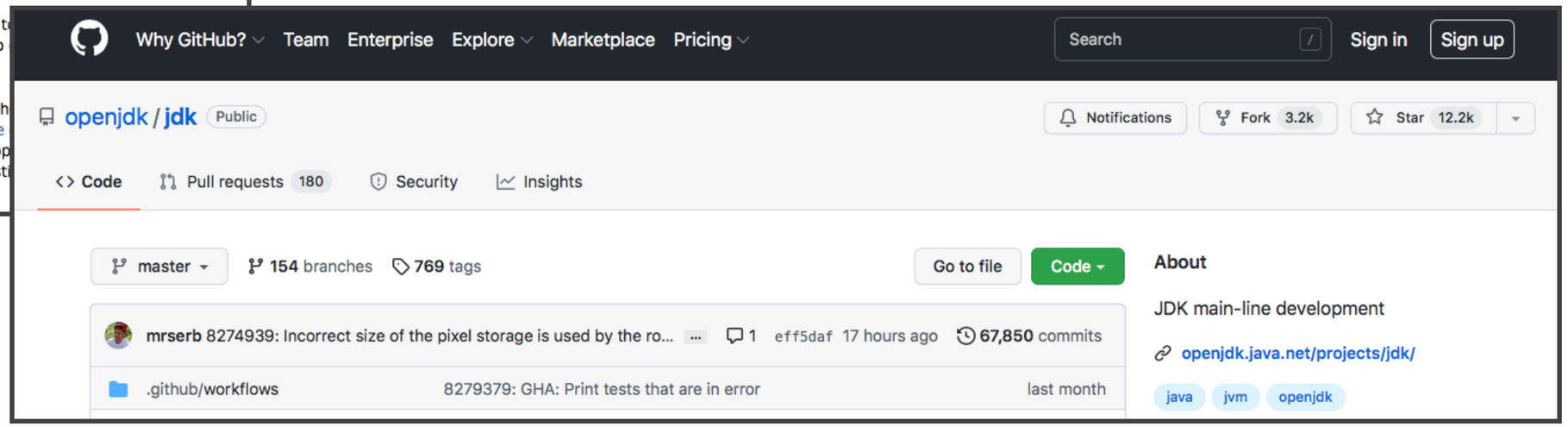
openjdk.net



OpenJDK

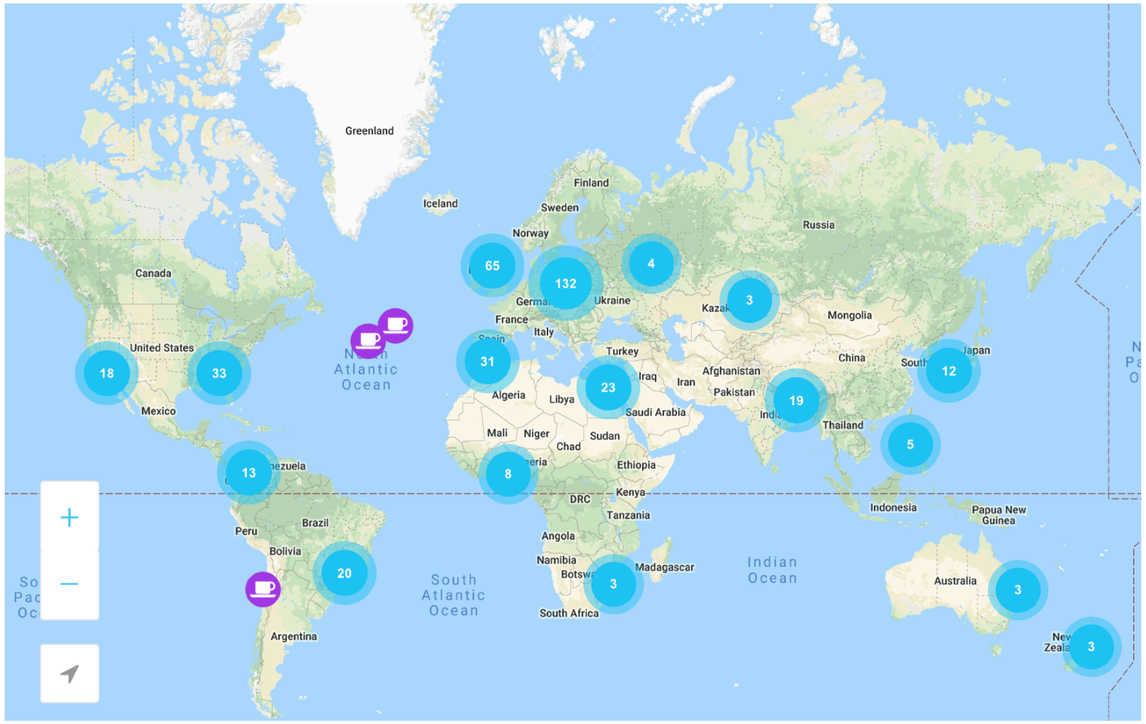
openjdk.net

Java enhancement proposals (JEPs)



Java Community

Java Community



Java User Groups, meetups, conferences

The Java Language

The Java Language

Familiar syntax

The Java Language

Familiar syntax

```
public class MyClass {
    public void aMethod(boolean choice) {




    }

    private void anotherMethod() {



    }
}
```

The Java Language

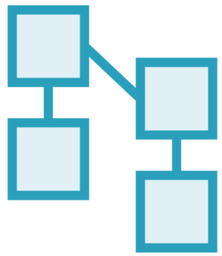
Familiar syntax

```
public class MyClass {
    public void aMethod(boolean choice) {
        if(choice) {
            //..
        } else {
            // ..
        }
    }

    private void anotherMethod() {
        for(int i = 1; 1 < 10; i++) {
            // ..
        }
    }
}
```

