

CRuby 1.9.1

Method Dispatch & Constant Lookup

Tilman Giese

May 25, 2010

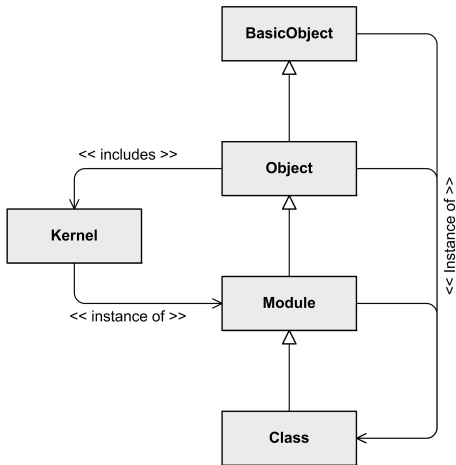
- ▶ Ruby Overview
 - ▶ History & Implementations
 - ▶ Meta Model
- ▶ CRuby 1.9.1
 - ▶ Source Code Structure
 - ▶ Method Dispatch
 - ▶ Constant Lookup

- ▶ Yukihiro Matsumoto
- ▶ Ruby 0.95: December 21, 1995

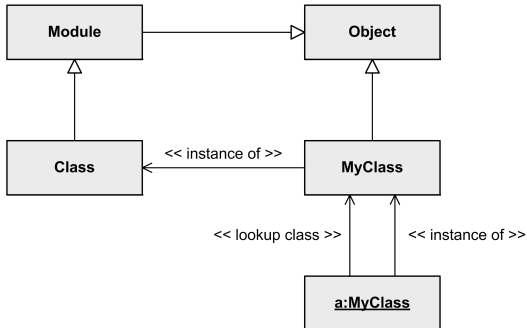
"I wanted a scripting language that was more powerful than Perl, and more object-oriented than Python. That's why I decided to design my own language."

- ▶ CRuby
 - ▶ 1.8.7 (aka MRI)
 - ▶ 1.9.1 (aka YARV)
- ▶ JRuby 1.5.0
- ▶ Rubinius 1.0.0
- ▶ IronRuby 1.0
- ▶ MacRuby 0.6

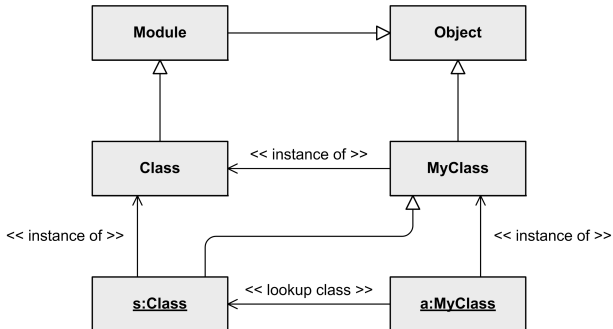
Ruby Meta Model



Singleton Classes

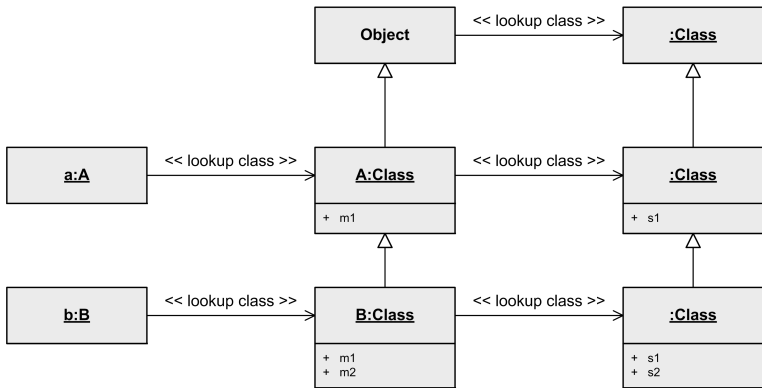


Singleton Classes



CRuby Source Code

Instance vs. Class/Module Methods



- ▶ global, 2048 entries
- ▶ cache entry is $5 * \text{long} = 40$ Bytes
- ▶ total cache size: 80 KiBytes
- ▶ $h(\text{class}, \text{mid}) = ((\text{class} \gg 3) \wedge \text{mid}) \& 0x7FF$
- ▶ $0x7FF =$ last 13 bits

```
def resolve_constant(nesting, constant)
  nesting = Module.nesting
  nesting << Object # Last resort
  nesting.each do |m|
    m.ancestors.each do |a| # includes m too
      if a.const_defined?(constant)
        return a.const_get(constant)
      end
    end
  end
  nesting[0].const_missing(constant)
end
```

- ▶ Method Dispatch
 - ▶ Mixins are part of the super class chain
 - ▶ Method lookup is easy to track
 - ▶ Global method cache might not be the best solution
- ▶ Constant Lookup
 - ▶ Unusual semantics
 - ▶ Constants are stored in the same table as instance variables
 - ▶ Straight forward implementation