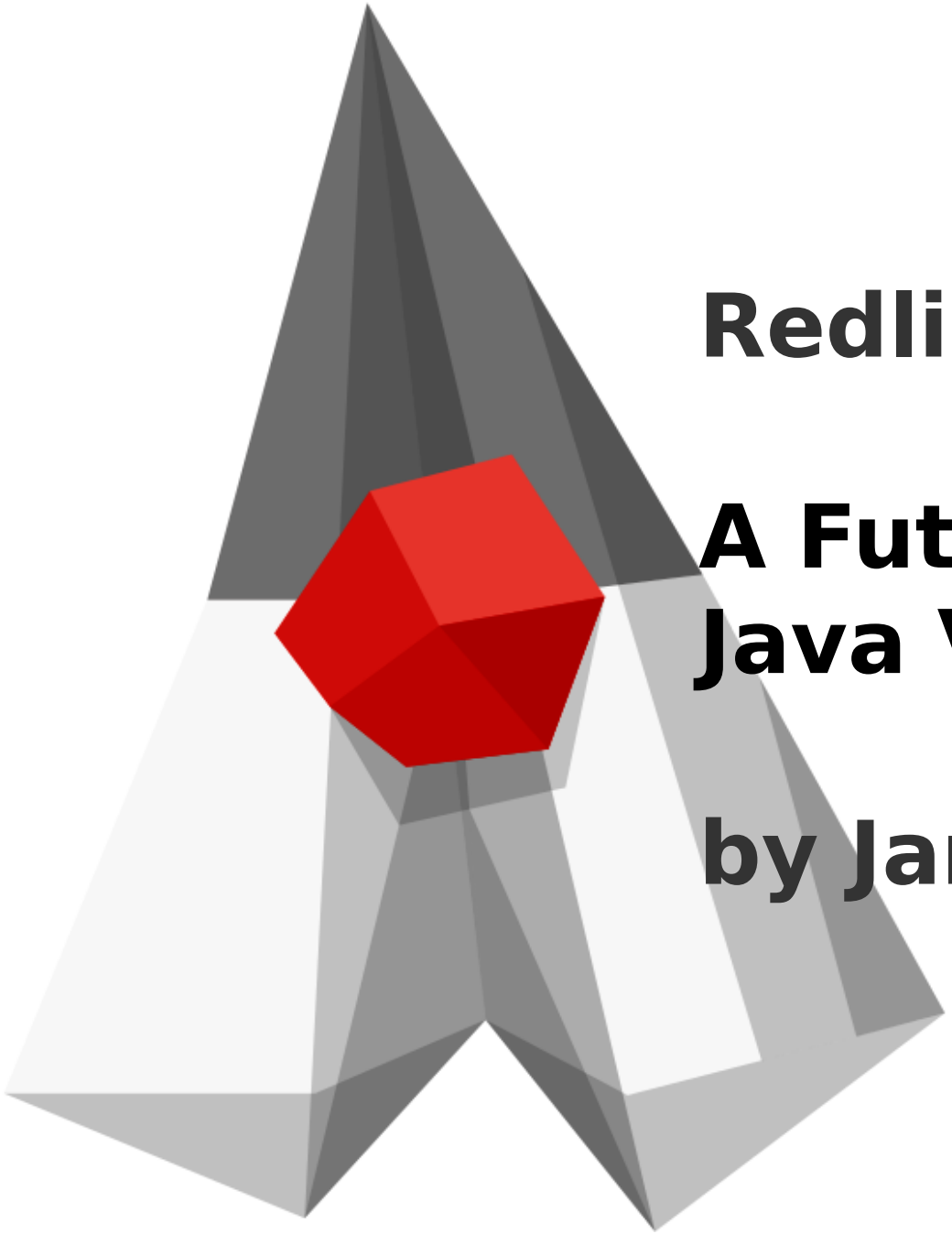
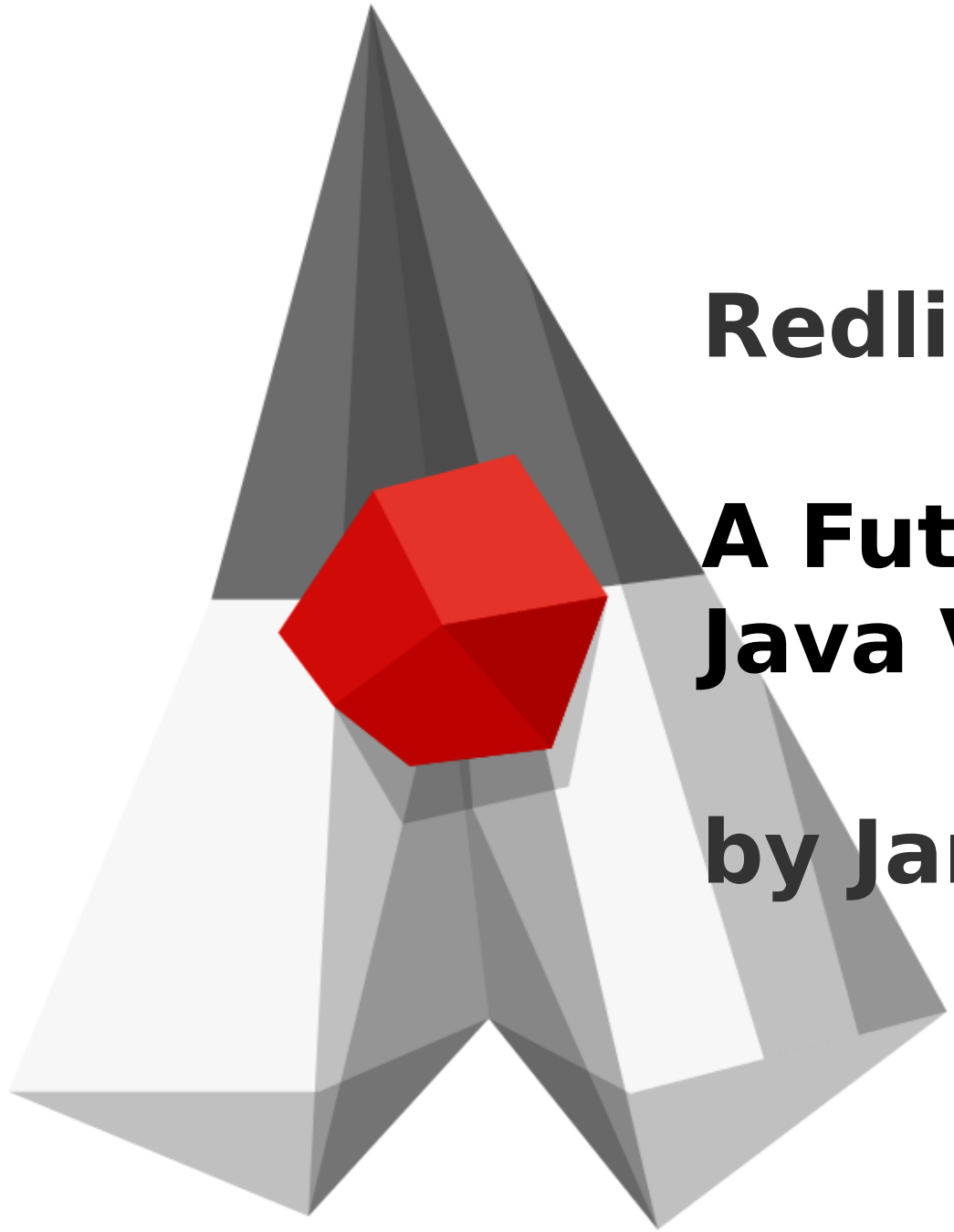