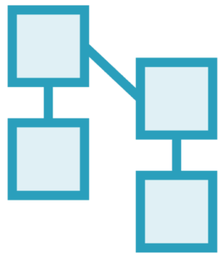
The Java Language

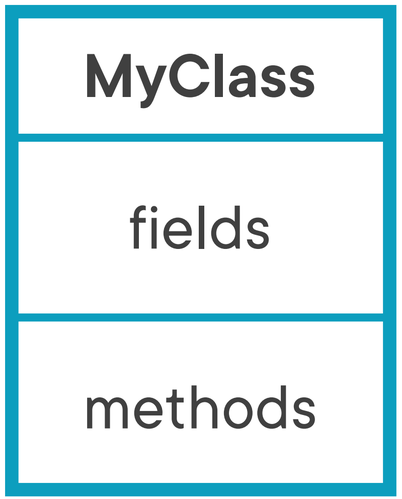
Object Oriented



The Java Language

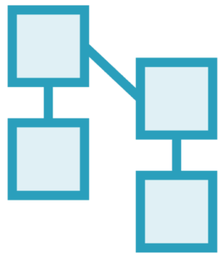
Object Oriented

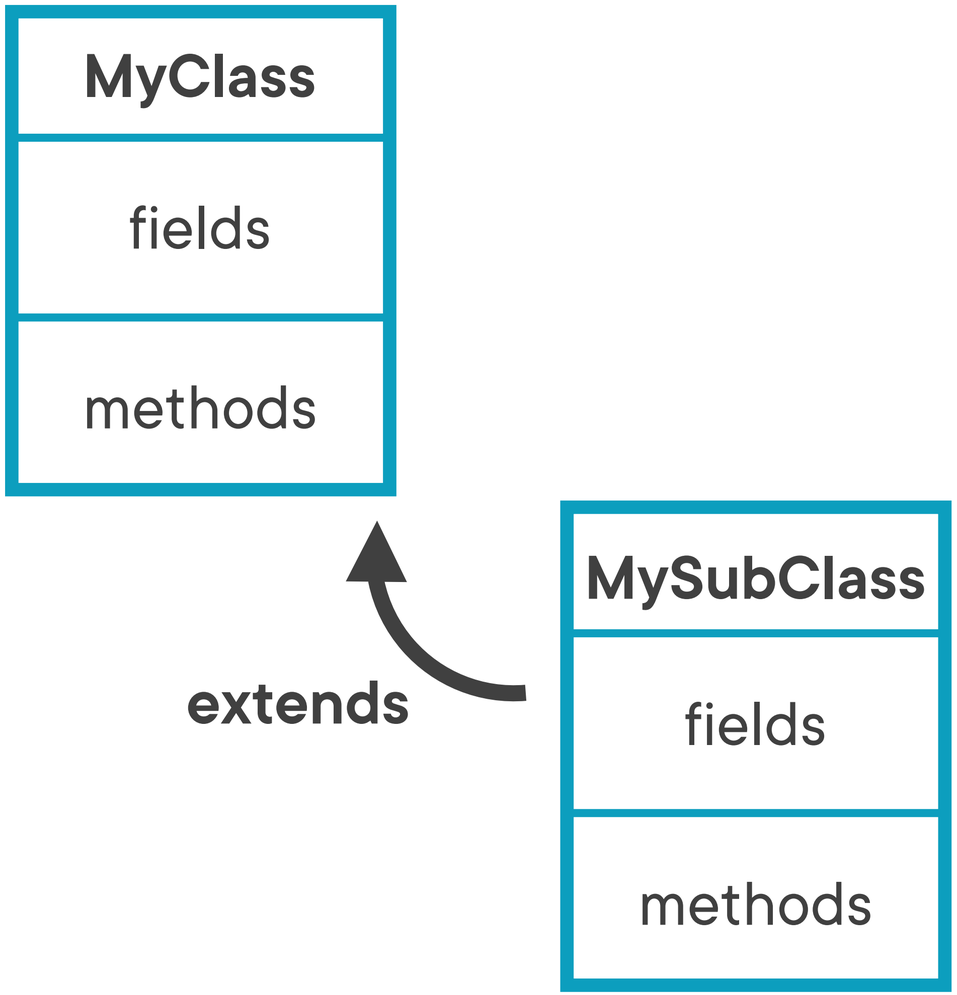




The Java Language

Object Oriented





Scalable Development

Scalable Development

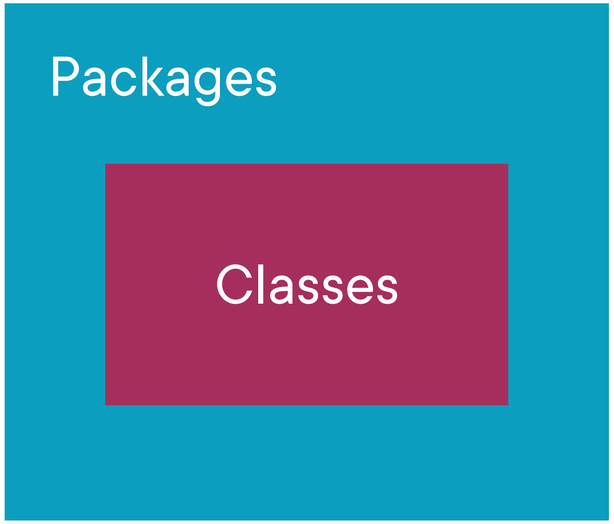
Hierarchical & structured codebases

Scalable Development

Hierarchical & structured codebases

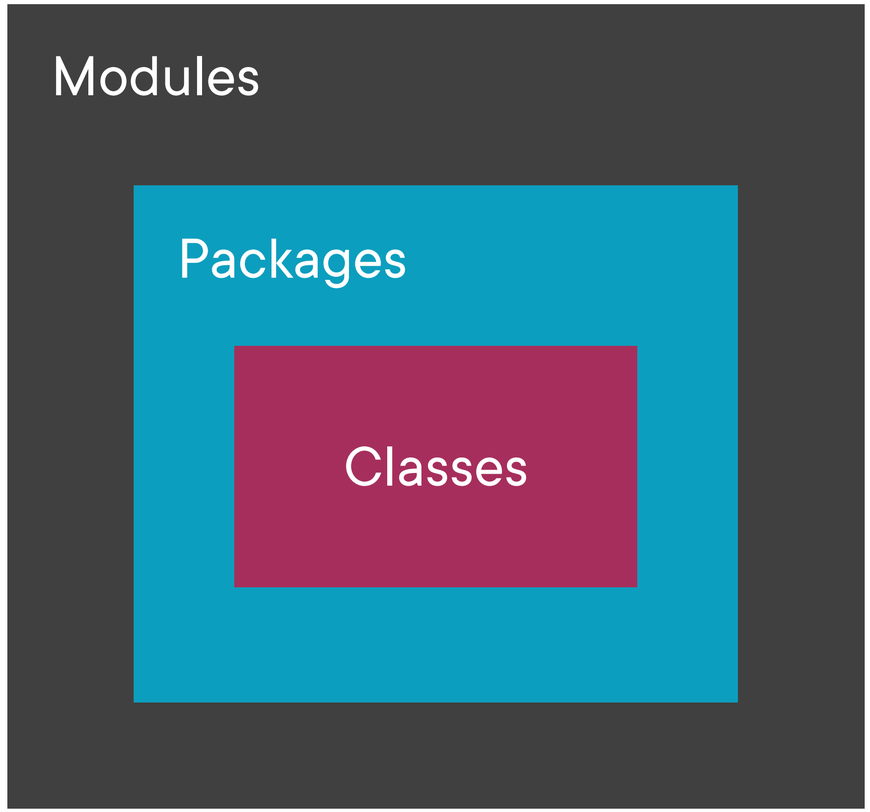
Scalable Development

Hierarchical & structured codebases



Scalable Development

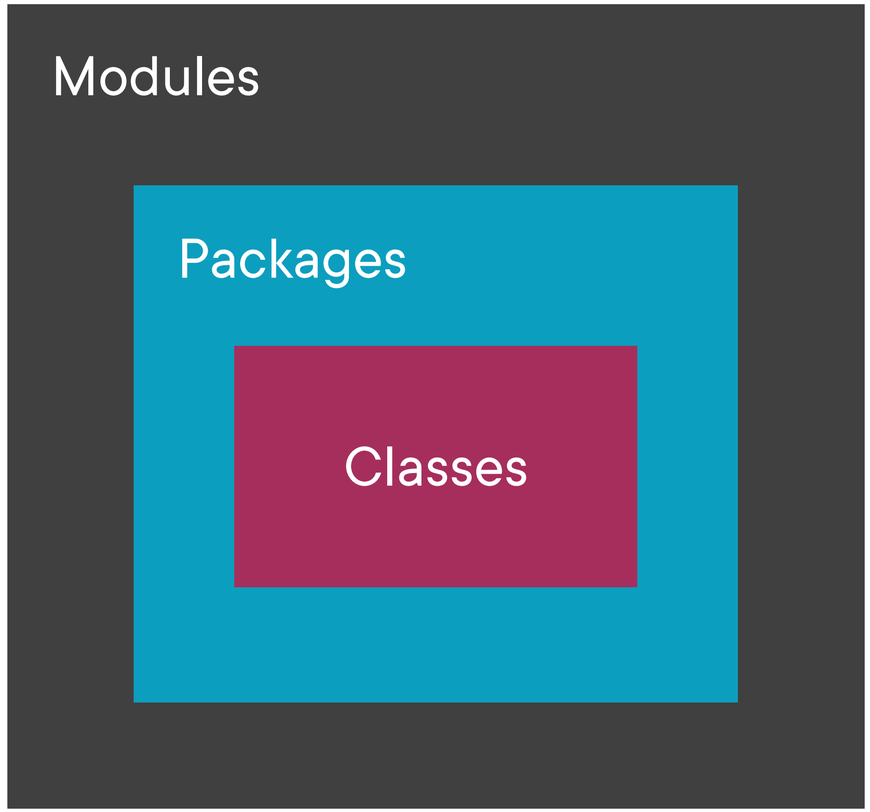
Hierarchical & structured codebases



Scalable Development

Hierarchical & structured codebases

Working with the Java Module System



Productivity: Type System

Productivity: Type System

Catch bugs early

Productivity: Type System

Catch bugs early

```
public class Hello {

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

}
```

