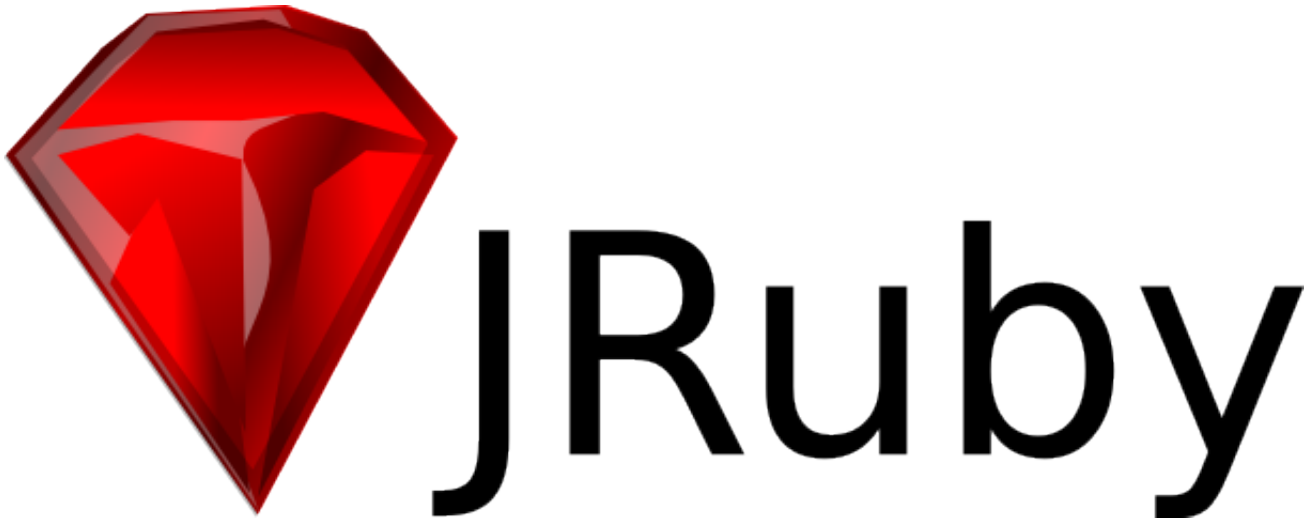
RubyConfBR 2011



on Steroids

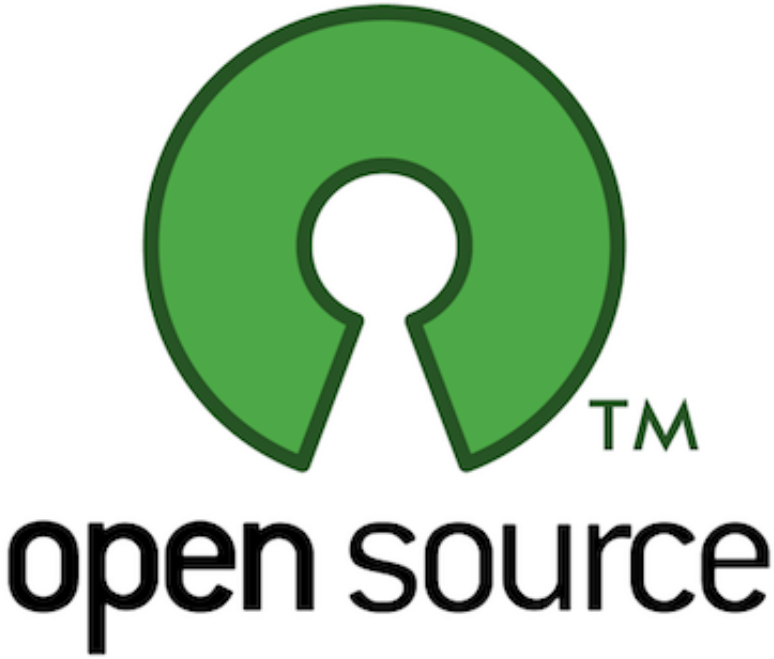
@qmx | http://qmx.me

whoami?