Redline Smalltalk:

A Future On The Java Virtual Machine

by James Ladd

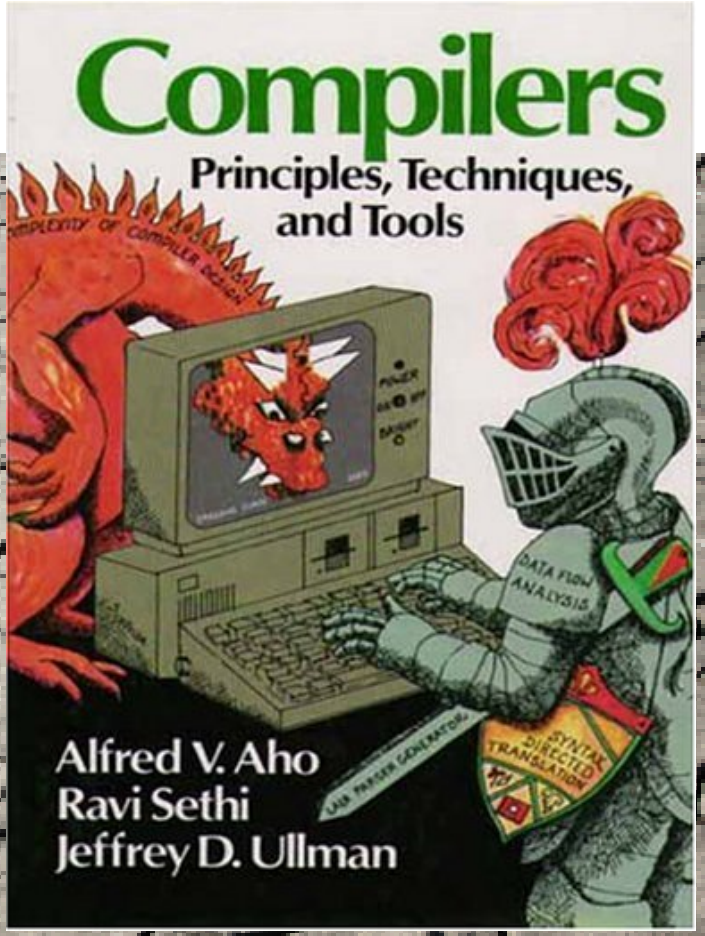


Redline Smalltalk:

A Future On The Java Virtual Machine

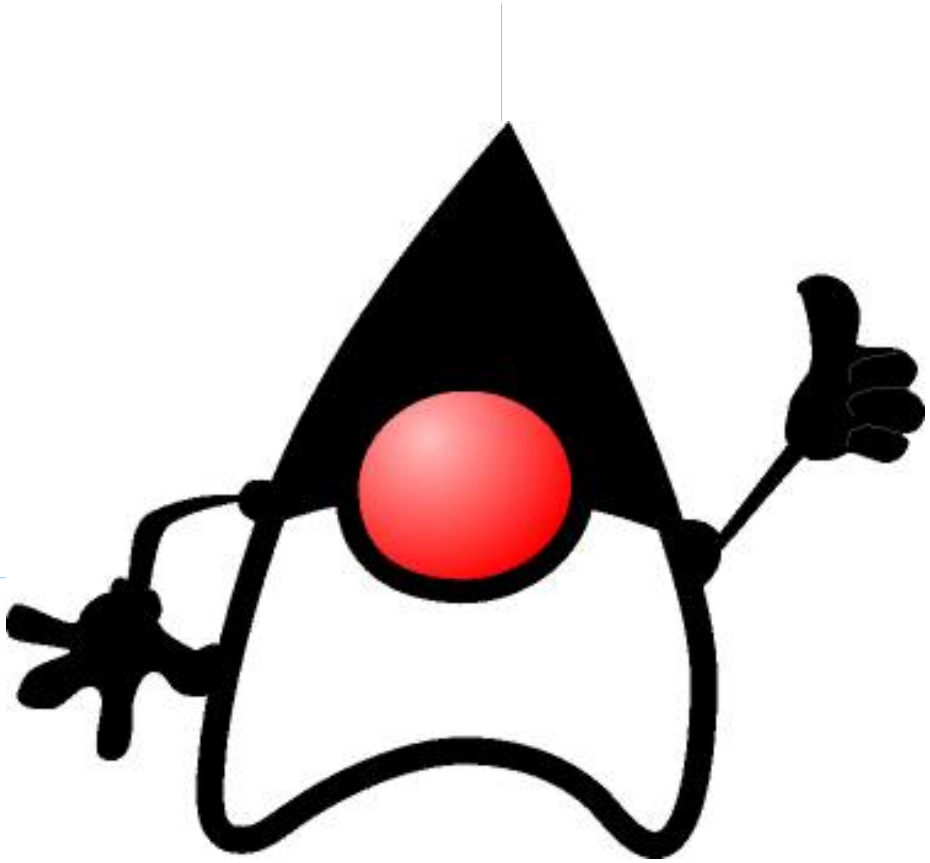
by James Ladd

HARD !


