

Closures and the Java™ VM -or- Language Parity

Neal Gafter
VS Languages

Microsoft®

JVM Languages

- Many Languages
- Many Language Features

JVM Languages

- Many Languages
- Many Language Features
- Many Libraries

J2SE Libraries

What language should libraries be written in?

- Java?

J2SE Libraries

What language should libraries be written in?

- Java?
- C++?

J2SE Libraries

What language should libraries be written in?

- Java?
- C++?
- M4?

J2SE Libraries

What language should libraries be written in?

- Java?
- C++?
- M4?
- Scala?
- Groovy?

J2SE Libraries

What language should libraries be written in?

- Java?
- C++?
- M4?
- Scala?
- Groovy?
- Ruby? Python?
- Clojure? JavaFXScript? Fan? PHP? Fortress?

Why Not?

- Language vs VM Mismatch
 - Translation artifacts
 - Visible to programmers

Why Not?

- Language vs VM Mismatch
 - Translation artifacts
 - Visible to programmers
- Performance

Why Not?

- Language vs VM Mismatch
 - Translation artifacts
 - Visible to programmers
- Performance
- Semantic Mismatch

Why Not?

- Language vs VM Mismatch
 - Translation artifacts
 - Visible to programmers
- Performance
- Semantic Mismatch
- Conservatism

Consequences

- Platform libraries styled to *Java* idioms
- Non-idiomatic in other languages
- *Languages are not on an equal footing*

Consequences

- Platform libraries styled to *Java* idioms
- Non-idiomatic in other languages

What do we do?

- Wrapper/shim libraries
- One for each language/API combination
- Supports interoperation with Java

What do we do?

- Wrapper/shim libraries
- One for each language/API combination
- Supports interoperation with Java
- But limited use with other languages

Can we do better?

- Add common language features to Java
 - Dynamic method dispatch
 - Proper tail calls
 - Rich identifiers (e.g. “+”)

Can we do better?

- Add common language features to Java
 - Dynamic method dispatch
 - Proper tail calls
 - Rich identifiers (e.g. “+”)
 - Keyword parameters
 - Tagged/extended numerics

Can we do better?

- Add common language features to Java
 - Dynamic method dispatch
 - Proper tail calls
 - Rich identifiers (e.g. “+”)
 - Keyword parameters
 - Tagged/extended numerics
 - **Lambda expressions**
 - Continuations

Why lambda expressions

- In every language on the JVM *except Java*
- Java's future APIs need them
 - Jsr166y ParallelArray
- Improves Interopability

Is lambda too hard?

Is lambda too hard?

- “VS 2008 is still relatively new, but there are a good chunk of VB.Net developers using lambda expressions now, and they love them. Can’t wait to get their hands on statement lambdas in VB10!”

Closures and the Java™ VM -or- Language Parity

Neal Gafter
VS Languages

Microsoft®