developer @

instructor @

opensource freak



dyn.js creator

@dynjs | http://dynjs.org

JRuby contributor

TorqueBox contributor

geek

JRuby

under the hood

JVM

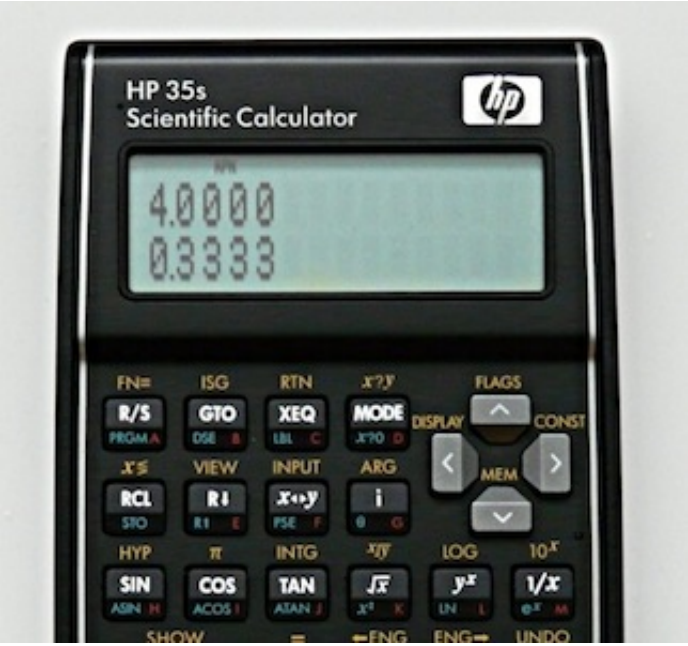
Java

run to the hills!

great VM

stack-based

stack-based


HP 35s
Scientific Calculator
4.0000
0.3333
FN=
ISG
RTN
FLAGS
R/S
GTO
XEQ
MODE
DISPLAY
CONST
MEM
VIEW
INPUT
ARG
RCL
STO
HYP
INTG
LOG
SIN
COS
TAN
SHOW
UNDO