Compilers
Principles, Techniques, and Tools
Alfred V. Aho
Ravi Sethi
Jeffrey D. Ullman

JVM = Java Virtual Machine

Why Java Virtual Machine?



Why The Java Virtual Machine?

Everywhere

On Everything

Great Performance

Always Improving

Why The Java Virtual Machine?

Some Companies

Only Support

The JVM!

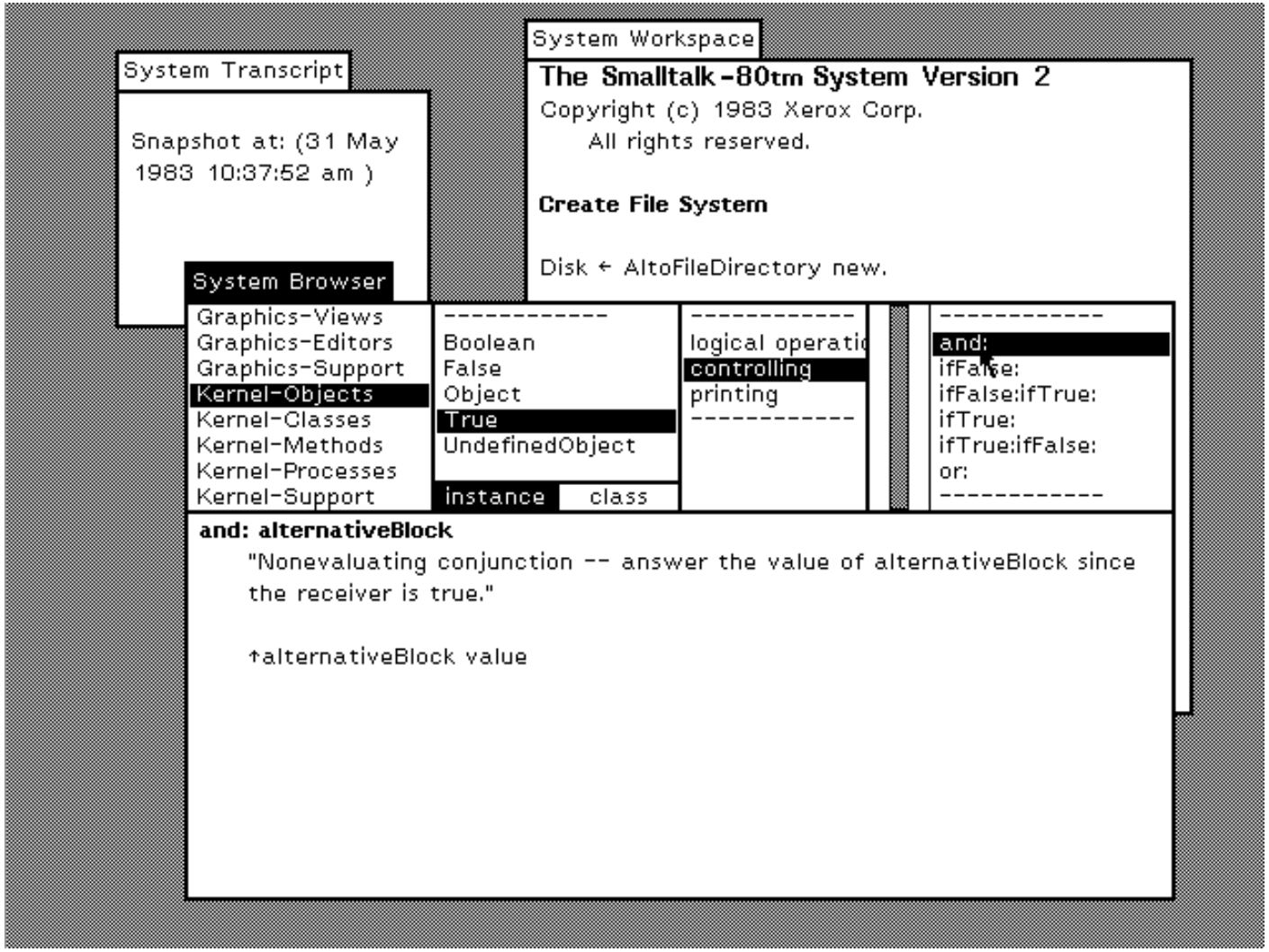
Why The Java Virtual Machine?

Some Companies

Only Support

The JVM!

Why Smalltalk?



Why Smalltalk?