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!";
                    ^
1 error
```

Java's Runtime

Java's Runtime

Portability

Java's Runtime

Portability

Managed Runtime

Java's Runtime

Portability

Managed Runtime

Performance

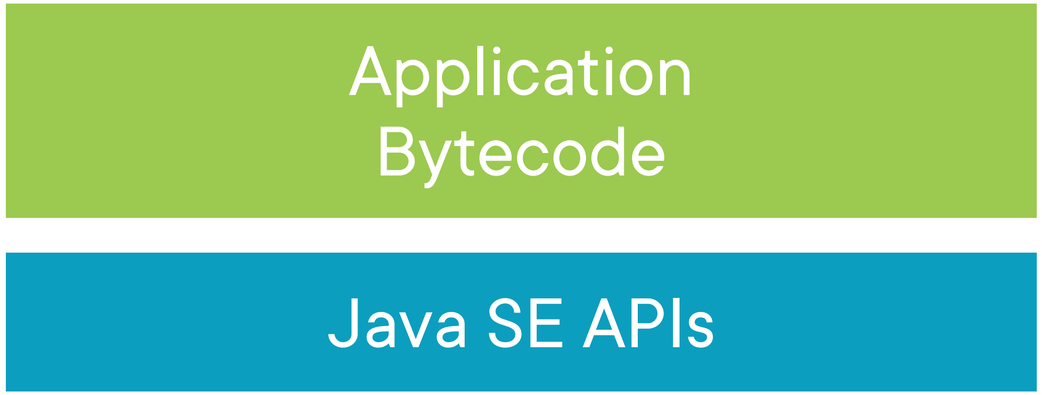
Java's Runtime: Portability

Java's Runtime: Portability

Write
Once
Run
Anywhere

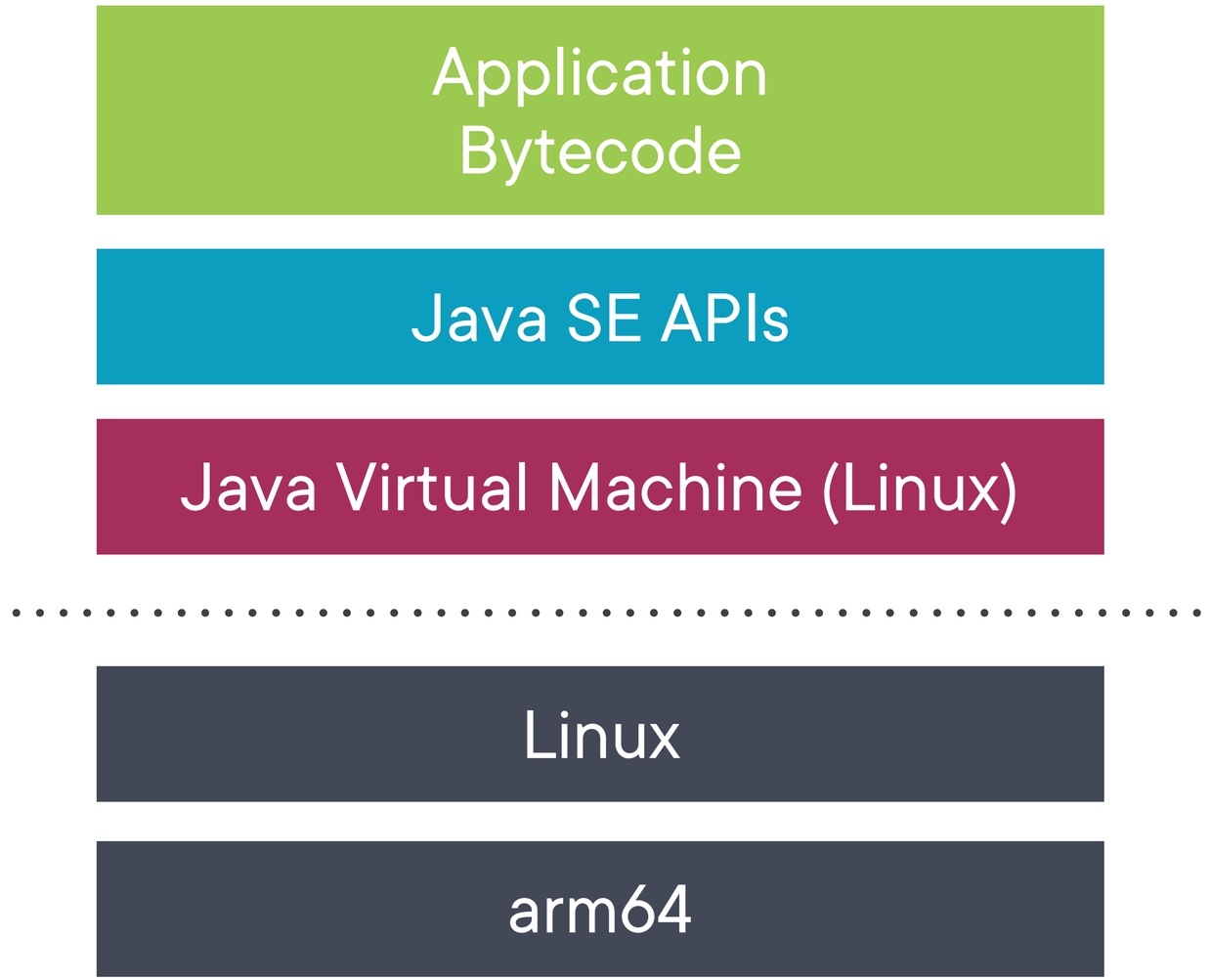
Java's Runtime: Portability

Write
Once
Run
Anywhere



Java's Runtime: Portability

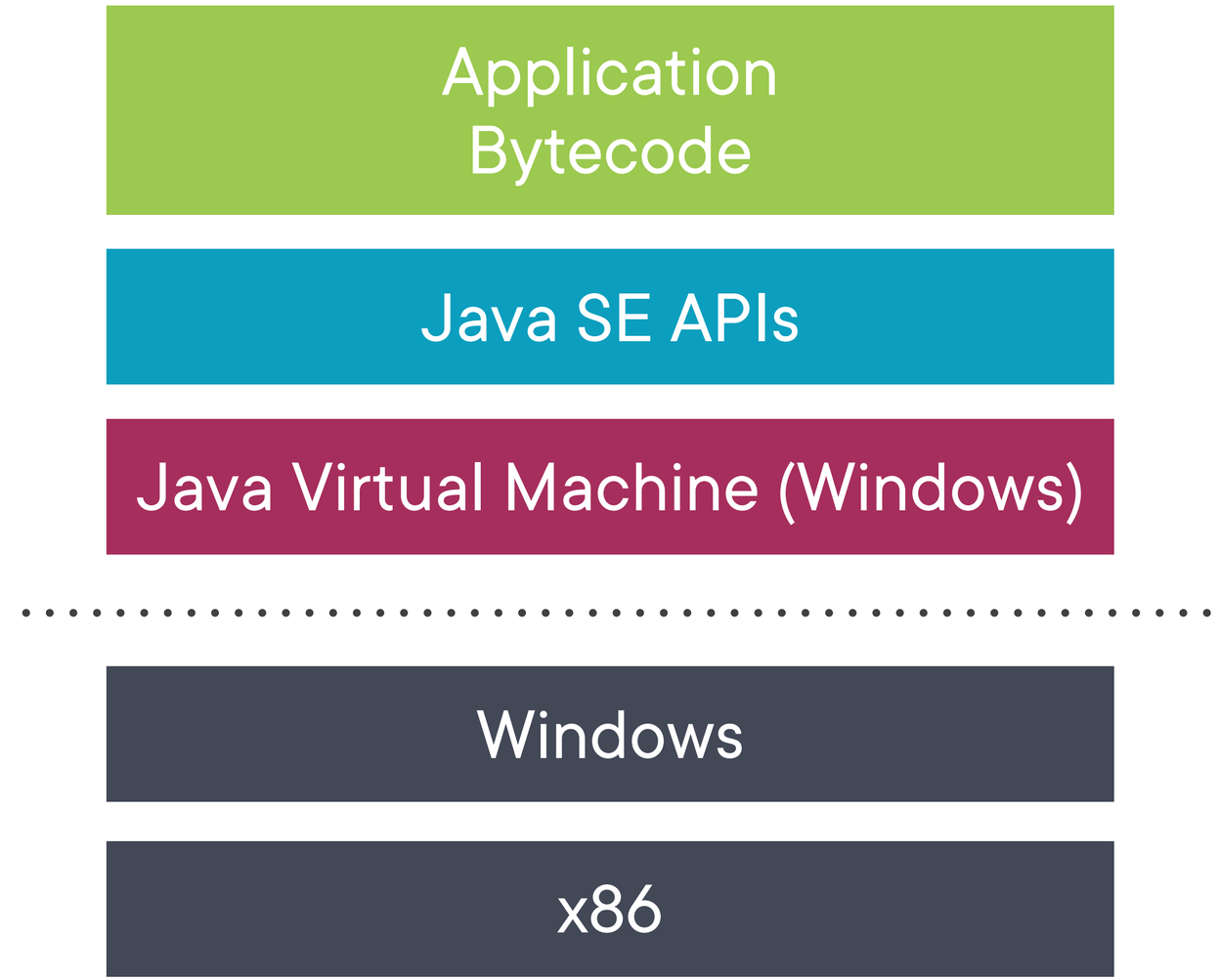
Write
Once
Run
Anywhere



Java's Runtime: Portability

Write
Once
Run
Anywhere

JVM for each OS/architecture

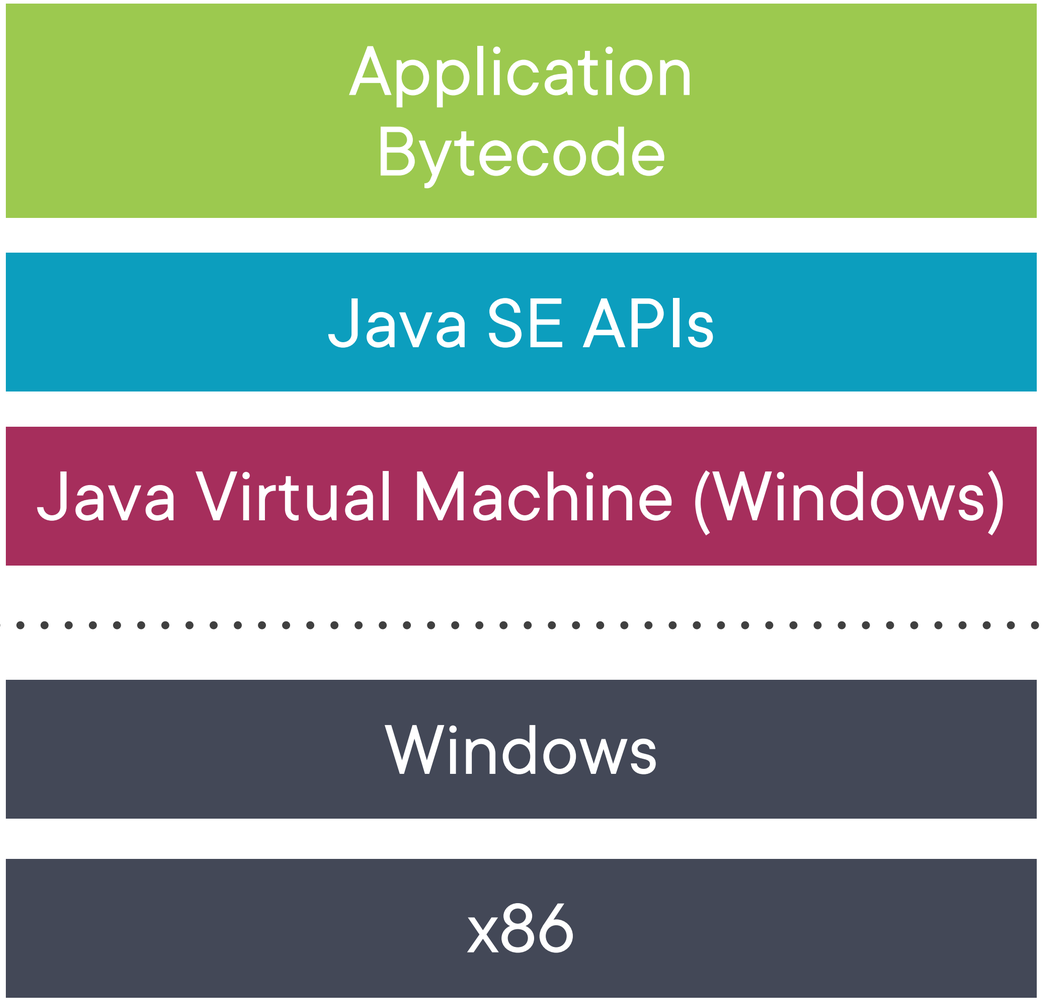


Java's Runtime: Portability

Write
Once
Run
Anywhere

JVM for each OS/architecture

Platform-agnostic APIs



Managed Runtime

Managed Runtime