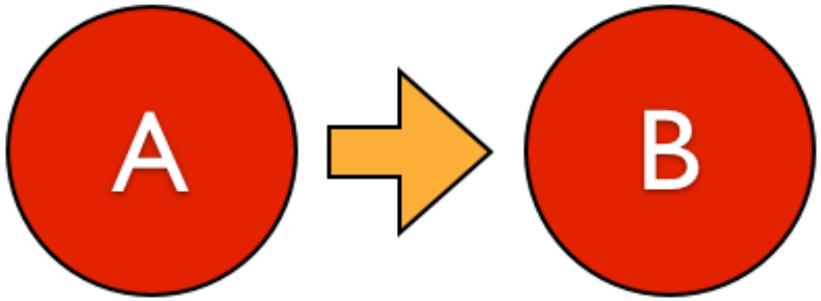
static-typed

static-typed

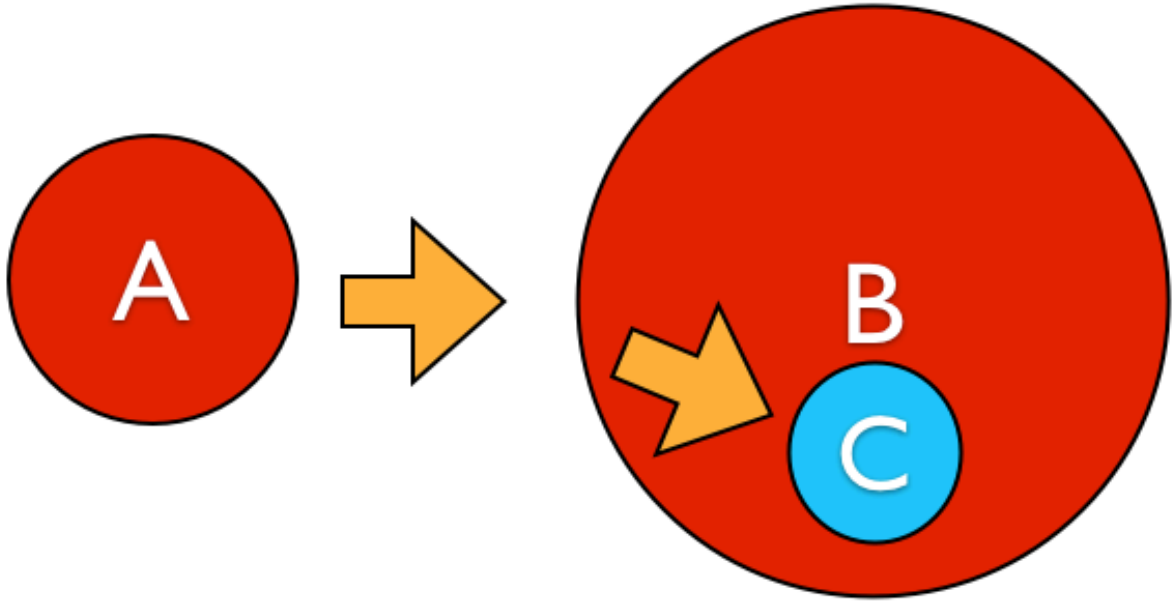
```
Bike bike = new Bike();
bike.stop();
```