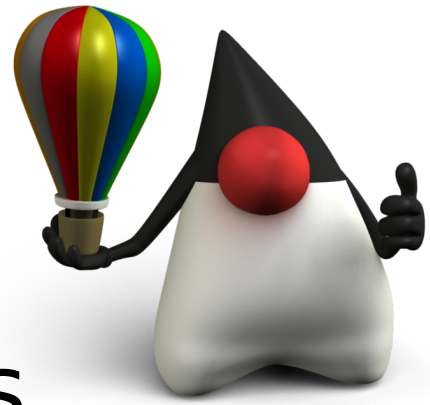
Productive

Flexible

Simple and Pure

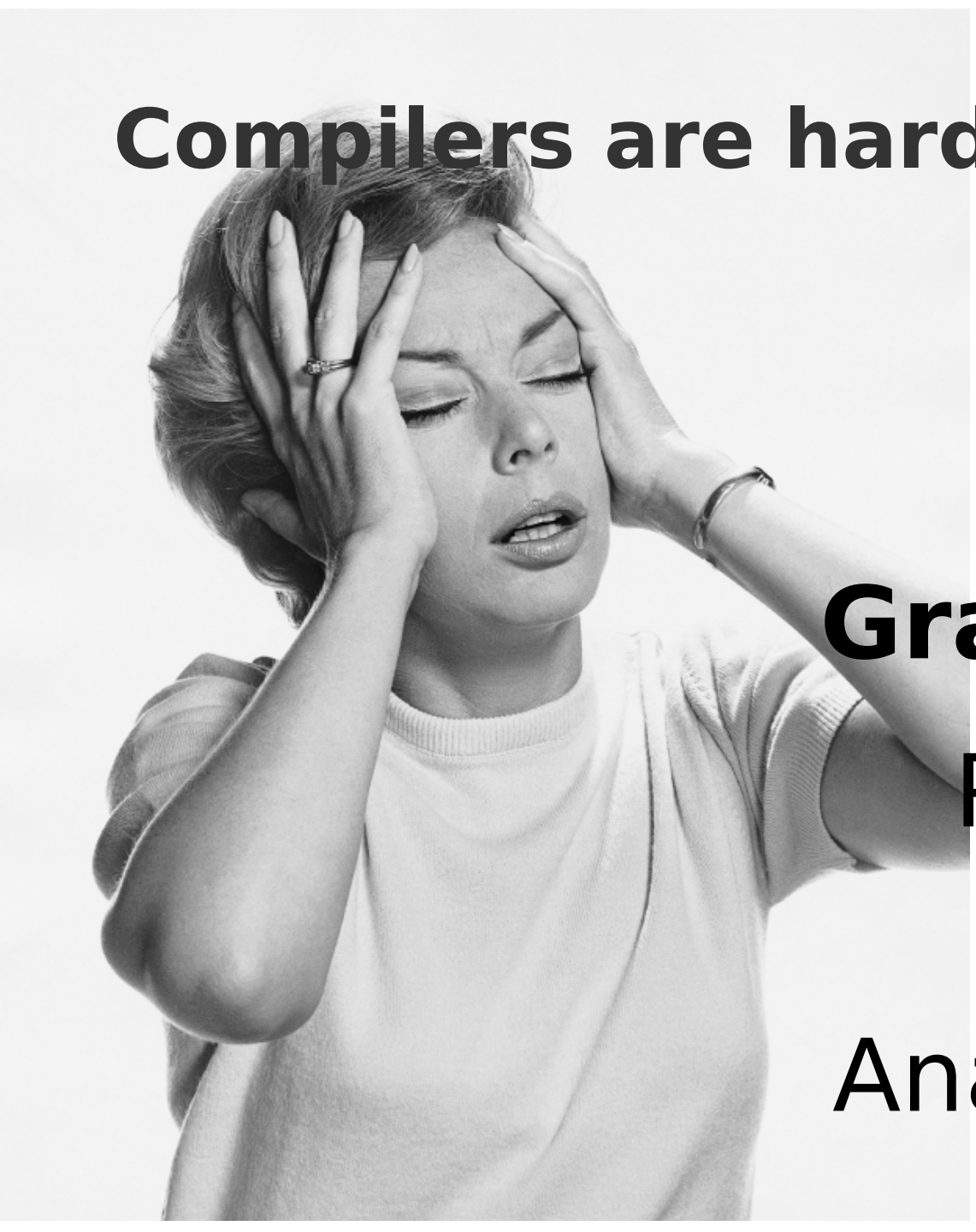
Easy to Learn

Why Smalltalk On The Java Virtual Machine?



"... because nothing is as productive as Smalltalk, and the App has to run on the Java Virtual Machine." – Redline Smalltalk.

Compilers are hard!



Lexer

Bytecodes

Grammar

Parser

Runtime

Analyser

The Pipe Line

From Source to Executable

Grammar: Defines Syntax -

From Small Things Big Things Grow

Parser:

Turns a Stream of Characters into a Tree of Nodes

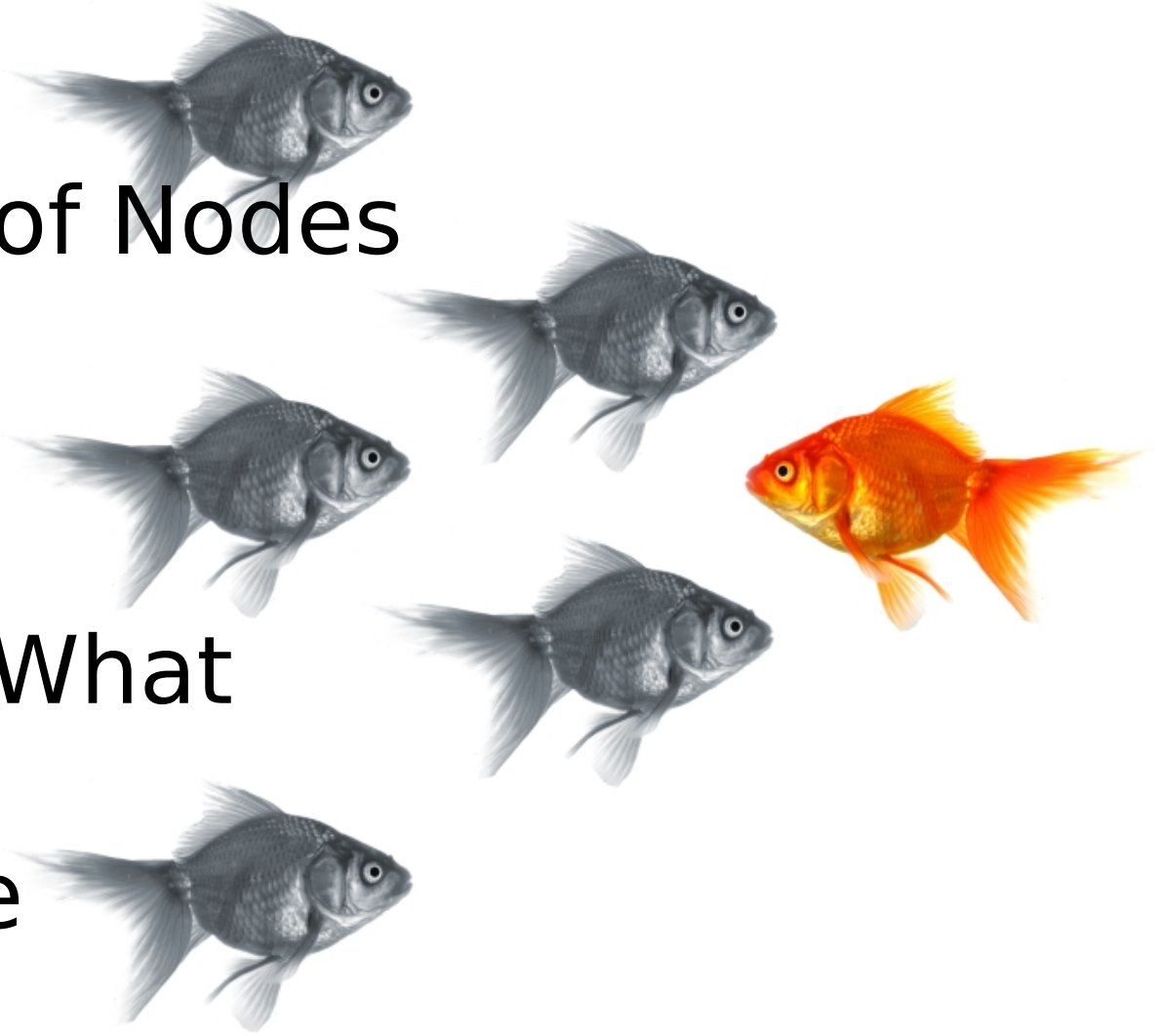
Analyser:

Walks Tree of Nodes

to

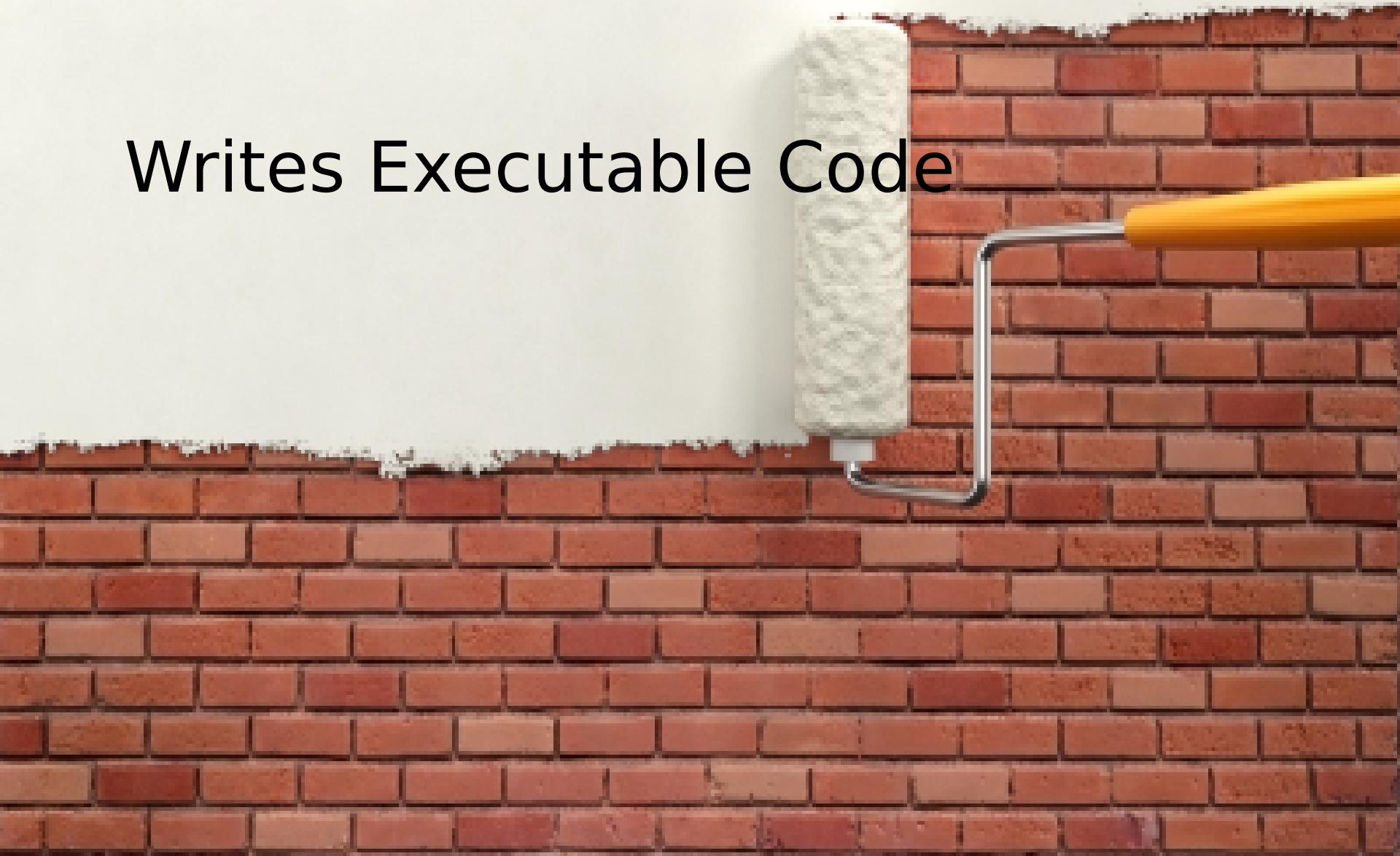
Determine What

to Generate



Generator:

Writes Executable Code



Runtime:

A Language Support Library

Class Library:

A Developer Support Library

Integration: Choice #1

Foreign Or Native?

Integration: Choice #2

Integrate With Other Classes?

You CAN make a language for the JVM...

Not as hard as you might think.

The Execution Pipeline

Bringing It All Together

Smalltalk-80

A Simple language:

5 Keywords Vs Java's 53

self, super,

true, false,

nil

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while
true	false	null		

Smalltalk-80

Simple expression syntax:

```
receiver message[: argument].
```

Smalltalk-80

Simple expression syntax:

```
receiver message[: argument].

1 + 1.
calendar at: date put: event.
vote increment.
```