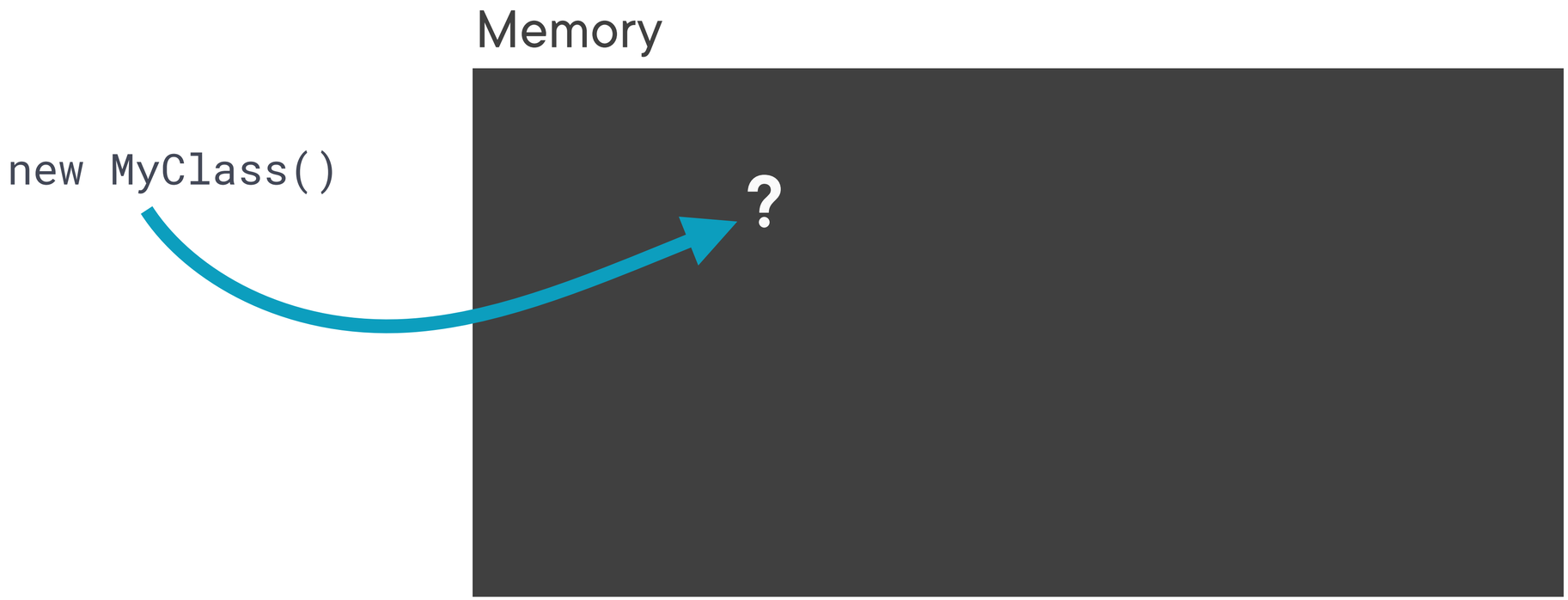
Automatic Memory Management

Managed Runtime

Memory

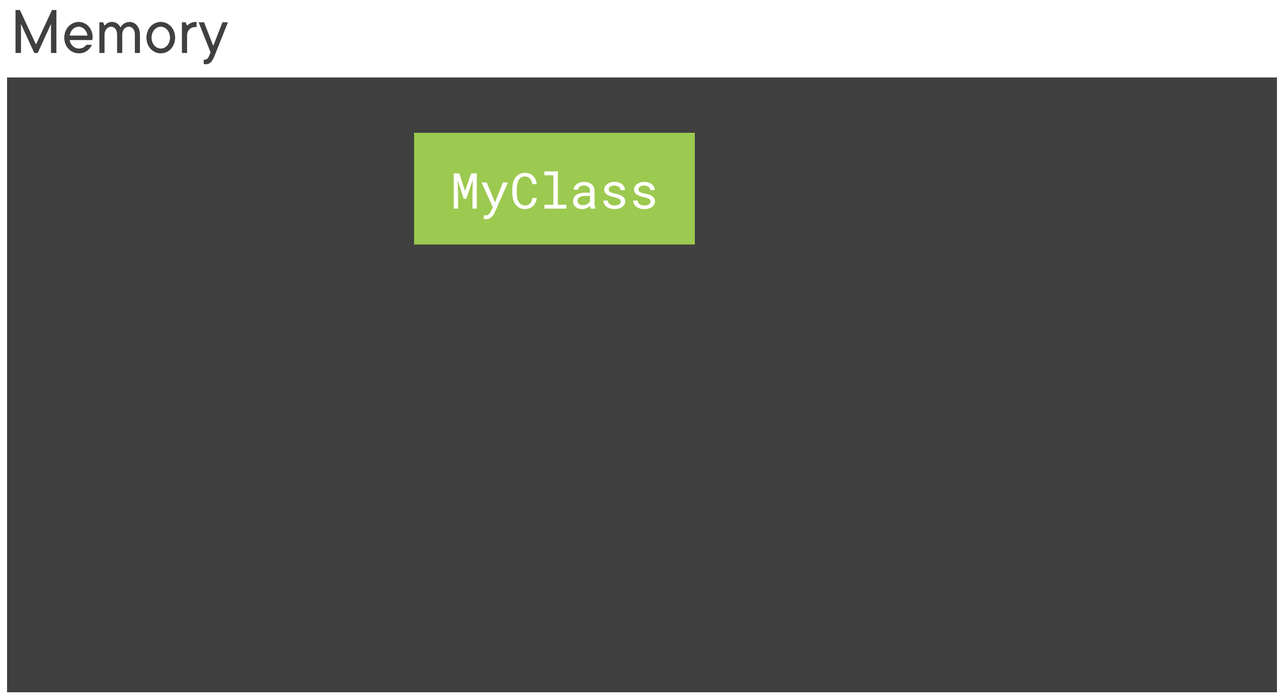
Automatic Memory Management

Managed Runtime



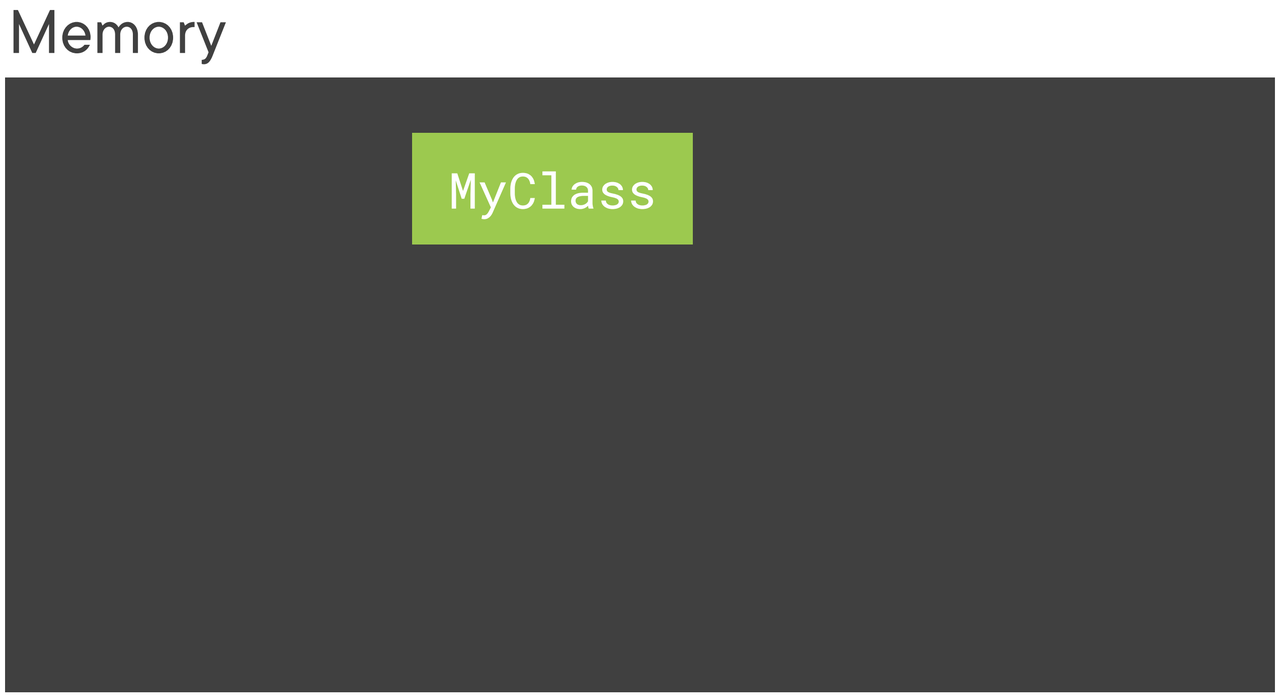
Automatic Memory Management

Managed Runtime



Automatic Memory Management

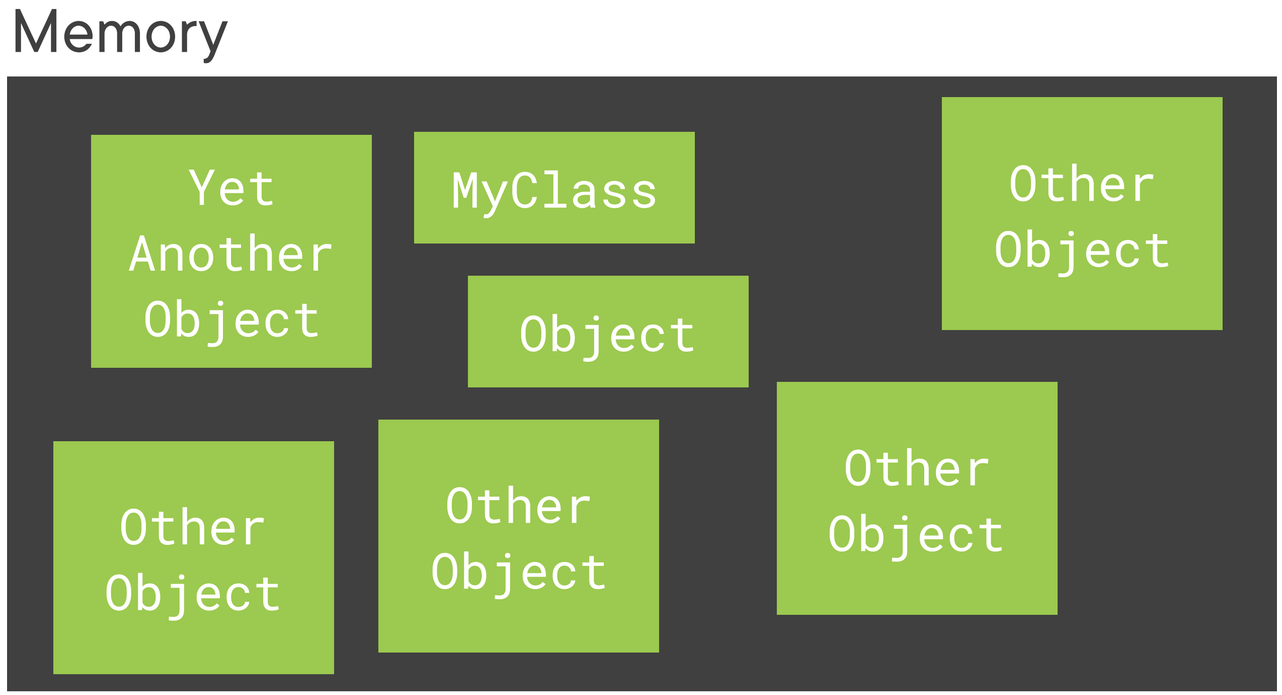
Managed Runtime



Automatic Memory Management

Garbage collection

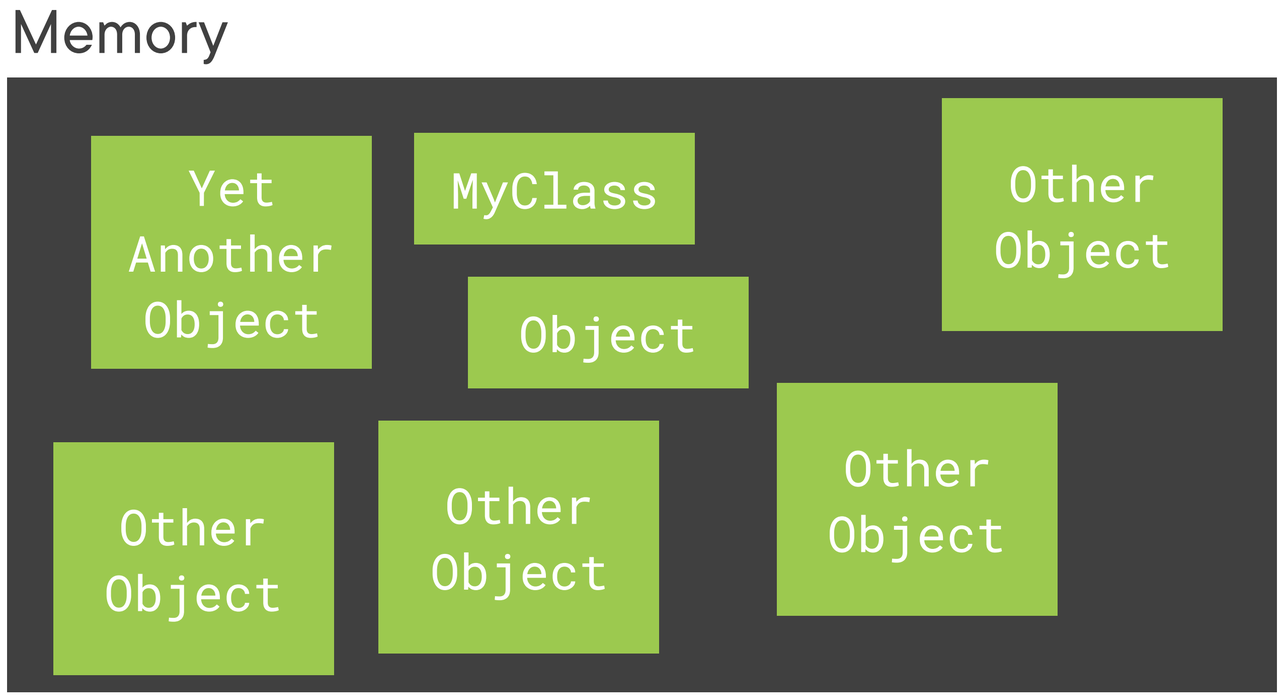
Managed Runtime



Automatic Memory Management

Garbage collection

Managed Runtime

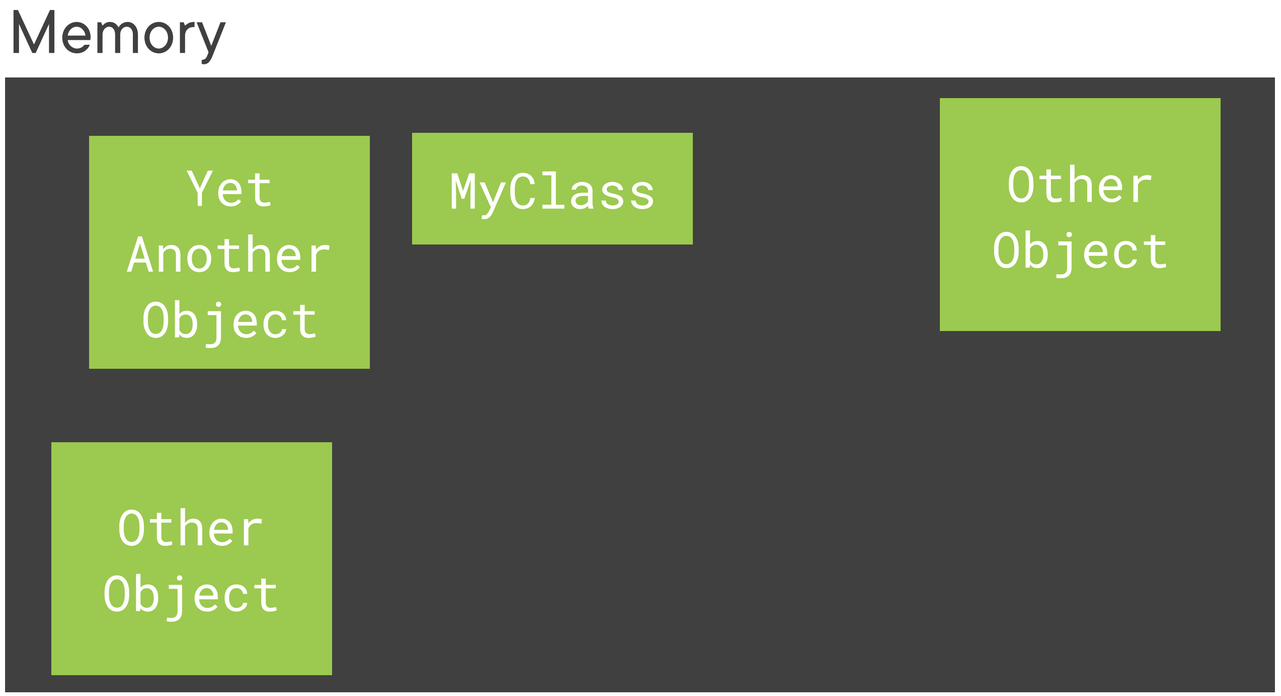


Garbage Collector

Automatic Memory Management

Garbage collection

Managed Runtime



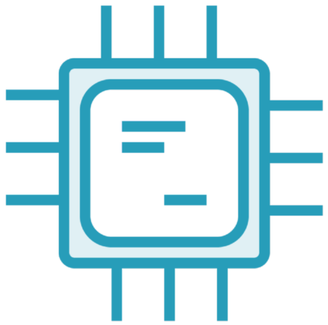
Garbage Collector

Automatic Memory Management

Garbage collection

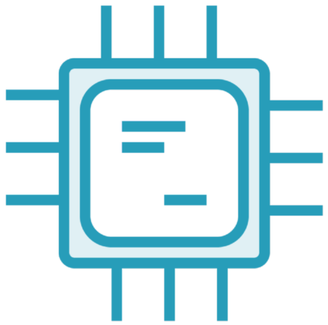