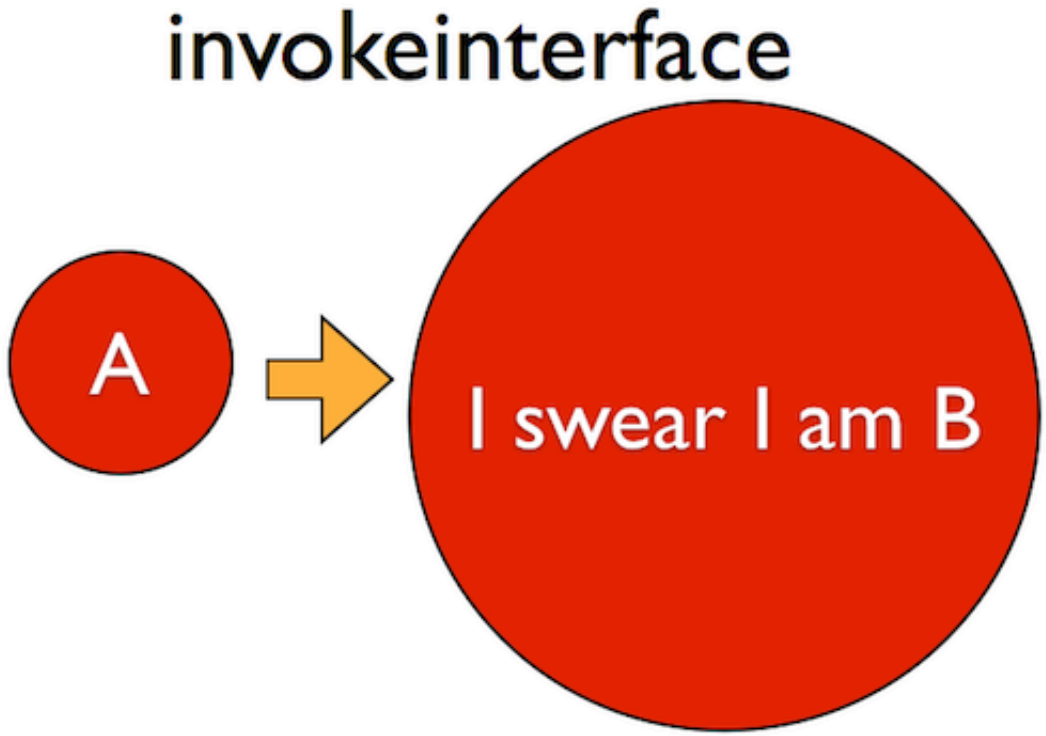
know thy enemy

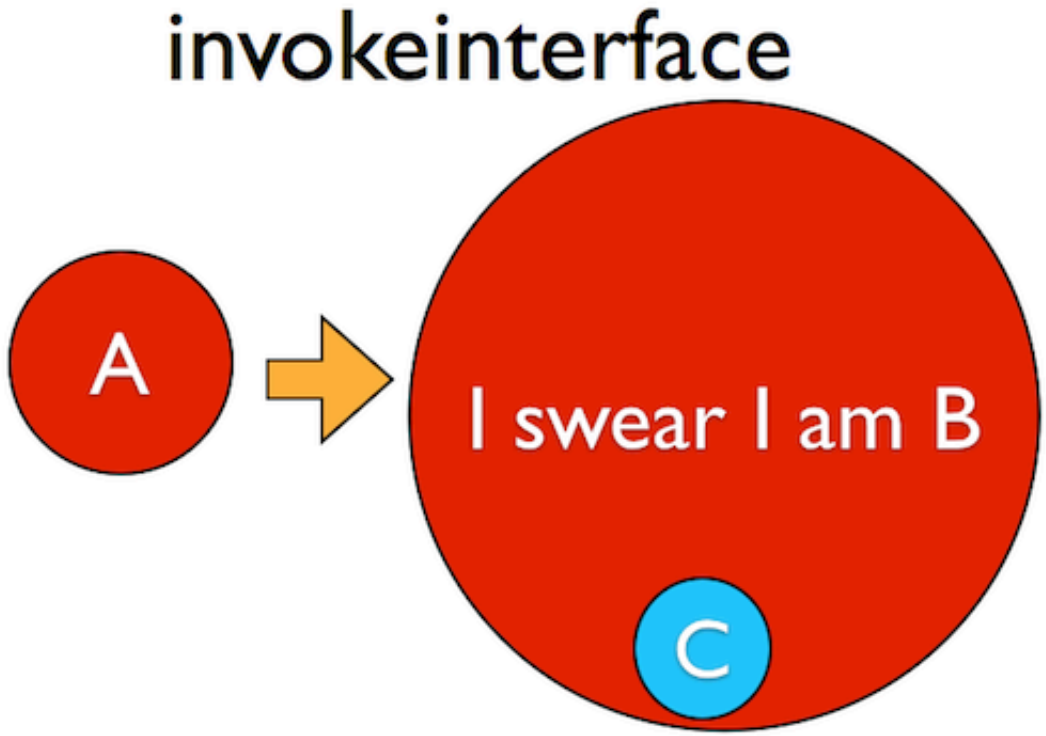
invokestatic

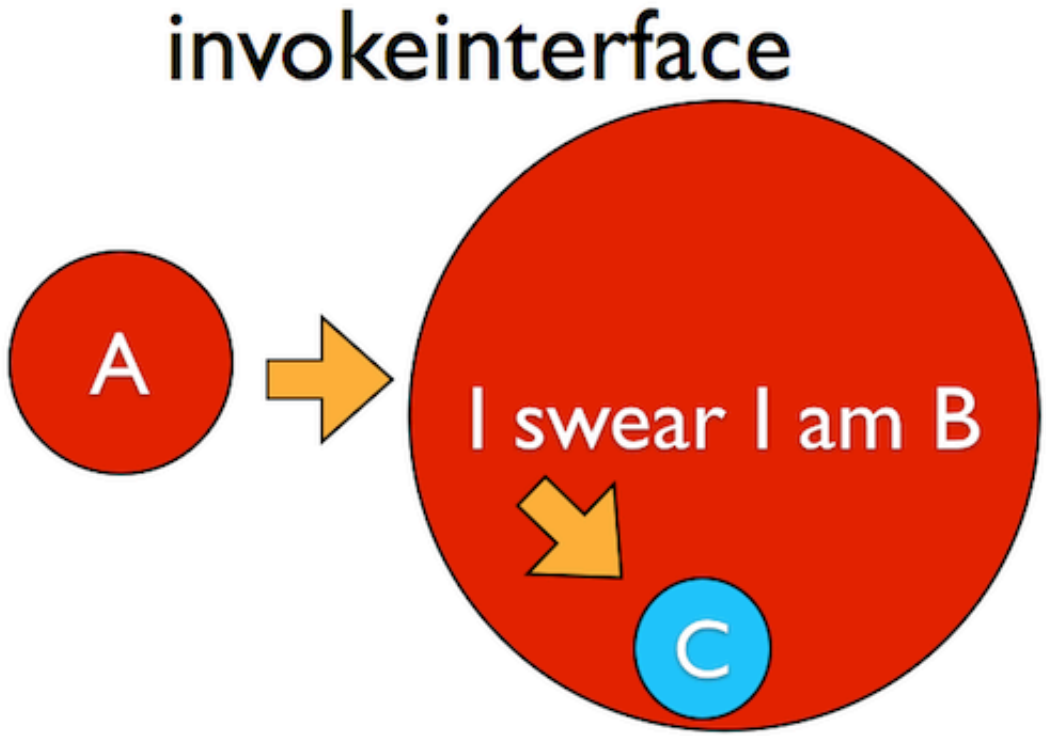


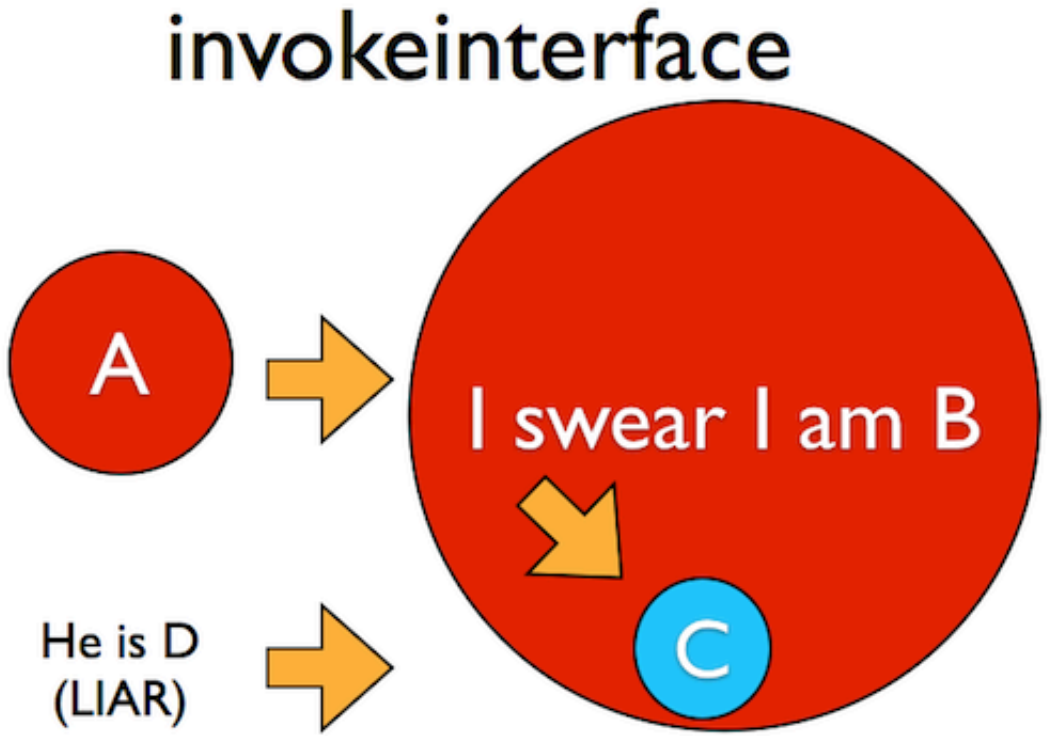
invokevirtual




invokeinterface
A
I swear I am B


invokeinterface
A
I swear I am B
C


invokeinterface
A
I swear I am B
C


invokeinterface
A
I swear I am B
C
He is D
(LIAR)

so

```
@shed = BikeShed.new
```

so.

```
@shed = BikeShed.new

# java Bike?
@shed << bike
```

what's the type of BikeShed?