Smalltalk-80

Simple expression syntax:

```
beer isGood: [beer drink]
```

Smalltalk-80 & Java

```
map at: key put: value.
```

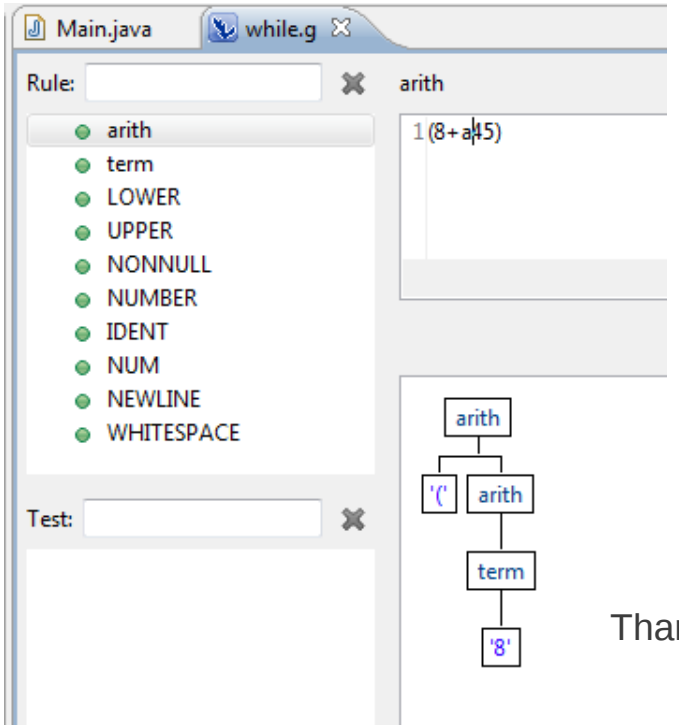
```
map.atPut(key, value);
```

Compiling An Example

```
Transcript show: 'Hello javaOne'
```

Compiling An Example - Grammar: Defines Syntax

```
keyword_expression returns [KeywordExpression keywordExpression]
  @init { kexpr = new KeywordExpression(); }
  : (
      WHITESPACE KEYWORD bod {
      $kexpr.add($KEYWORD.text, $KEYWORD.line,   $bod.binObjDescription); }
    )+
  ;
```

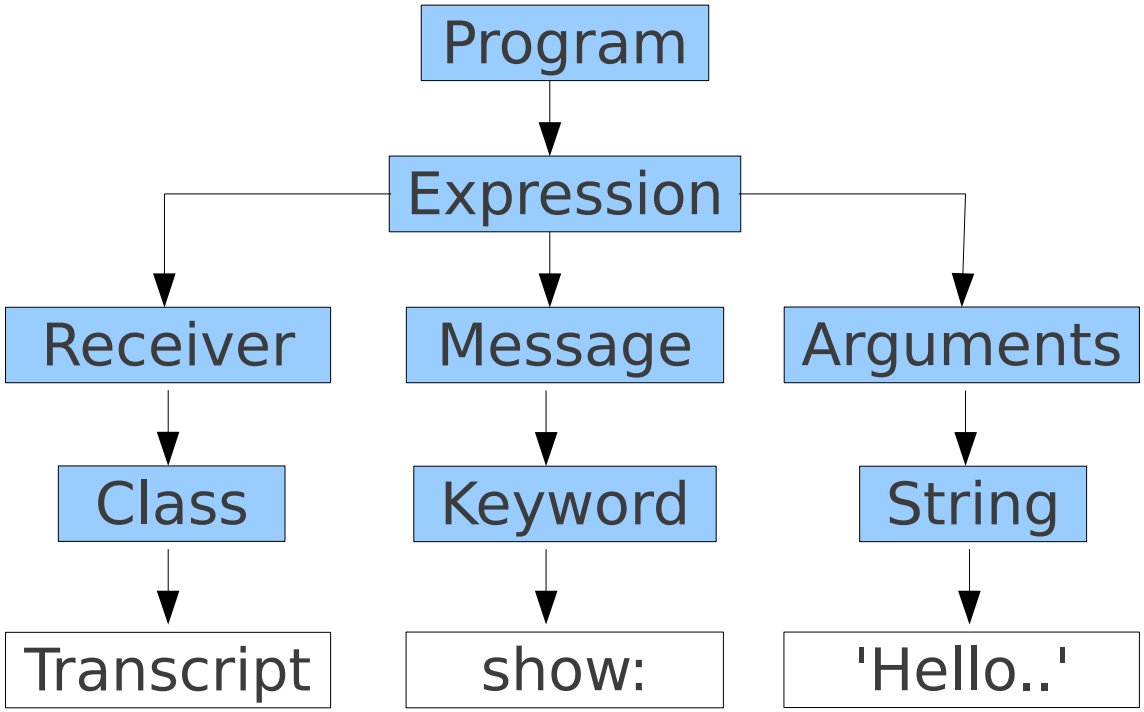


← ANTLRWorks for Eclipse and IntelliJ.
I used both.

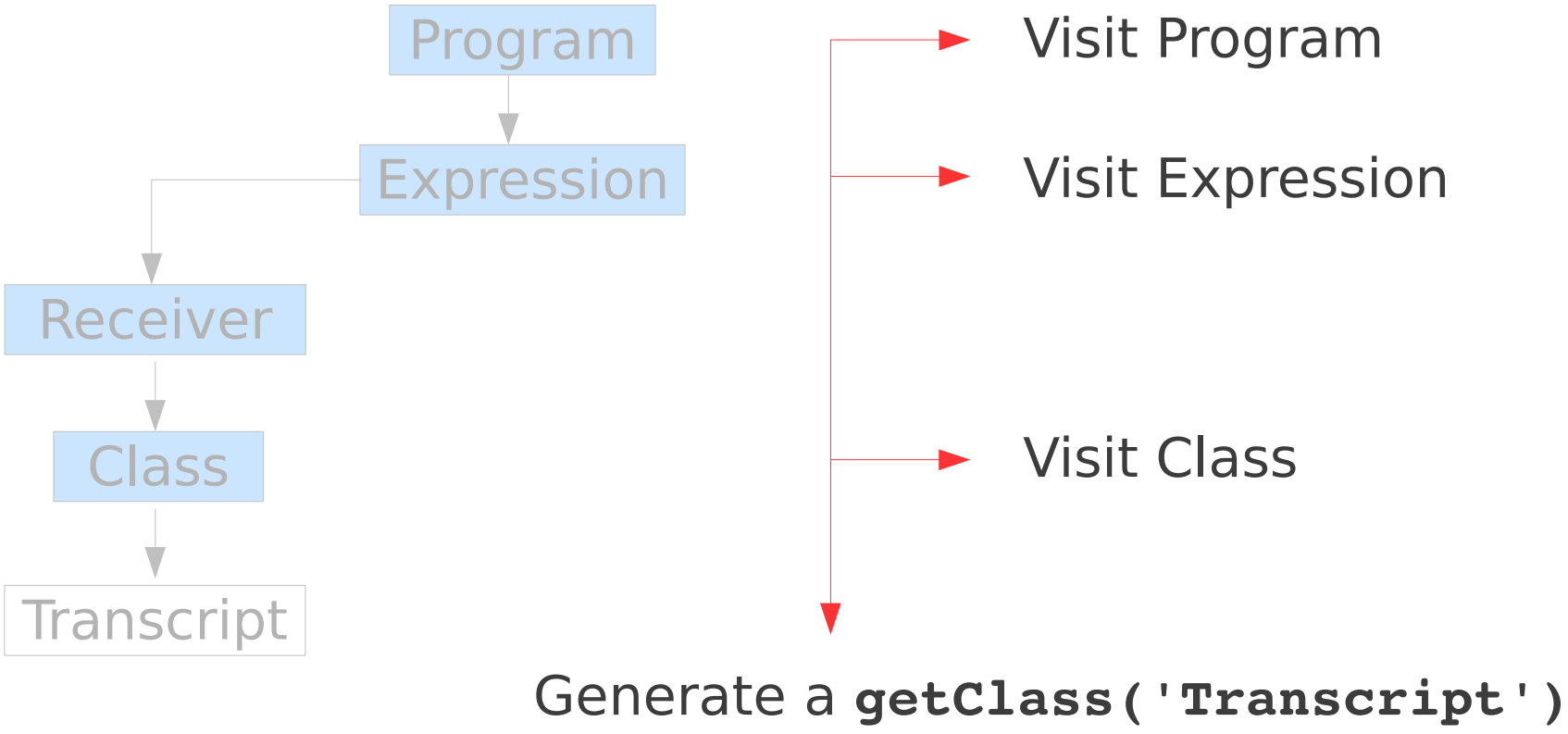
Thank you - Jim Idle - Temporal Wave LLC (jimi@idle.ws)