Managed Runtime: Multi-threading

Managed Runtime: Multi-threading

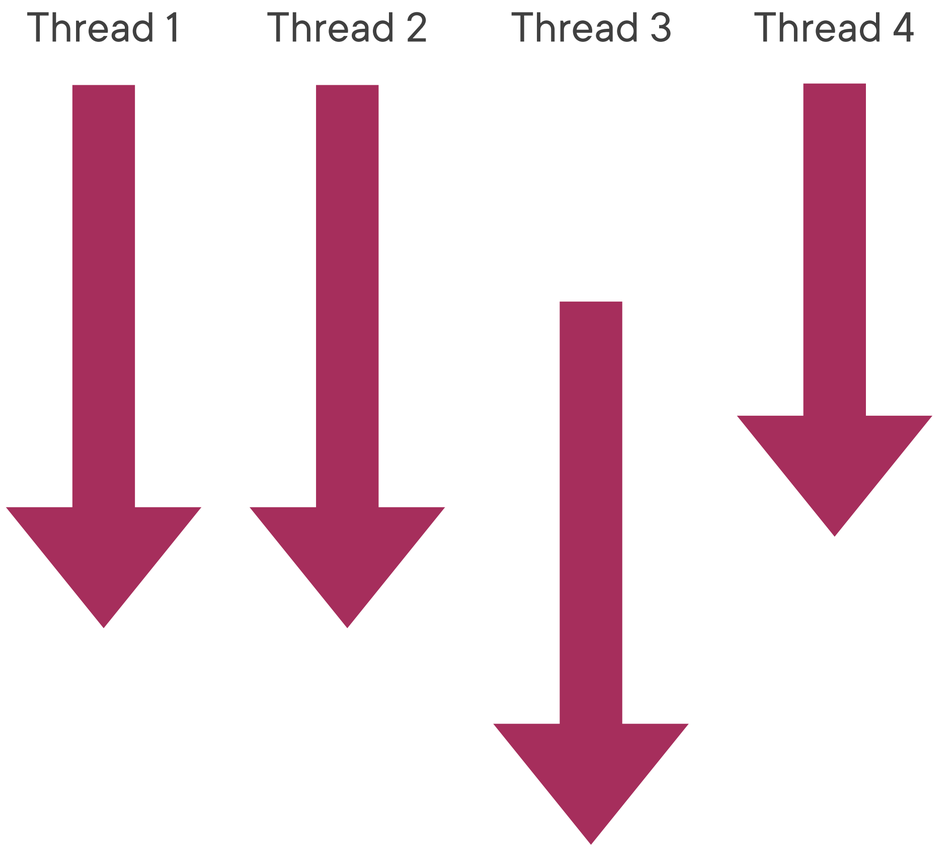


Multi-core
processors

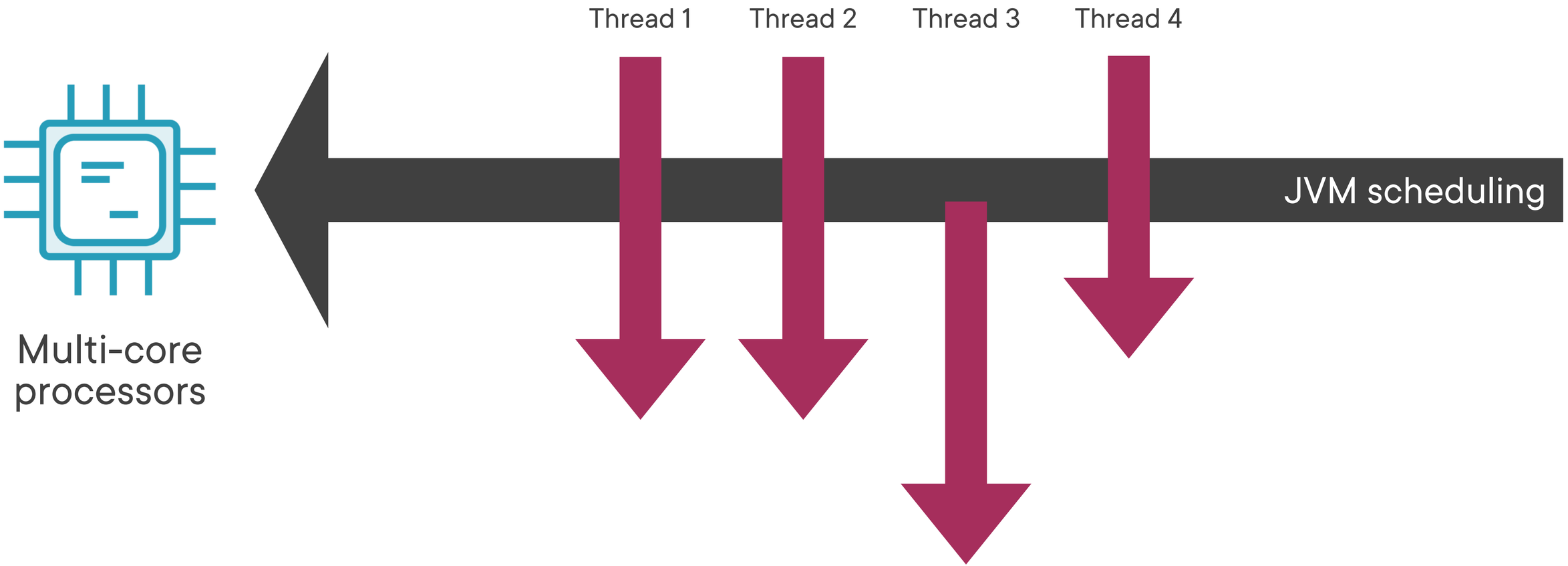
Managed Runtime: Multi-threading



Multi-core processors



Managed Runtime: Multi-threading



Managed Runtime: Performance

Managed Runtime: Performance

Java Virtual Machine (Windows)

Windows

x86

Managed Runtime: Performance

Just-in-time compilation

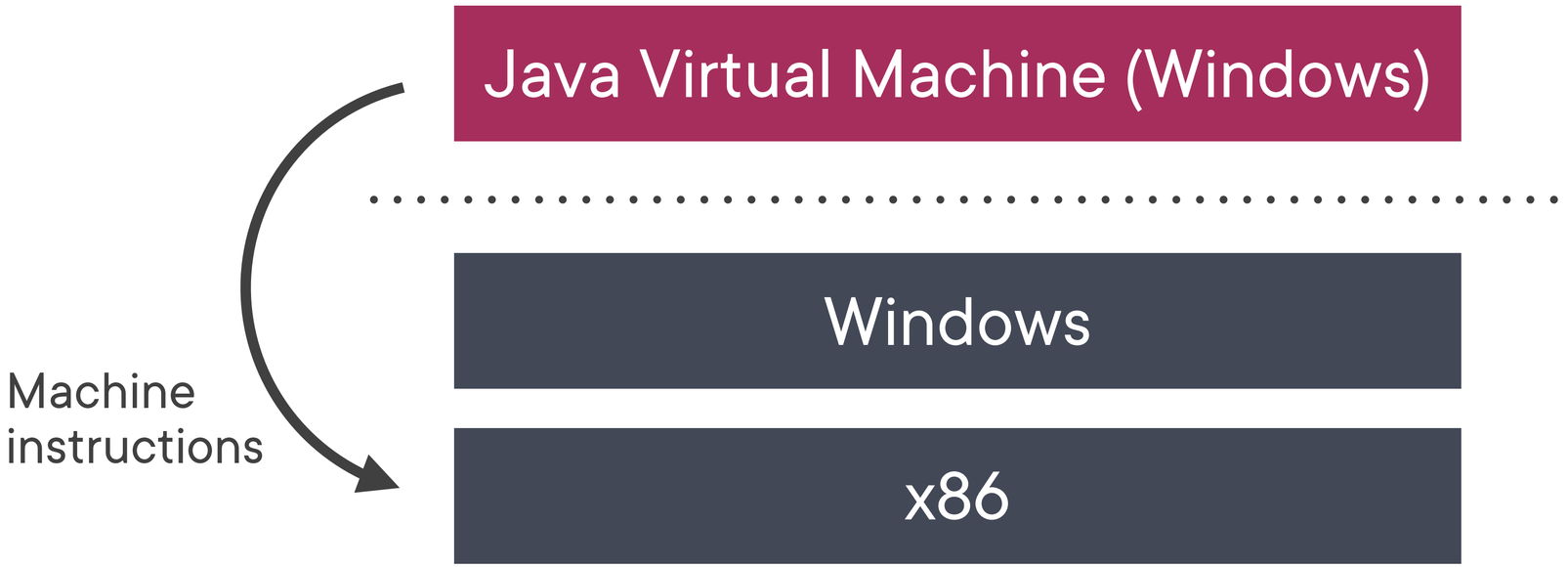
Java Virtual Machine (Windows)

Windows

x86

Managed Runtime: Performance

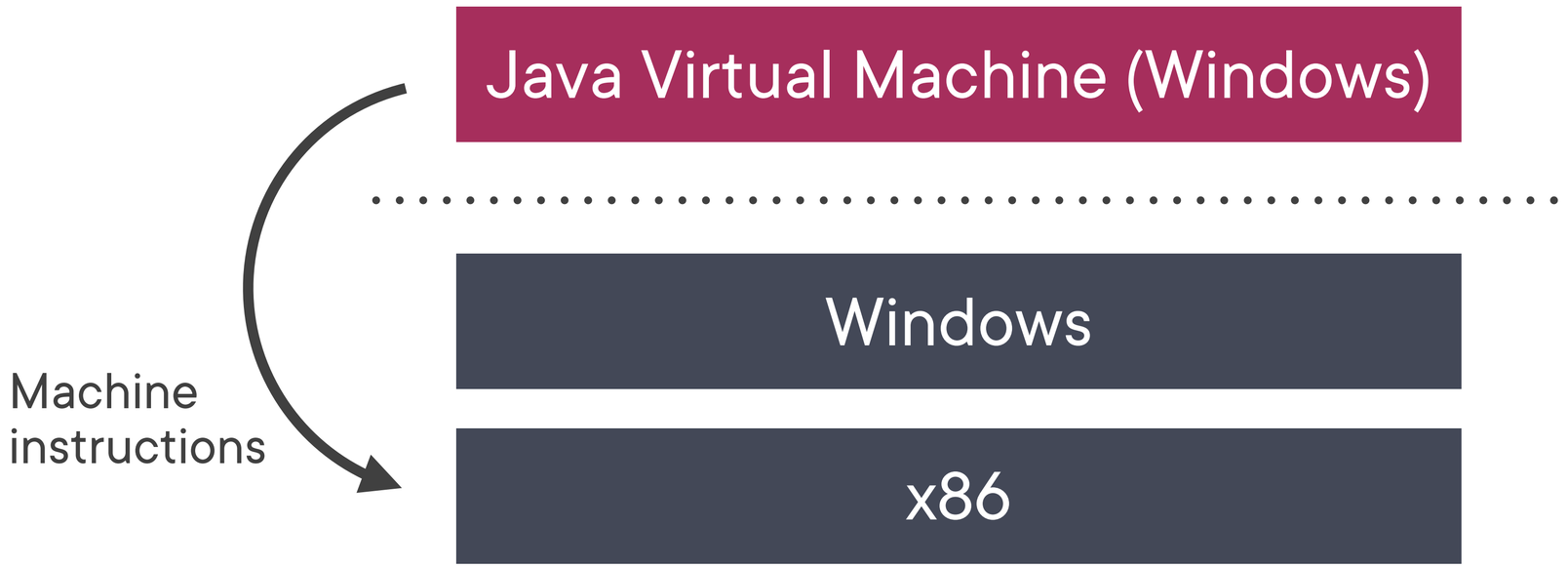
Just-in-time compilation



Managed Runtime: Performance

Just-in-time compilation

Specialized to executing processor

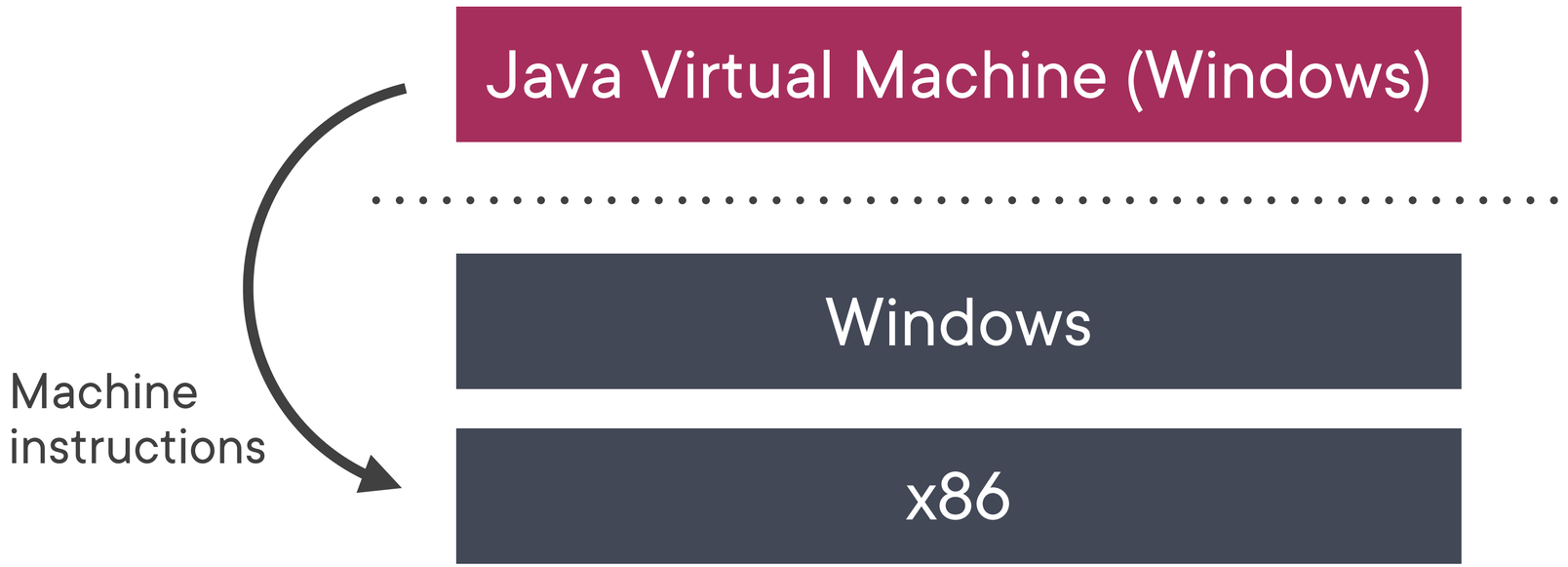


Managed Runtime: Performance

Just-in-time compilation

Specialized to executing processor