MetaClass

MetaClass

```
public static IRubyObject
invoke(ThreadContext context,
    IRubyObject self, String
name, IRubyObject arg0) {
    return
self.getMetaClass().finvoke(co
ntext, self, name, arg0);
```

```
}
```

ZOMG

focus, focus!

```
self.getMetaClass().finvoke(co
ntext, self, name, arg0);
```

method missing!

```
DynamicMethod method =
searchMethod(name);
if
(shouldCallMethodMissing(metho
d)) {
    return ......
}
return method.call(......)
```

few calls later...

inside the metaclass

```
DynamicMethod method =
getMethods().get(name);
```

getMethods()?

it's a Map!

```
Map<String, DynamicMethod>
```

Ruby Methods become Java Classes

back to the BikeShed...

```
# pseudocode
BikeShed.methods['<<']
   .call(context @shed, bike)
```

```
25: aload_1
26: aload_2
27: aload           5
29: invokestatic    #31
```

this #31

```
RubyKernel.sleep:(Lorg/jruby/r
untime/ThreadContext;Lorg/jrub
y/runtime/builtin/IRubyObject;
[Lorg/jruby/runtime/builtin/IR
ubyObject;)Lorg/jruby/runtime/
builtin/IRubyObject
```

WTF

this #31 for dummies

Method RubyKernel.sleep

has arguments (ThreadContext, IRubyObject, IRubyObject[])