Compiling An Example - Parser: Characters to Nodes

```
Transcript show: 'Hello javaOne'
```



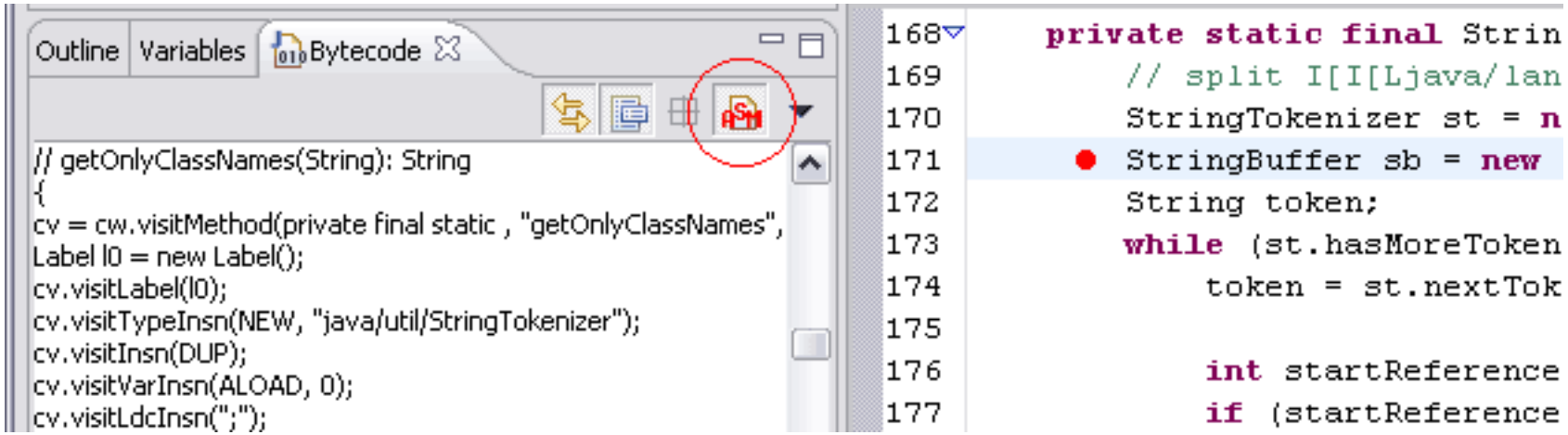
Compiling An Example - Analyser: What To Generate



Compiling An Example - Generator: Writes Bytecode

Object Web - ASM Java bytecode manipulation and analysis framework

Generate a **`getClass('Transcript')`**



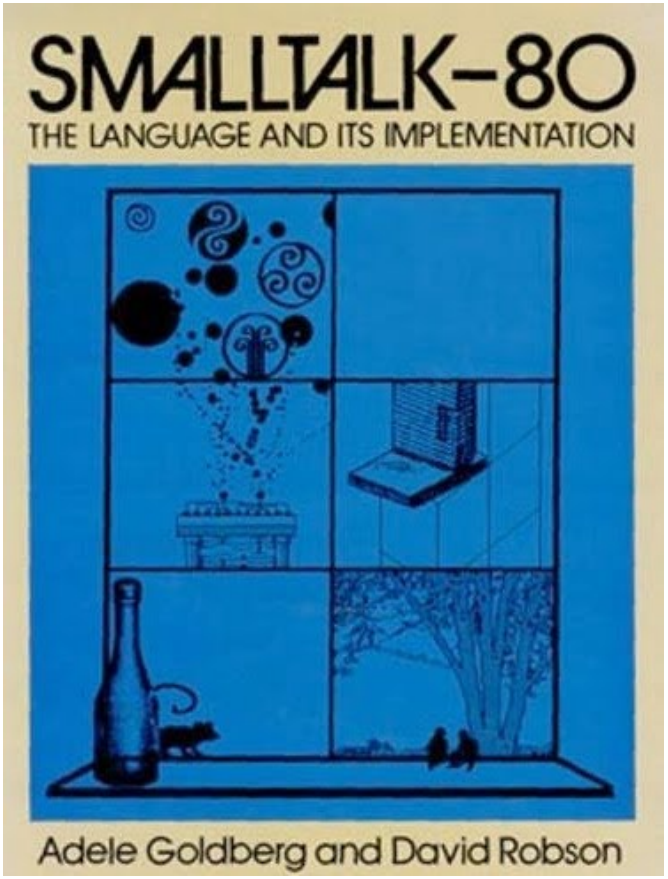
Compiling An Example - Runtime: Supports Language

Message Passing
– Custom Method Lookup And Dispatch.
– Charles Nutter added InvokeDynamic.