Based on actual execution of code



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

James Governor, RedMonk analyst & co-founder

Comparing Java to Other Languages

Comparing Java to Other Languages

C# / .Net

Comparing Java to Other Languages

C# / .Net

C / C++

Comparing Java to Other Languages

C# / .Net

C / C++

JavaScript

C# / .Net

C# / .Net

Common Language Runtime (CLR): a managed runtime

C# / .Net

Common Language Runtime (CLR): a managed runtime

Intermediate Language (IL): like Java bytecode

C# / .Net

Common Language Runtime (CLR): a managed runtime

Intermediate Language (IL): like Java bytecode

C# / .Net

Common Language Runtime (CLR): a managed runtime

Intermediate Language (IL): like Java bytecode

Wider range of language features

C# / .Net

Common Language Runtime (CLR): a managed runtime

Intermediate Language (IL): like Java bytecode

Wider range of language features

More liberal in breaking backward compatibility

C# / .Net

Common Language Runtime (CLR): a managed runtime

Intermediate Language (IL): like Java bytecode

Wider range of language features

More liberal in breaking backward compatibility

Not originally cross-platform

C / C++

C / C++

Similar syntax

C / C++

Similar syntax

Unmanaged languages

C / C++

Similar syntax

Unmanaged languages

Low-level: more freedom, also more error prone

C / C++

Similar syntax

Unmanaged languages

Low-level: more freedom, also more error prone

Separately compile for each target platform

JavaScript

JavaScript

JavaScript with Node.js: managed runtime

JavaScript

JavaScript with Node.js: managed runtime

High-level code

JavaScript

JavaScript with Node.js: managed runtime

High-level code

JavaScript

JavaScript with Node.js: managed runtime

High-level code

Interpreted language: no compilation

JavaScript

JavaScript with Node.js: managed runtime

High-level code

Interpreted language: no compilation

No static type system

JavaScript

JavaScript with Node.js: managed runtime

High-level code

Interpreted language: no compilation

No static type system (TypeScript!)

JavaScript

JavaScript with Node.js: managed runtime

High-level code

Interpreted language: no compilation

No static type system (TypeScript!)

Single-threaded

Summary

Summary

Summary

Readability & stability

Summary

Readability & stability

Open community, open-source

Summary

Readability & stability

Open community, open-source

Object-oriented on managed runtime