and returns an IRubyObject

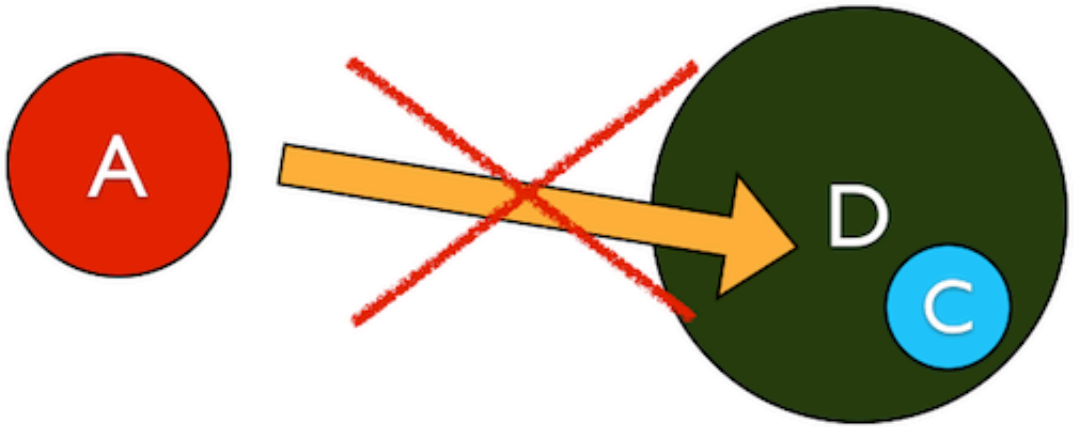
hard to write

we ought to know the types

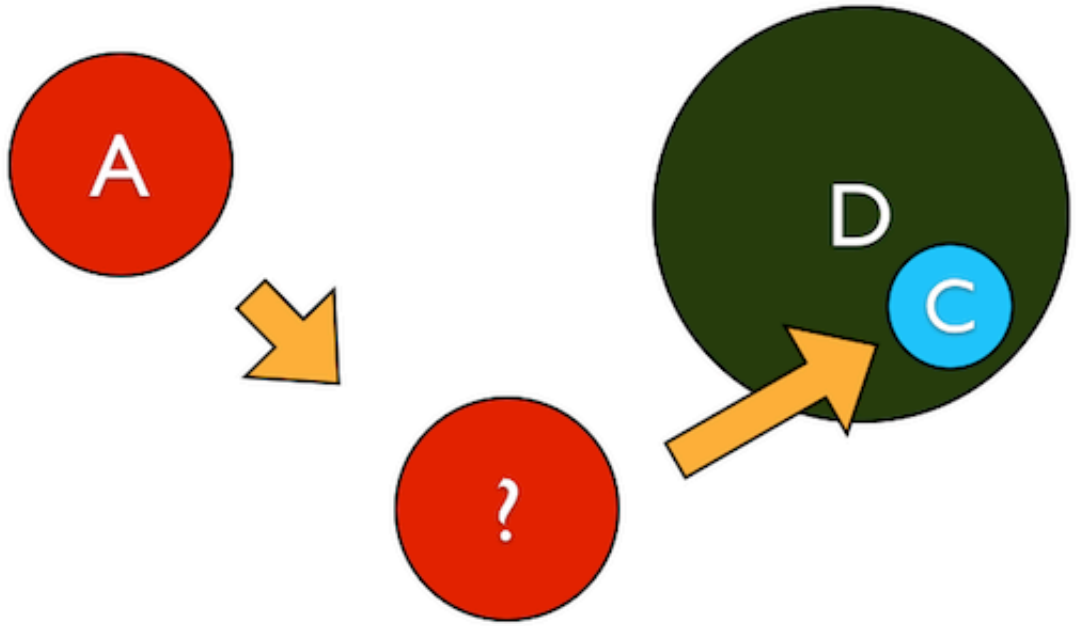
Java7 new features

invokedynamic

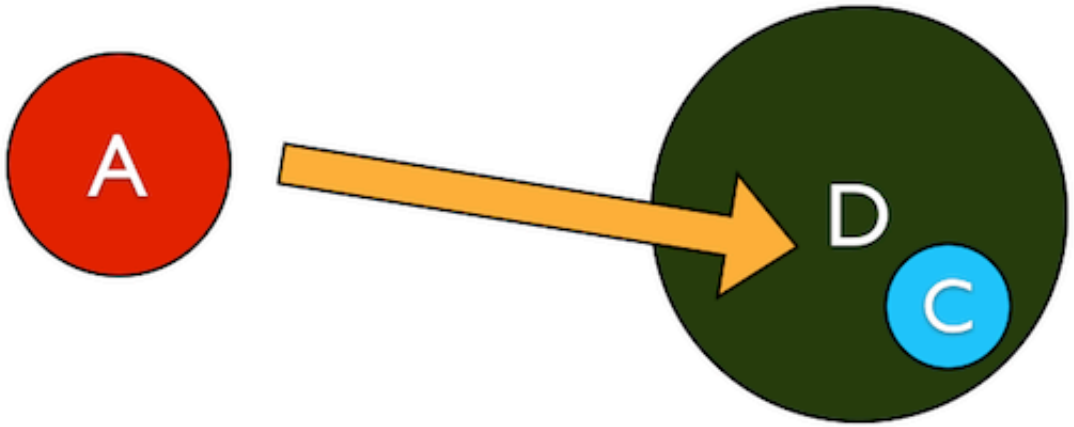
invokedynamic



invokedynamic



invokedynamic



what this means?

our @shed << bike

```
invokedynamic <<(Object) ...
```

caveat - java hates <<

<< == append(..)

where the types gone?

polymorphic signatures!

where the searchMethod gone?

(void *)

function pointers!

MethodHandles

referencing @shed.<<

```
# what signature do I want?
type =
methodType(IRubyObject.class,
IRubyObject.class)
handle = lookup()
   .findVirtual("append",
type);
```

ugly?

only if you don't know reflection api

performs almost equal invokevirtual

why this matters?

quoting @fabiokung

"deleting code is...

quoting @fabiokung

better than writing good code"

why this matters?

JIT

Adaptive Optimization

inlining

inlining

```
class Shouter
  def shout!
    puts "yay!"
  end
end
```

inlining

```
shouter = Shouter.new
100.times { shouter.shout!}
```

inlining

```
100.times { puts "yay!" }
```

MHs fold away

Ruby Constants

constants

```
X = "lol"
 => "lol"
X = "wut"
(irb):2 warning: already
initialized constant X
 => "wut"
```

Y U NO CONSTANT!

before Java7

generations

synchronized access

after Java7

SwitchPoint

global JVM fuse

fast and slow paths

SwitchPoints

two MHs

One points to the direct access codepath

Other points to the re-assign codepath

thread-safe!

why should I care?

granted that you use constants as constants

can be faster than stock java code!

current status

some stuff got slower

some stuff got wicked fast, w/o optimization

the future is bright ahead!

that's all folks!

?

thanks!