Smalltalk Object / Metaclass Hierarchy
– Not Java Hierarchy

Block Support
– Lambdas in Smalltalk.

Compiling An Example - Class Library: Supports You



Can use any Java Library with dynamic adaptors.

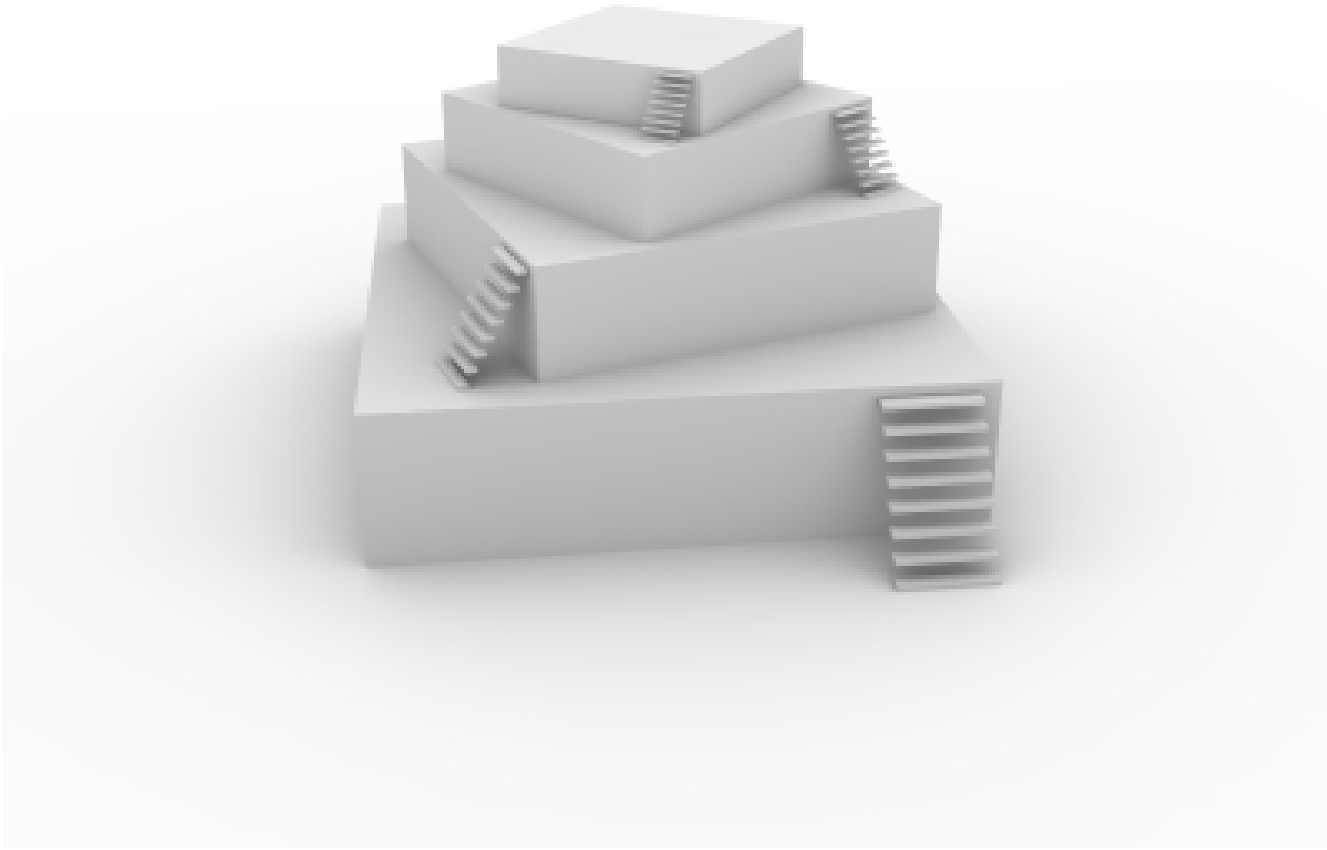
Compiling An Example - Integration

```
+ stdout: string

    JVM getStatic: 'java/lang/System' named: 'out'
          as: 'Ljava/io/PrintStream;'.

      JVM arg: 0.

      JVM invokeVirtual: 'java/io/PrintStream'
          method: 'print'
          matching: '(Ljava/lang/Object;)V'.
```

Compilers Are Not *That* Hard

Your Own Language?

The Java Virtual Machine (JVM) is a great platform.



Redline is Smalltalk *for you*...

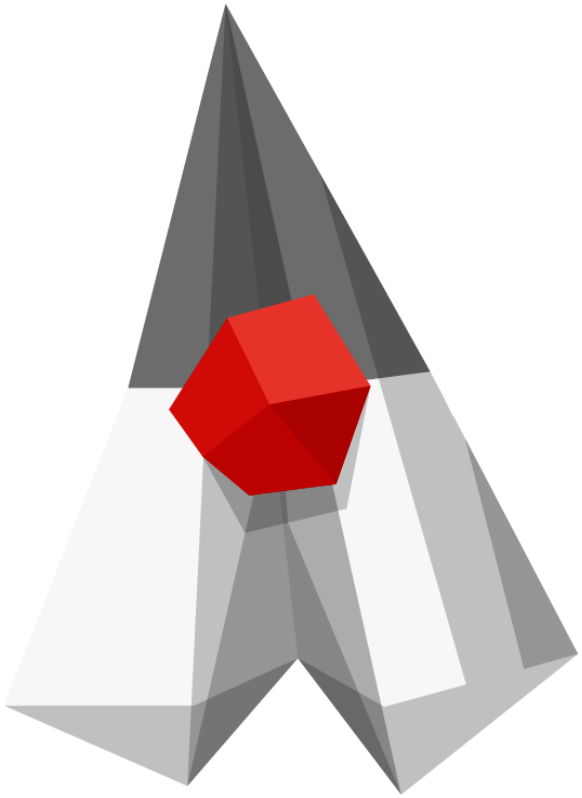
...supporting the way you work today.

Ken Gilmer

Konstantin Bulenkov

You can make a language for the JVM...

Not as hard as you might think.



Questions?

James Ladd @jamesladd
object@redline.st http://redline.st
www.jamesladdcode.com